In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
df = pd.read_csv("dataset.csv", low_memory=False)
```

In [4]:
```
df
```

Out[4]:

	stn_code	sampling_date	state	location	agency	type	so2	no2	rspm
0	150	February - M021990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	4.8	17.4	NaN
1	151	February - M021990	Andhra Pradesh	Hyderabad	NaN	Industrial Area	3.1	7.0	NaN
2	152	February - M021990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	6.2	28.5	NaN
3	150	March - M031990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	6.3	14.7	NaN
4	151	March - M031990	Andhra Pradesh	Hyderabad	NaN	Industrial Area	4.7	7.5	NaN
...	...	...	...	...	...	...	...	...	...
435737	SAMP	24-12-15	West Bengal	ULUBERIA	West Bengal State Pollution Control Board	RIRUO	22.0	50.0	143.0
435738	SAMP	29-12-15	West Bengal	ULUBERIA	West Bengal State Pollution Control Board	RIRUO	20.0	46.0	171.0
435739	NaN	NaN	andaman-and-nicobar-islands	NaN	NaN	NaN	NaN	NaN	NaN
435740	NaN	NaN	Lakshadweep	NaN	NaN	NaN	NaN	NaN	NaN
435741	NaN	NaN	Tripura	NaN	NaN	NaN	NaN	NaN	NaN

435742 rows × 13 columns

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

Out[5]:

	stn_code	sampling_date	state	location	agency	type	so2	no2	rspm	spm	locat
0	150	February - M021990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	4.8	17.4	NaN	NaN	
1	151	February - M021990	Andhra Pradesh	Hyderabad	NaN	Industrial Area	3.1	7.0	NaN	NaN	
2	152	February - M021990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	6.2	28.5	NaN	NaN	
3	150	March - M031990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	6.3	14.7	NaN	NaN	
4	151	March - M031990	Andhra Pradesh	Hyderabad	NaN	Industrial Area	4.7	7.5	NaN	NaN	

In [6]:
```
df.tail()
```

Out[6]:

	stn_code	sampling_date	state	location	agency	type	so2	no2	rspm	sp
435737	SAMP	24-12-15	West Bengal	ULUBERIA	West Bengal State Pollution Control Board	RIRUO	22.0	50.0	143.0	Na
435738	SAMP	29-12-15	West Bengal	ULUBERIA	West Bengal State Pollution Control Board	RIRUO	20.0	46.0	171.0	Na
435739	NaN	NaN	andaman-and-nicobar-islands	NaN	NaN	NaN	NaN	NaN	NaN	Na
435740	NaN	NaN	Lakshadweep	NaN	NaN	NaN	NaN	NaN	NaN	Na
435741	NaN	NaN	Tripura	NaN	NaN	NaN	NaN	NaN	NaN	Na

In [7]:
```
df.nunique()
```

Out[7]:
```
stn_code                         745
sampling_date                   5485
state                             37
location                         304
agency                            64
type                              10
so2                             4197
no2                             6864
rspm                            6065
spm                             6668
location_monitoring_station      991
pm2_5                            433
date                            5067
dtype: int64
```

In [8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 435742 entries, 0 to 435741
Data columns (total 13 columns):
 #   Column                        Non-Null Count   Dtype  
---  ------                        --------------   -----  
 0   stn_code                      291665 non-null  object 
 1   sampling_date                 435739 non-null  object 
 2   state                         435742 non-null  object 
 3   location                      435739 non-null  object 
 4   agency                        286261 non-null  object 
 5   type                          430349 non-null  object 
 6   so2                           401096 non-null  float64
 7   no2                           419509 non-null  float64
 8   rspm                          395520 non-null  float64
 9   spm                           198355 non-null  float64
 10  location_monitoring_station   408251 non-null  object 
 11  pm2_5                         9314 non-null    float64
 12  date                          435735 non-null  object 
dtypes: float64(5), object(8)
memory usage: 43.2+ MB
```

In [9]:
```
df.isnull().sum()
```

Out[9]:
```
stn_code                        144077
sampling_date                        3
state                                0
location                             3
agency                          149481
type                              5393
so2                              34646
no2                              16233
rspm                             40222
spm                             237387
location_monitoring_station      27491
pm2_5                           426428
date                                 7
dtype: int64
```

In [10]:
```
df.describe()
```

Out[10]:

	so2	no2	rspm	spm	pm2_5
count	401096.000000	419509.000000	395520.000000	198355.000000	9314.000000
mean	10.829414	25.809623	108.832784	220.783480	40.791467
std	11.177187	18.503086	74.872430	151.395457	30.832525
min	0.000000	0.000000	0.000000	0.000000	3.000000
25%	5.000000	14.000000	56.000000	111.000000	24.000000
50%	8.000000	22.000000	90.000000	187.000000	32.000000
75%	13.700000	32.200000	142.000000	296.000000	46.000000
max	909.000000	876.000000	6307.033333	3380.000000	504.000000

In [11]:
```
colu = ['stn_code','agency','sampling_date','location_monitoring_station']
df2 = df.drop(colu,axis=1)
```

In [12]:
```
df2
```

Out[12]:

	state	location	type	so2	no2	rspm	spm	pm2_5	date
0	Andhra Pradesh	Hyderabad	Residential, Rural and other Areas	4.8	17.4	NaN	NaN	NaN	2/1/1990
1	Andhra Pradesh	Hyderabad	Industrial Area	3.1	7.0	NaN	NaN	NaN	2/1/1990
2	Andhra Pradesh	Hyderabad	Residential, Rural and other Areas	6.2	28.5	NaN	NaN	NaN	2/1/1990
3	Andhra Pradesh	Hyderabad	Residential, Rural and other Areas	6.3	14.7	NaN	NaN	NaN	3/1/1990
4	Andhra Pradesh	Hyderabad	Industrial Area	4.7	7.5	NaN	NaN	NaN	3/1/1990
...	...	...	...	...	...	...	...	...	...
435737	West Bengal	ULUBERIA	RIRUO	22.0	50.0	143.0	NaN	NaN	12/24/2015
435738	West Bengal	ULUBERIA	RIRUO	20.0	46.0	171.0	NaN	NaN	12/29/2015
435739	andaman-and-nicobar-islands	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
435740	Lakshadweep	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
435741	Tripura	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

435742 rows × 9 columns

In [13]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 435742 entries, 0 to 435741
Data columns (total 9 columns):
 #   Column    Non-Null Count   Dtype  
---  ------    --------------   -----  
 0   state     435742 non-null  object 
 1   location  435739 non-null  object 
 2   type      430349 non-null  object 
 3   so2       401096 non-null  float64
 4   no2       419509 non-null  float64
 5   rspm      395520 non-null  float64
 6   spm       198355 non-null  float64
 7   pm2_5     9314 non-null    float64
 8   date      435735 non-null  object 
dtypes: float64(5), object(4)
memory usage: 29.9+ MB
```

In [14]:
```
df2['state'].value_counts()
```

Out[14]:
```
Maharashtra                   60384
Uttar Pradesh                 42816
Andhra Pradesh                26368
Punjab                        25634
Rajasthan                     25589
Kerala                        24728
Himachal Pradesh              22896
West Bengal                   22463
Gujarat                       21279
Tamil Nadu                    20597
Madhya Pradesh                19920
Assam                         19361
Odisha                        19279
Karnataka                     17119
Delhi                          8551
Chandigarh                     8520
Chhattisgarh                   7831
Goa                            6206
Jharkhand                      5968
Mizoram                        5338
Telangana                      3978
Meghalaya                      3853
Puducherry                     3785
Haryana                        3420
Nagaland                       2463
Bihar                          2275
Uttarakhand                    1961
Jammu & Kashmir                1289
Daman & Diu                     782
Dadra & Nagar Haveli            634
Uttaranchal                     285
Arunachal Pradesh                90
Manipur                          76
Sikkim                            1
andaman-and-nicobar-islands       1
Lakshadweep                       1
Tripura                           1
Name: state, dtype: int64
```

In [15]:
```
df2['type'].value_counts()
```

Out[15]:
```
Residential, Rural and other Areas    179014
Industrial Area                        96091
Residential and others                 86791
Industrial Areas                       51747
Sensitive Area                          8980
Sensitive Areas                         5536
RIRUO                                   1304
Sensitive                                495
Industrial                               233
Residential                              158
Name: type, dtype: int64
```

In [16]:
```
df2 = df2.dropna(axis = 0, subset = ['type'])
```

In [17]:
```
a = list(df2['type'])
for i in range(0, len(df2)):
    if str(a[i][0]) == 'R' and a[i][1] == 'e':
        a[i] = 'Residential'
    elif str(a[i][0]) == 'I':
        a[i] = 'Industrial'
    else:
        a[i] = 'Other'
```

In [18]:
```
df2['type'] = a
df2['type'].value_counts()
```

```
C:\Users\lenovo\AppData\Local\Temp\ipykernel_12812\1320634438.py:1: SettingWithCop
yWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  df2['type'] = a
```

Out[18]:
```
Residential    265963
Industrial     148071
Other           16315
Name: type, dtype: int64
```

In [19]:
```
df2
```

Out[19]:

	state	location	type	so2	no2	rspm	spm	pm2_5	date
0	Andhra Pradesh	Hyderabad	Residential	4.8	17.4	NaN	NaN	NaN	2/1/1990
1	Andhra Pradesh	Hyderabad	Industrial	3.1	7.0	NaN	NaN	NaN	2/1/1990
2	Andhra Pradesh	Hyderabad	Residential	6.2	28.5	NaN	NaN	NaN	2/1/1990
3	Andhra Pradesh	Hyderabad	Residential	6.3	14.7	NaN	NaN	NaN	3/1/1990
4	Andhra Pradesh	Hyderabad	Industrial	4.7	7.5	NaN	NaN	NaN	3/1/1990
...	...	...	...	...	...	...	...	...	...
435734	West Bengal	ULUBERIA	Other	20.0	44.0	148.0	NaN	NaN	12/15/2015
435735	West Bengal	ULUBERIA	Other	17.0	44.0	131.0	NaN	NaN	12/18/2015
435736	West Bengal	ULUBERIA	Other	18.0	45.0	140.0	NaN	NaN	12/21/2015
435737	West Bengal	ULUBERIA	Other	22.0	50.0	143.0	NaN	NaN	12/24/2015
435738	West Bengal	ULUBERIA	Other	20.0	46.0	171.0	NaN	NaN	12/29/2015

430349 rows × 9 columns

In [20]:
```
df2.isnull().sum()
```

Out[20]:
```
state            0
location         0
type             0
so2          34188
no2          15848
rspm         35030
spm         236748
pm2_5       421035
date             4
dtype: int64
```

In [21]:
```
percent = df2['so2'].isnull().sum()/df2.shape[0]
print(np.round(percent,2)*100)
```

```
8.0
```

In [22]:
```
percent = df2['no2'].isnull().sum()/df2.shape[0]
print(np.round(percent,2)*100)
```

```
4.0
```

In [23]:
```
percent = df2['rspm'].isnull().sum()/df2.shape[0]
print(np.round(percent,2)*100)
```

```
8.0
```

In [24]:
```
df2.describe()
```

Out[24]:

	so2	no2	rspm	spm	pm2_5
count	396161.000000	414501.000000	395319.000000	193601.000000	9314.000000
mean	10.758950	25.787465	108.888120	221.709847	40.791467
std	11.116237	18.454241	74.851223	151.394367	30.832525
min	0.000000	0.000000	0.000000	0.000000	3.000000
25%	5.000000	14.000000	56.000000	112.000000	24.000000
50%	8.000000	22.000000	90.000000	188.000000	32.000000
75%	13.500000	32.100000	142.000000	297.000000	46.000000
max	909.000000	876.000000	6307.033333	2610.000000	504.000000

In [25]:
```
df2['no2'].fillna(df['no2'].mean(),inplace = True)
```

```
C:\Users\lenovo\AppData\Local\Temp\ipykernel_12812\2964757312.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df2['no2'].fillna(df['no2'].mean(),inplace = True)
```

In [26]:
```
df2
```

Out[26]:

	state	location	type	so2	no2	rspm	spm	pm2_5	date
0	Andhra Pradesh	Hyderabad	Residential	4.8	17.4	NaN	NaN	NaN	2/1/1990
1	Andhra Pradesh	Hyderabad	Industrial	3.1	7.0	NaN	NaN	NaN	2/1/1990
2	Andhra Pradesh	Hyderabad	Residential	6.2	28.5	NaN	NaN	NaN	2/1/1990
3	Andhra Pradesh	Hyderabad	Residential	6.3	14.7	NaN	NaN	NaN	3/1/1990
4	Andhra Pradesh	Hyderabad	Industrial	4.7	7.5	NaN	NaN	NaN	3/1/1990
...	...	...	...	...	...	...	...	...	...
435734	West Bengal	ULUBERIA	Other	20.0	44.0	148.0	NaN	NaN	12/15/2015
435735	West Bengal	ULUBERIA	Other	17.0	44.0	131.0	NaN	NaN	12/18/2015
435736	West Bengal	ULUBERIA	Other	18.0	45.0	140.0	NaN	NaN	12/21/2015
435737	West Bengal	ULUBERIA	Other	22.0	50.0	143.0	NaN	NaN	12/24/2015
435738	West Bengal	ULUBERIA	Other	20.0	46.0	171.0	NaN	NaN	12/29/2015

430349 rows × 9 columns

In [27]:
```
df2.isnull().sum()
```

Out[27]:
```
state           0
location        0
type            0
so2         34188
no2             0
rspm        35030
spm        236748
pm2_5      421035
date            4
dtype: int64
```

In [28]:
```
df3 = df2.dropna(axis = 0, subset = ['rspm'])
```

In [29]:
```
df3.isnull().sum()
```

Out[29]:
```
state           0
location        0
type            0
so2         28801
no2             0
rspm            0
spm        229052
pm2_5      386065
date            4
dtype: int64
```

In [30]:
```
percent = df3['so2'].isnull().sum()/df3.shape[0]
print(np.round(percent,2)*100)
```

```
7.000000000000001
```

In [31]:
```
df3.describe()
```

Out[31]:

	so2	no2	rspm	spm	pm2_5
count	366518.000000	395319.000000	395319.000000	166267.000000	9254.000000
mean	10.352297	25.751649	108.888120	218.490501	40.701051
std	10.374559	17.747523	74.851223	148.768286	30.728628
min	0.000000	0.000000	0.000000	0.000000	3.000000
25%	4.833750	14.000000	56.000000	110.000000	24.000000
50%	8.000000	22.100000	90.000000	184.000000	32.000000
75%	13.000000	32.000000	142.000000	292.000000	46.000000
max	909.000000	876.000000	6307.033333	2610.000000	504.000000

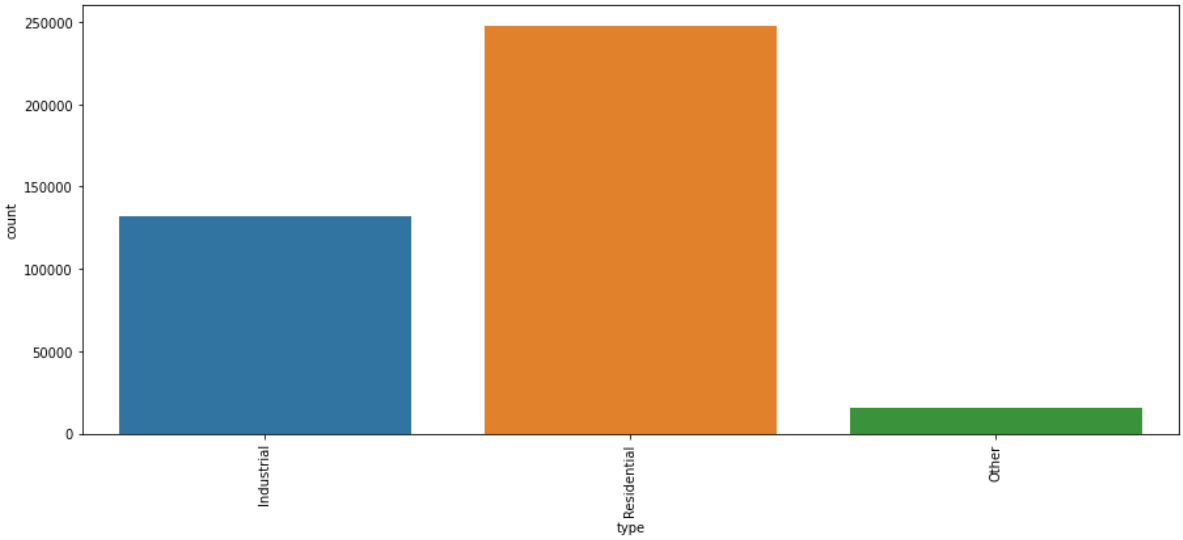
In [32]:
```
df3['so2'].fillna(df['so2'].mean(),inplace = True)
```

```
C:\Users\lenovo\AppData\Local\Temp\ipykernel_12812\2524380249.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df3['so2'].fillna(df['so2'].mean(),inplace = True)
```

In [33]:
```
df3.isnull().sum()
```

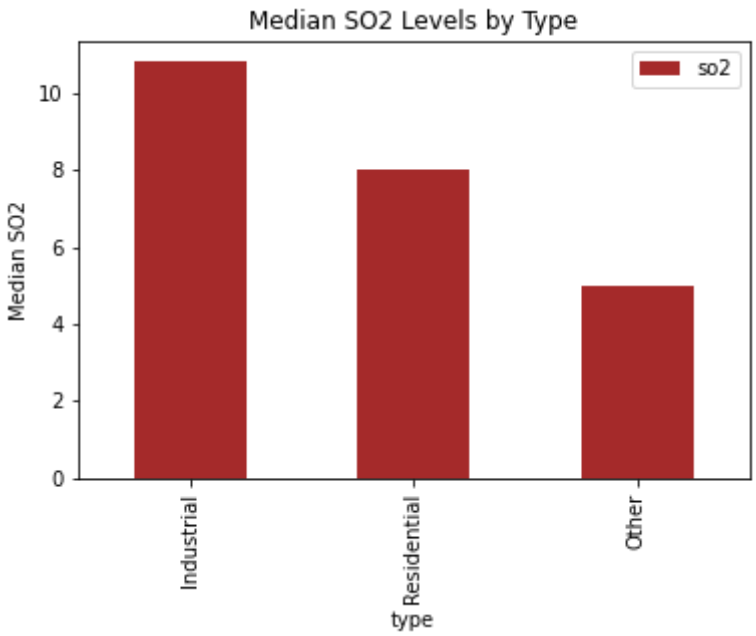
Out[33]:
```
state           0
location        0
type            0
so2             0
no2             0
rspm            0
spm        229052
pm2_5      386065
date            4
dtype: int64
```

In [34]:
```
plt.figure(figsize=(15,6))
sns.countplot(x= 'type',data = df3)
plt.xticks(rotation = 90)
plt.show()
```



In [35]:
```
df3[['so2', 'type']].groupby(['type']).median().sort_values('so2',ascending= False

plt.xlabel('type')
plt.ylabel('Median SO2')
plt.title('Median SO2 Levels by Type')
```

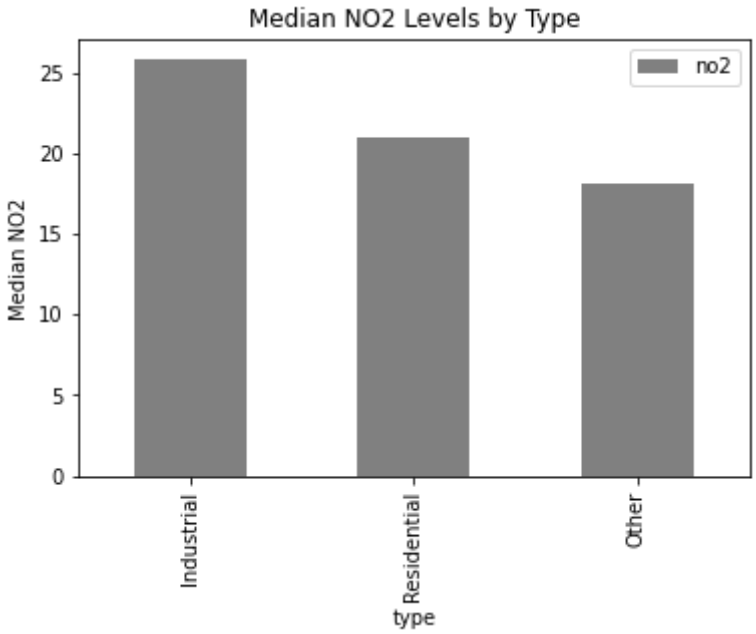
Out[35]:
```
Text(0.5, 1.0, 'Median SO2 Levels by Type')
```



In [36]:
```
df3[['no2', 'type']].groupby(['type']).median().sort_values('no2',ascending= False
```

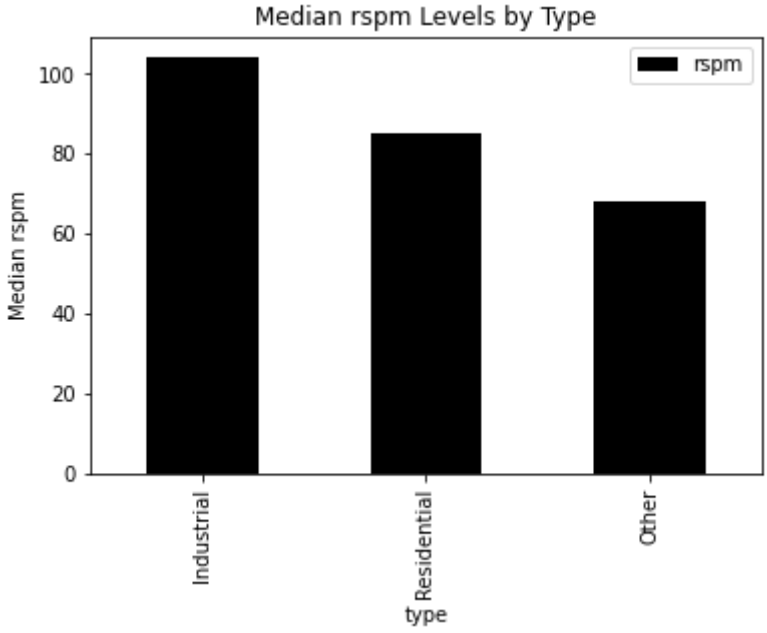
```
plt.xlabel('type')
plt.ylabel('Median NO2')
plt.title('Median NO2 Levels by Type')
```

Out[36]:
```
Text(0.5, 1.0, 'Median NO2 Levels by Type')
```



In [37]:
```
df3[['rspm', 'type']].groupby(['type']).median().sort_values('rspm',ascending= Fals

plt.xlabel('type')
plt.ylabel('Median rspm')
plt.title('Median rspm Levels by Type')
```

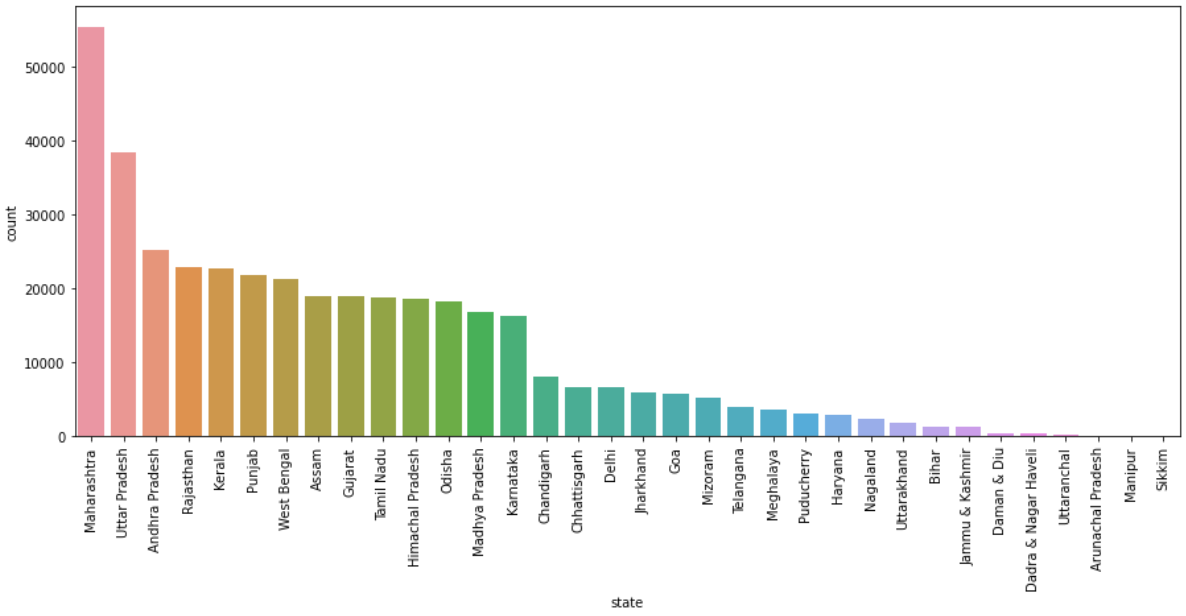
Out[37]:
```
Text(0.5, 1.0, 'Median rspm Levels by Type')
```



In [38]:
```
df3['state'].value_counts()
```

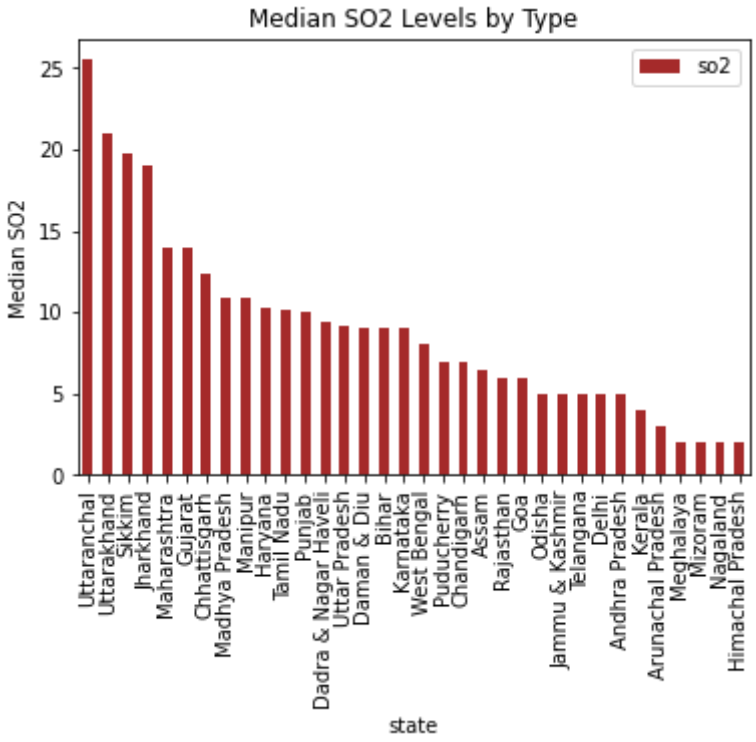
Out[38]:
```
Maharashtra              55439
Uttar Pradesh            38507
Andhra Pradesh           25228
Rajasthan                22954
Kerala                   22682
Punjab                   21808
West Bengal              21295
Assam                    19083
Gujarat                  18942
Tamil Nadu               18792
Himachal Pradesh         18625
Odisha                   18333
Madhya Pradesh           16874
Karnataka                16256
Chandigarh                8142
Chhattisgarh              6764
Delhi                     6667
Jharkhand                 5877
Goa                       5804
Mizoram                   5328
Telangana                 3976
Meghalaya                 3711
Puducherry                3032
Haryana                   2923
Nagaland                  2462
Uttarakhand               1917
Bihar                     1333
Jammu & Kashmir           1257
Daman & Diu                439
Dadra & Nagar Haveli       438
Uttaranchal                265
Arunachal Pradesh           89
Manipur                     76
Sikkim                       1
Name: state, dtype: int64
```

In [39]:
```
plt.figure(figsize=(15, 6))

sns.countplot(x='state', data=df3, order=df3['state'].value_counts().index)
plt.xticks(rotation=90)

plt.show()
```



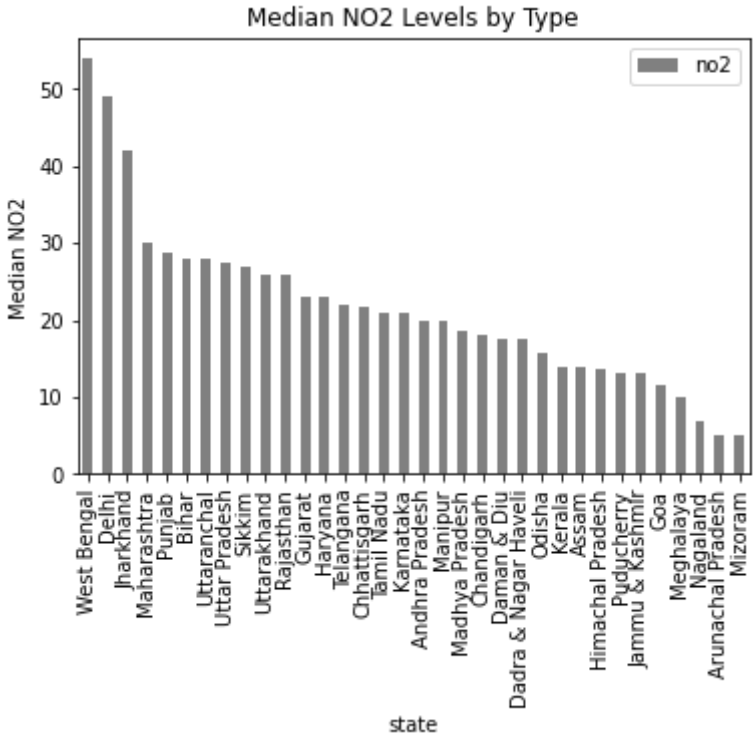
In [40]:
```
df3[['so2', 'state']].groupby(['state']).median().sort_values('so2',ascending= Fals

plt.xlabel('state')
plt.ylabel('Median SO2')
plt.title('Median SO2 Levels by Type')
```

Out[40]:
```
Text(0.5, 1.0, 'Median SO2 Levels by Type')
```



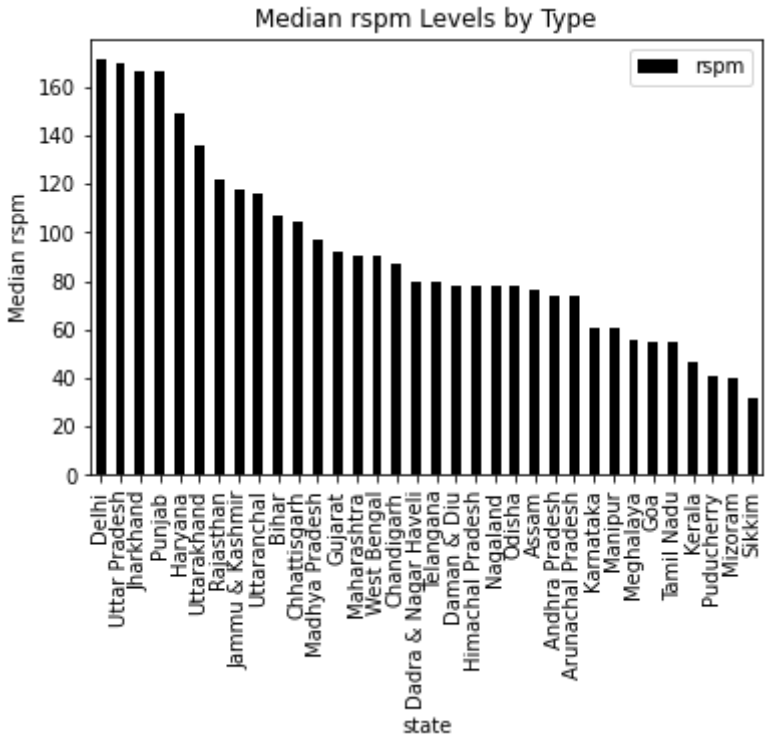
In [41]:
```
df3[['no2', 'state']].groupby(['state']).median().sort_values('no2',ascending= Fals

plt.xlabel('state')
plt.ylabel('Median NO2')
plt.title('Median NO2 Levels by Type')
```

Out[41]:
```
Text(0.5, 1.0, 'Median NO2 Levels by Type')
```



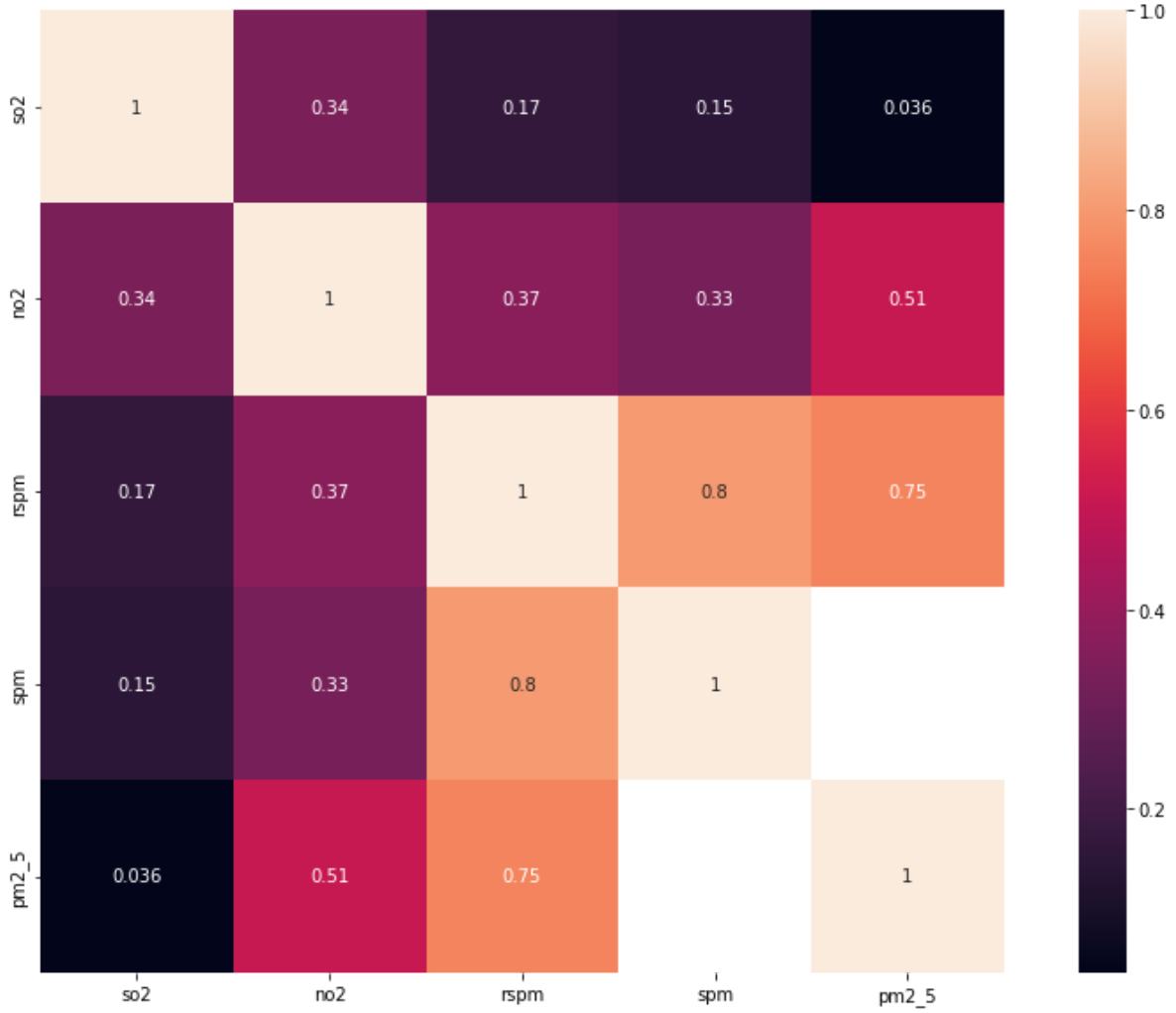
In [42]:
```
df3[['rspm', 'state']].groupby(['state']).median().sort_values('rspm',ascending= F

plt.xlabel('state')
plt.ylabel('Median rspm')
plt.title('Median rspm Levels by Type')
```

Out[42]:
```
Text(0.5, 1.0, 'Median rspm Levels by Type')
```



In [43]:
```
corrmat = df.corr()
f, ax = plt.subplots(figsize = (15, 10))
sns.heatmap(corrmat, vmax = 1, square = True, annot = True)
```

Out[43]:
```
<AxesSubplot:>
```


so2
no2
rspm
spm
pm2_5
1
0.34
0.17
0.15
0.036
0.34
1
0.37
0.33
0.51
0.17
0.37
1
0.8
0.75
0.15
0.33
0.8
1
0.036
0.51
0.75
1
so2
no2
rspm
spm
pm2_5
1.0
0.8
0.6
0.4
0.2

In [ ]: