NAME - VEDANT MAULEKHI ROLL NO. - 211156

1) COAL INDIA OPEN-PIT BLASTING

Problem statement :- Doing data analysis of Pollution Component data of Singrauli Mines specifically NO2 to study and predict the acid rains in surrounding area . We make use of various methods to do so :-

1. Curve fitting analysis of different pollutants and also studying about NO2 .
2. Forecasting and doing prediction analysis of NO2
3. Exploratory analysis of different pollutants and building heatmaps and correlation maps for pollutants .
4. Studying Blasting times of different pollutants and NO2 to study when the concentration is the highest .

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df = pd.read_csv('Open pit blasting 01-02-2023 000000 To 01-05-2023 235959.csv')
import pandas as pd
df.head(5)



df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8643 entries, 0 to 8642
Data columns (total 13 columns):
 #   Column                                                  Non-Null Count  Dtype
---  ------                                                  --------------  -----
 0   #                                                       8643 non-null   int64
 1   From                                                    8643 non-null   object
 2   To (Interval: 15M)                                      8640 non-null   object
 3   Singrauli, Surya Kiran Bhawan Dudhichua  PM10 (µg/m3)   6962 non-null   float64
 4   Singrauli, Surya Kiran Bhawan Dudhichua  PM2.5 (µg/m3)  8417 non-null   float64
 5   Singrauli, Surya Kiran Bhawan Dudhichua  NO (µg/m3)     7274 non-null   float64
 6   Singrauli, Surya Kiran Bhawan Dudhichua  NO2 (µg/m3)    8227 non-null   float64
 7   Singrauli, Surya Kiran Bhawan Dudhichua  NOX (ppb)      8228 non-null   float64
 8   Singrauli, Surya Kiran Bhawan Dudhichua  CO (mg/m3)     8147 non-null   float64
 9   Singrauli, Surya Kiran Bhawan Dudhichua  SO2 (µg/m3)    7192 non-null   float64
 10  Singrauli, Surya Kiran Bhawan Dudhichua  NH3 (µg/m3)    8317 non-null   float64
 11  Singrauli, Surya Kiran Bhawan Dudhichua  Ozone (µg/m3)  8190 non-null   float64
 12  Singrauli, Surya Kiran Bhawan Dudhichua  Benzene (µg/m3) 2448 non-null   float64
dtypes: float64(10), int64(1), object(2)
memory usage: 877.9+ KB
```

In [2]:
```
df = pd.read_csv('Open pit blasting 01-02-2023 000000 To 01-05-2023 235959.csv',index_col=0)
df.head()
```

Out[2]:

#	From	To (Interval: 15M)	Singrauli, Surya Kiran Bhawan Dudhichua PM10 (µg/m3)	Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (µg/m3)	Singrauli, Surya Kiran Bhawan Dudhichua NO (µg/m3)	Singrauli, Surya Kiran Bhawan Dudhichua NO2 (µg/m3)	Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb)	Singrauli, Surya Kiran Bhawan Dudhichua CO (mg/m3)	Singrauli, Surya Kiran Bhawan Dudhichua SO2 (µg/m3)	Singrauli, Surya Kiran Bhawan Dudhichua NH3 (µg/m3)	Singrauli, Surya Kiran Bhawan Dudhichua Ozone (µg/m3)	Singrauli, Surya Kiran Bhawan Dudhichua Benzene (µg/m3)
1	2023-02-01 00:00:00	2023-02-01 00:15:00	95.0	35.0	NaN	90.1	56.2	0.31	NaN	17.7	28.1	0.4
2	2023-02-01 00:15:00	2023-02-01 00:30:00	95.0	35.0	NaN	88.0	55.1	0.33	NaN	18.3	27.1	0.4
3	2023-02-01 00:30:00	2023-02-01 00:45:00	95.0	35.0	NaN	87.7	55.2	0.38	NaN	19.7	24.9	0.4
4	2023-02-01 00:45:00	2023-02-01 01:00:00	122.0	34.0	NaN	88.9	55.7	0.38	NaN	21.3	21.9	0.4
5	2023-02-01 01:00:00	2023-02-01 01:15:00	122.0	34.0	NaN	90.0	55.8	0.38	NaN	22.3	16.7	0.4

In [3]:
```
df.columns =  [ 'from', 'to', 'PM10', 'PM2.5', 'NO','NO2','NOX','CO','SO2','NH3','Ozone','Benzene' ]
df = df.drop('to', axis=1)
```

```
df.head()
```

Out[3]:

#	from	PM10	PM2.5	NO	NO2	NOX	CO	SO2	NH3	Ozone	Benzene
1	2023-02-01 00:00:00	95.0	35.0	NaN	90.1	56.2	0.31	NaN	17.7	28.1	0.4
2	2023-02-01 00:15:00	95.0	35.0	NaN	88.0	55.1	0.33	NaN	18.3	27.1	0.4
3	2023-02-01 00:30:00	95.0	35.0	NaN	87.7	55.2	0.38	NaN	19.7	24.9	0.4
4	2023-02-01 00:45:00	122.0	34.0	NaN	88.9	55.7	0.38	NaN	21.3	21.9	0.4
5	2023-02-01 01:00:00	122.0	34.0	NaN	90.0	55.8	0.38	NaN	22.3	16.7	0.4

In [4]:

```
df = df.iloc[:-3]
df.tail()
```

Out[4]:

#	from	PM10	PM2.5	NO	NO2	NOX	CO	SO2	NH3	Ozone	Benzene
8636	2023-05-01 22:45:00	19.0	11.0	17.9	100.0	67.8	0.63	10.0	10.7	26.1	0.1
8637	2023-05-01 23:00:00	19.0	11.0	17.9	100.0	67.7	0.57	10.0	10.4	30.9	0.1
8638	2023-05-01 23:15:00	19.0	11.0	19.6	100.2	69.2	0.58	9.9	10.5	29.6	0.1
8639	2023-05-01 23:30:00	19.0	11.0	20.8	100.2	70.2	0.58	9.5	10.8	30.0	0.1
8640	2023-05-01 23:45:00	32.0	6.0	21.8	98.8	70.3	NaN	NaN	11.0	33.5	0.1

In [5]:

```
date_format = '%Y-%m-%d %H:%M:%S'

# Convert the 'from' column to numerical datetime representation
df['from'] = pd.to_datetime(df['from'], format=date_format)
```

In [6]:

```
df.set_index('from', inplace=True)
df.head()
```

Out[6]:

from	PM10	PM2.5	NO	NO2	NOX	CO	SO2	NH3	Ozone	Benzene
2023-02-01 00:00:00	95.0	35.0	NaN	90.1	56.2	0.31	NaN	17.7	28.1	0.4
2023-02-01 00:15:00	95.0	35.0	NaN	88.0	55.1	0.33	NaN	18.3	27.1	0.4
2023-02-01 00:30:00	95.0	35.0	NaN	87.7	55.2	0.38	NaN	19.7	24.9	0.4
2023-02-01 00:45:00	122.0	34.0	NaN	88.9	55.7	0.38	NaN	21.3	21.9	0.4
2023-02-01 01:00:00	122.0	34.0	NaN	90.0	55.8	0.38	NaN	22.3	16.7	0.4

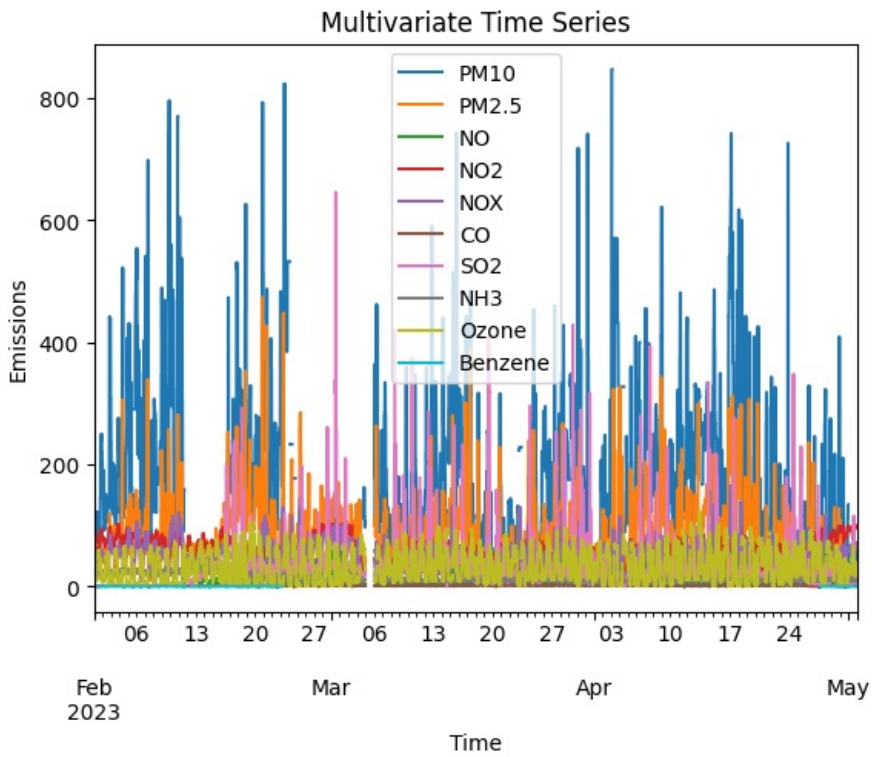
We draw a Multivariate Time series

In [7]:

```
df.plot()
plt.xlabel('Time')
plt.ylabel('Emissions')
plt.title('Multivariate Time Series')
```

Out[7]:

```
Text(0.5, 1.0, 'Multivariate Time Series')
```



Multivariate plot we can observe that some of the variables have significant concentration in the pollutants whereas some pollutants have comparatoively less concentration.

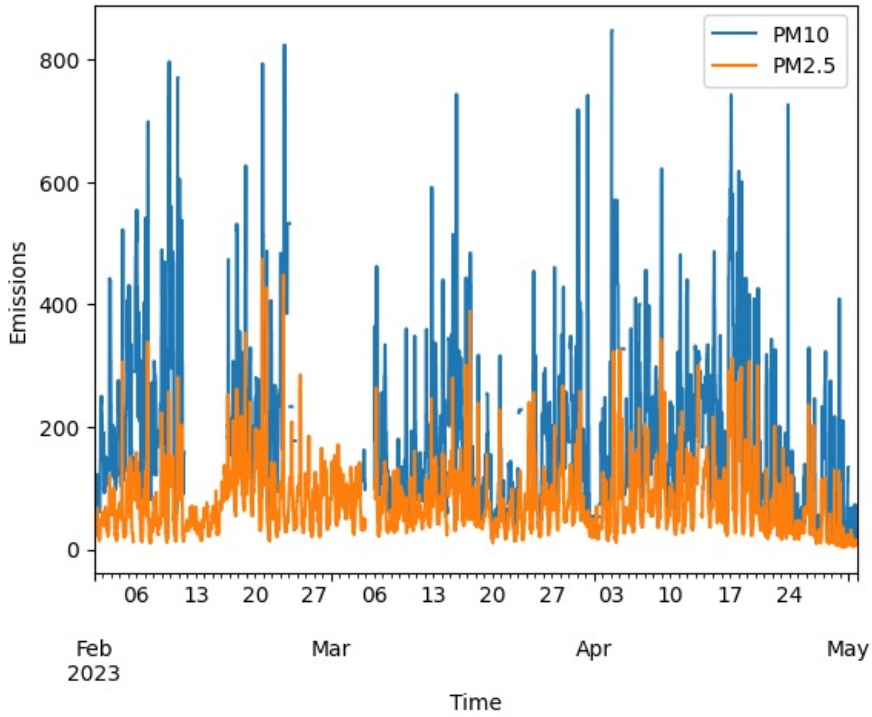
Compare different pollutants over the time series .

Using curve fitting to analyse different pollutants

PM10 AND PM2.5

In [8]:
```
columns_to_plot = ['PM10', 'PM2.5']

df[columns_to_plot].plot()
plt.xlabel('Time')
plt.ylabel('Emissions')
```

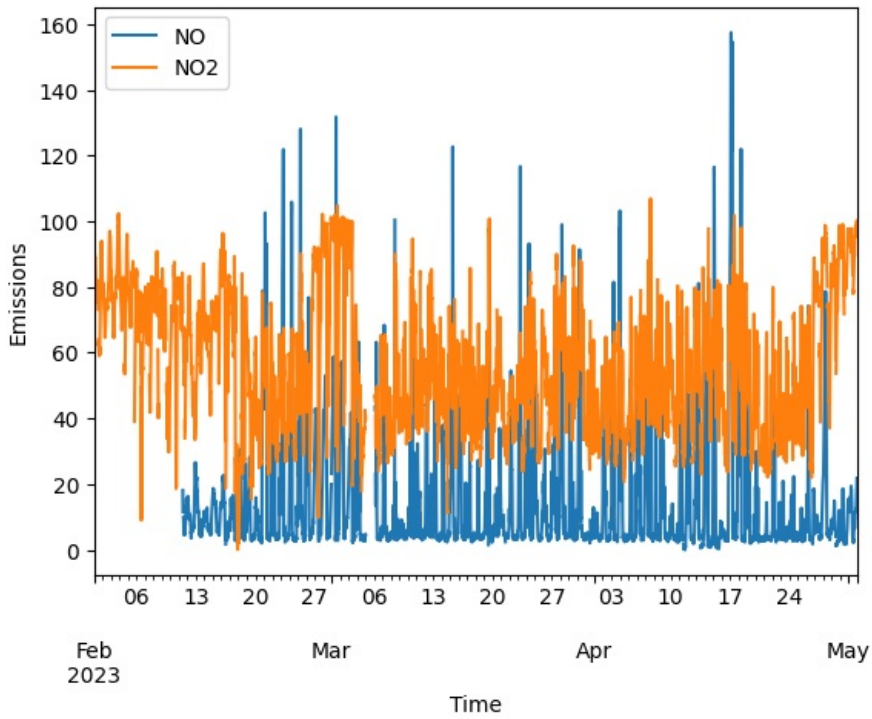
Out[8]:
```
Text(0, 0.5, 'Emissions')
```



NO AND NO2

In [9]:
```
columns_to_plot=['NO','NO2']

df[columns_to_plot].plot()
plt.xlabel('Time')
plt.ylabel('Emissions')
```

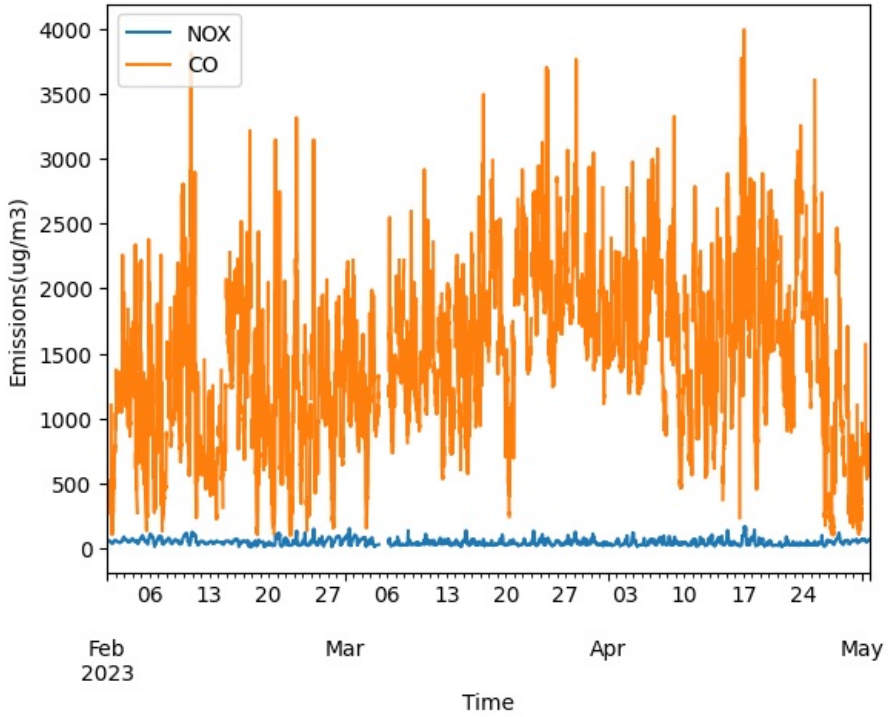
Out[9]: Text(0, 0.5, 'Emissions')



NOX AND CO

In [10]:
```
import pandas as pd

# Increase the values of a specific column by 1000 times
df['CO'] *= 1000


columns_to_plot=['NOX','CO']

df[columns_to_plot].plot()
plt.xlabel('Time')
plt.ylabel('Emissions(ug/m3)')
```

Out[10]: Text(0, 0.5, 'Emissions(ug/m3)')

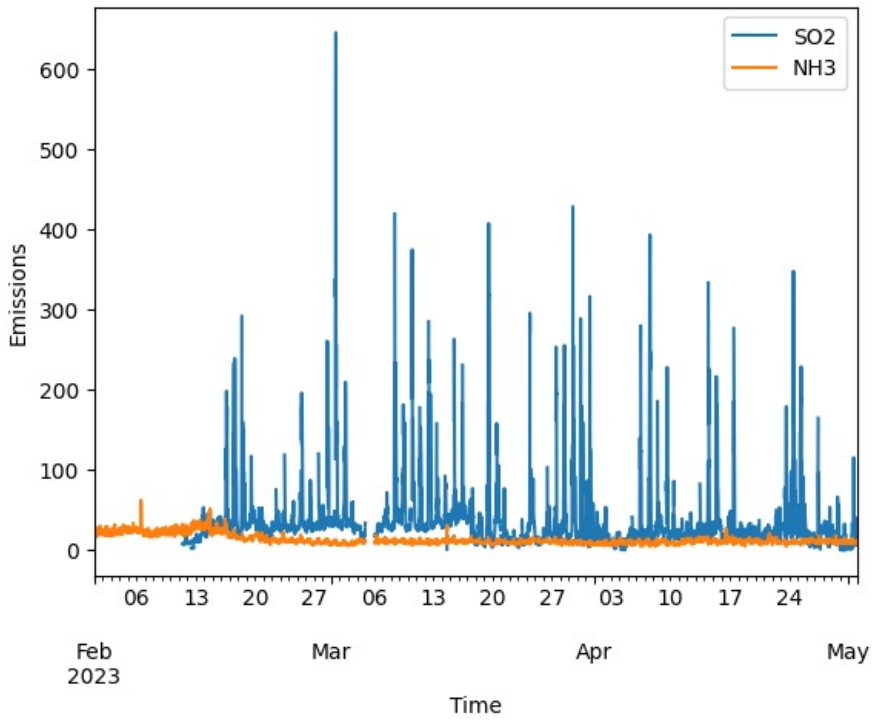


In [ ]:
```
NOX is not in less amount its just that CO is released in large amounts during coal blasting .
```

SO2 AND NH3

In [11]:
```
columns_to_plot=['SO2','NH3']

df[columns_to_plot].plot()
plt.xlabel('Time')
plt.ylabel('Emissions')
```

Out[11]: Text(0, 0.5, 'Emissions')

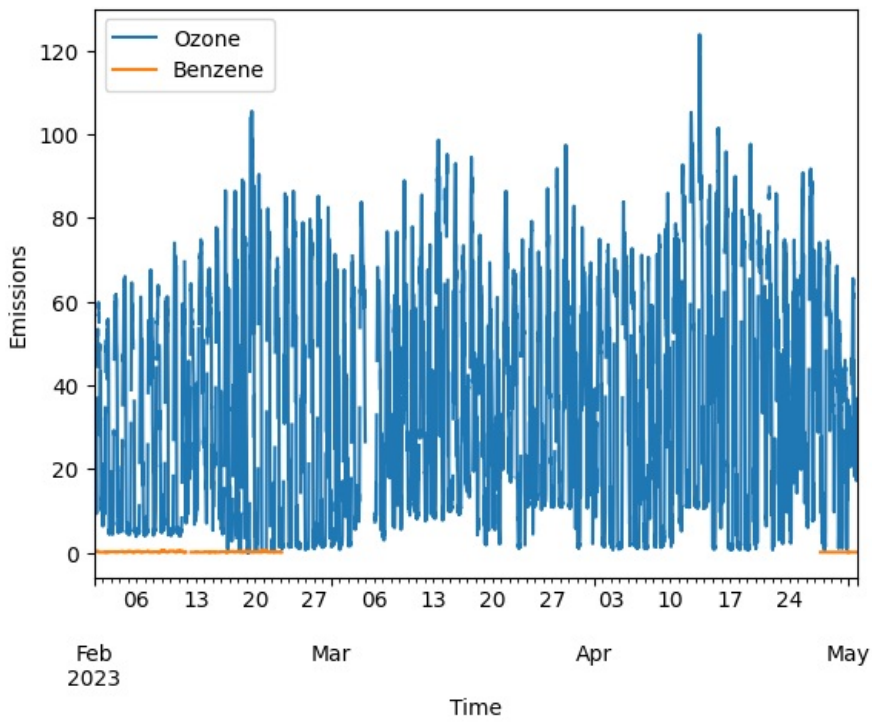


NH3 concentration is very less compared to rest gases.

OZONE AND BENZENE

In [12]:
```
columns_to_plot=['Ozone','Benzene']

df[columns_to_plot].plot()
plt.xlabel('Time')
plt.ylabel('Emissions')
```

Out[12]:
```
Text(0, 0.5, 'Emissions')
```



Benzene concentration is very less compared to rest gases .

In [13]:
```
!pip install pmdarima
import statsmodels.api as sm
import pandas.plotting
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
```

```
Requirement already satisfied: pmdarima in c:\users\vedant\appdata\local\programs\python\python310\lib\site-pac
kages (2.0.3)
Requirement already satisfied: joblib>=0.11 in c:\users\vedant\appdata\local\programs\python\python310\lib\site
-packages (from pmdarima) (1.2.0)
Requirement already satisfied: Cython!=0.29.18,!=0.29.31,>=0.29 in c:\users\vedant\appdata\local\programs\pytho
n\python310\lib\site-packages (from pmdarima) (0.29.35)
Requirement already satisfied: numpy>=1.21.2 in c:\users\vedant\appdata\local\programs\python\python310\lib\sit
e-packages (from pmdarima) (1.25.0)
Requirement already satisfied: pandas>=0.19 in c:\users\vedant\appdata\local\programs\python\python310\lib\site
-packages (from pmdarima) (2.0.2)
Requirement already satisfied: scikit-learn>=0.22 in c:\users\vedant\appdata\local\programs\python\python310\li
b\site-packages (from pmdarima) (1.2.2)
Requirement already satisfied: scipy>=1.3.2 in c:\users\vedant\appdata\local\programs\python\python310\lib\site
-packages (from pmdarima) (1.10.1)
Requirement already satisfied: statsmodels>=0.13.2 in c:\users\vedant\appdata\local\programs\python\python310\l
ib\site-packages (from pmdarima) (0.14.0)
Requirement already satisfied: urllib3 in c:\users\vedant\appdata\local\programs\python\python310\lib\site-pack
ages (from pmdarima) (2.0.3)
Requirement already satisfied: setuptools!=50.0.0,>=38.6.0 in c:\users\vedant\appdata\local\programs\python\pyt
hon310\lib\site-packages (from pmdarima) (58.1.0)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\vedant\appdata\local\programs\python\python31
0\lib\site-packages (from pandas>=0.19->pmdarima) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\vedant\appdata\local\programs\python\python310\lib\site
-packages (from pandas>=0.19->pmdarima) (2023.3)
Requirement already satisfied: tzdata>=2022.1 in c:\users\vedant\appdata\local\programs\python\python310\lib\si
te-packages (from pandas>=0.19->pmdarima) (2023.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\vedant\appdata\local\programs\python\python310\
lib\site-packages (from scikit-learn>=0.22->pmdarima) (3.1.0)
Requirement already satisfied: patsy>=0.5.2 in c:\users\vedant\appdata\local\programs\python\python310\lib\site
-packages (from statsmodels>=0.13.2->pmdarima) (0.5.3)
Requirement already satisfied: packaging>=21.3 in c:\users\vedant\appdata\local\programs\python\python310\lib\s
ite-packages (from statsmodels>=0.13.2->pmdarima) (23.0)
Requirement already satisfied: six in c:\users\vedant\appdata\local\programs\python\python310\lib\site-packages
(from patsy>=0.5.2->statsmodels>=0.13.2->pmdarima) (1.16.0)
```

Data Analysis

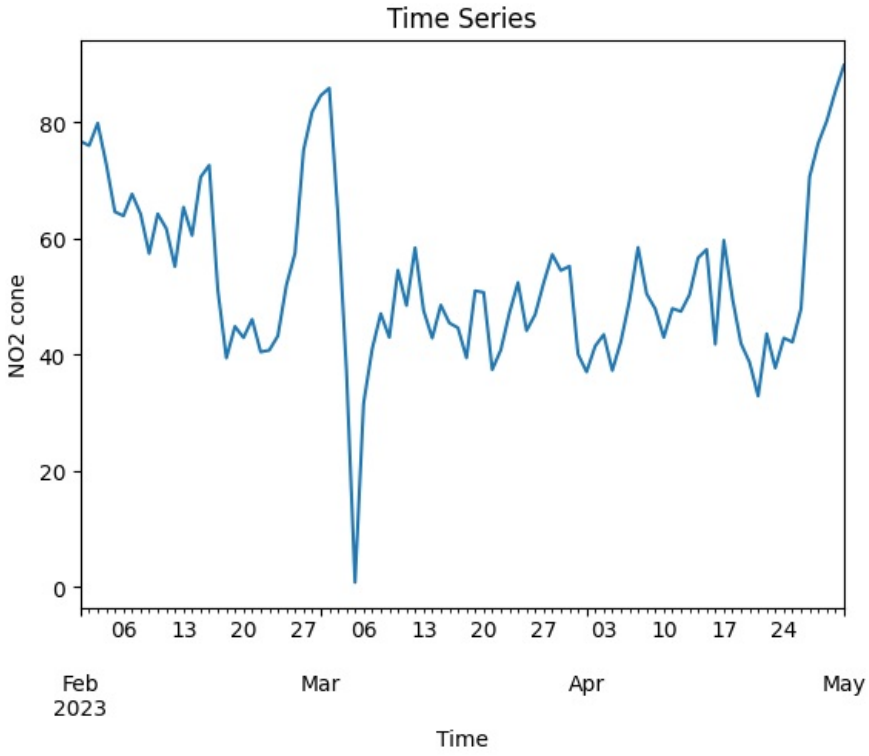
Since we have some null values in the dataset , we need to find a way to replace them , we are going to try three methods:-

1. Replacing with zeroes
2. Replacing with mean
3. Interpolation

We start with NO2 analysis

1) Replacing with zeroes

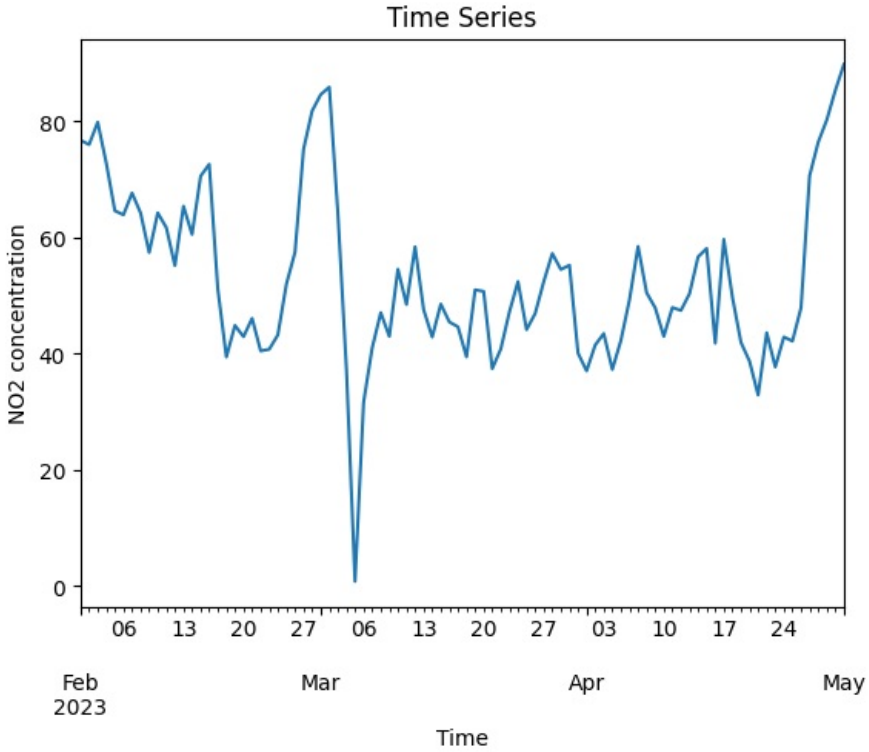
In [14]:
```
f1 = df['NO2'].copy()
f1.fillna(0, inplace=True)
f1.index = pd.to_datetime(f1.index)  # Convert index to datetime if it's not already
f1 = f1.resample('D').mean()
f1.plot()
plt.xlabel('Time')
plt.ylabel('NO2 cone')
plt.title('Time Series')
plt.show()
f1.head()
```



Out[14]:
```
from
2023-02-01    76.706250
2023-02-02    75.983333
2023-02-03    79.833333
2023-02-04    72.782292
2023-02-05    64.594792
Freq: D, Name: NO2, dtype: float64
```

In [ ]:

In [15]:
```
f1 = df['NO2'].copy()

f1.fillna(0, inplace=True)
f1 = f1.resample('D').mean()
f1.plot()
plt.xlabel('Time')
plt.ylabel('NO2 concentration')
plt.title('Time Series')
plt.show()
f1.head()
```



Out[15]:
```
from
2023-02-01    76.706250

2023-02-02    75.983333
2023-02-03    79.833333
2023-02-04    72.782292
2023-02-05    64.594792
Freq: D, Name: NO2, dtype: float64
```

In [16]:
```
f1
```

Out[16]:
```
from
2023-02-01    76.706250
2023-02-02    75.983333
2023-02-03    79.833333
2023-02-04    72.782292
2023-02-05    64.594792
                 ...
2023-04-27    70.600000
2023-04-28    76.368750
2023-04-29    80.210417
2023-04-30    85.304167
2023-05-01    89.779167
Freq: D, Name: NO2, Length: 90, dtype: float64
```

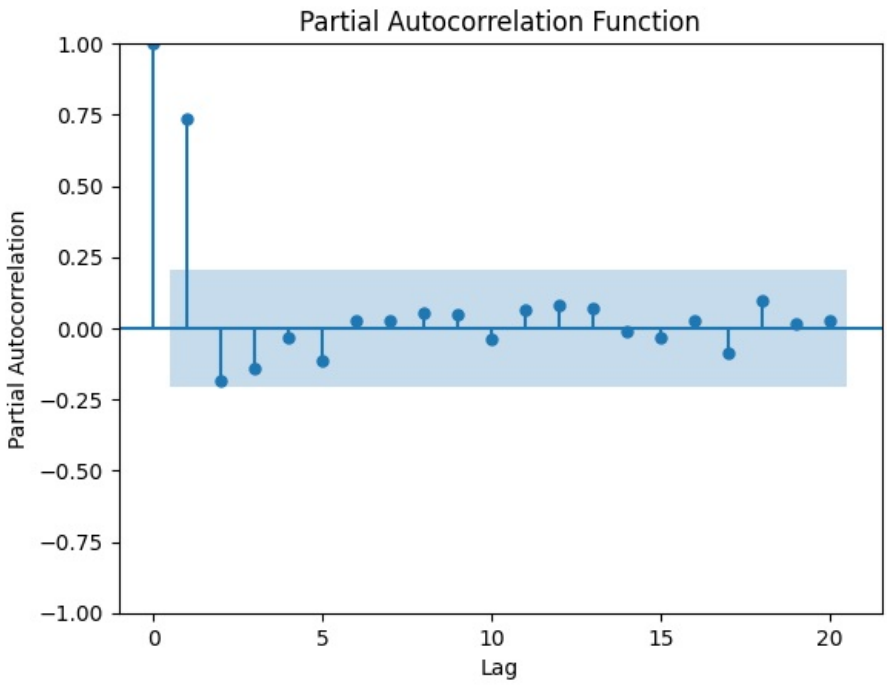
In [17]:
```
from statsmodels.tsa.stattools import adfuller

adf_test = adfuller(f1)
print(f'p-value: {adf_test[1]}')
```

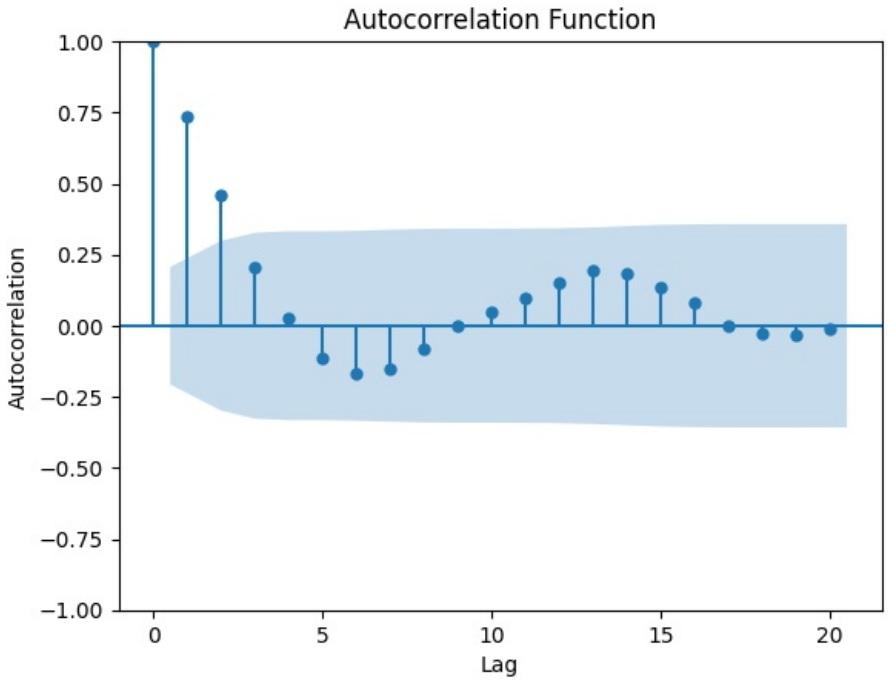
```
p-value: 0.002564843072917356
```

Since we get such a low p value , we see stationarity in time series data.

Now we try to make an ARIMA model for NO2

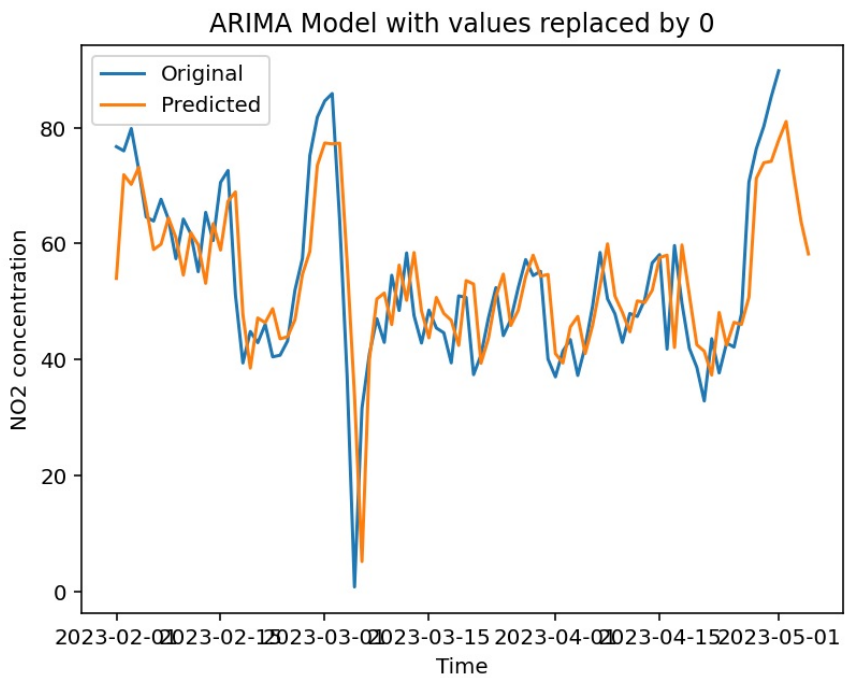
In [18]:
```
from statsmodels.graphics.tsaplots import plot_pacf, plot_acf

# Plot PACF
plot_pacf(f1)
plt.xlabel('Lag')
plt.ylabel('Partial Autocorrelation')
plt.title('Partial Autocorrelation Function')
plt.show()

# Plot ACF
plot_acf(f1)
plt.xlabel('Lag')
plt.ylabel('Autocorrelation')
plt.title('Autocorrelation Function')
plt.show()
```





In [ ]:

In [57]:
```
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt

%matplotlib notebook

data_series = f1
data_series = pd.Series(data_series)

model = sm.tsa.ARIMA(data_series, order=(3, 0, 0))
result = model.fit()

predictions = result.predict(start='2023-02-01', end='2023-05-05')

plt.plot(data_series, label='Original')
plt.plot(predictions, label='Predicted')
plt.xlabel('Time')
plt.ylabel('NO2 concentration')
plt.title('ARIMA Model with values replaced by 0')
plt.legend()
plt.show()
```



2) Replacing with mean

2) Replacing with mean

In [20]:
```
f2= df['NO2'].copy()

f2.fillna(14.65, inplace=True)
f2
```

Out[20]:
```
from
2023-02-01 00:00:00     90.1
2023-02-01 00:15:00     88.0
2023-02-01 00:30:00     87.7
2023-02-01 00:45:00     88.9
2023-02-01 01:00:00     90.0
                       ...
2023-05-01 22:45:00    100.0
2023-05-01 23:00:00    100.0
2023-05-01 23:15:00    100.2
2023-05-01 23:30:00    100.2
2023-05-01 23:45:00     98.8
Name: NO2, Length: 8640, dtype: float64
```

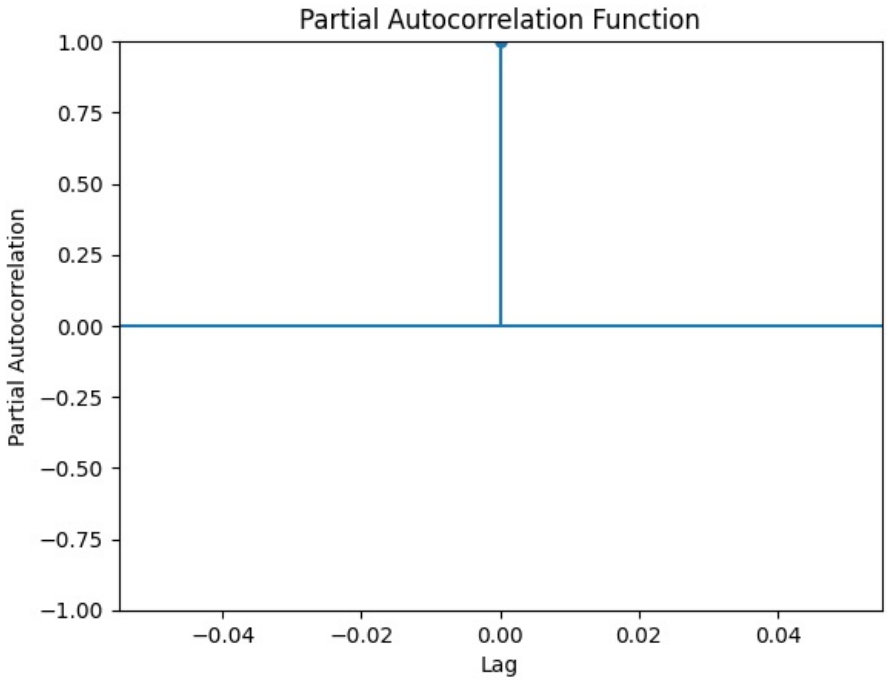
In [21]:
```
f2=f2.resample('D').mean()
```

In [22]:
```
from statsmodels.tsa.stattools import adfuller

adf_test = adfuller(f2)
print(f'p-value: {adf_test[1]}')
```

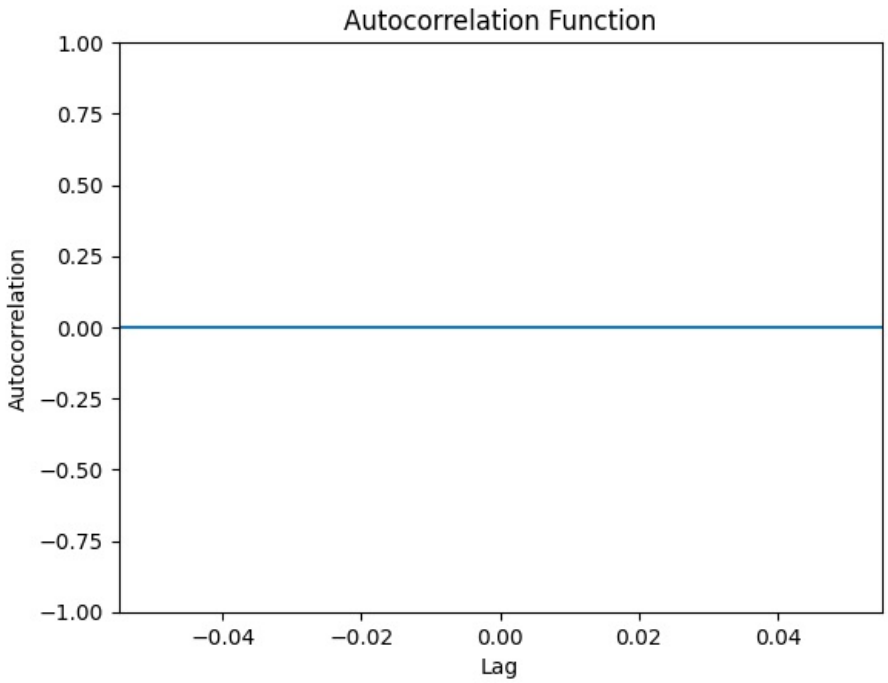
```
p-value: 0.004572055440975622
```

In [ ]:
```
very low p value indicates stationarity
```

In [64]:
```
from statsmodels.graphics.tsaplots import plot_pacf, plot_acf

# Plot PACF
plot_pacf(f2)
plt.xlabel('Lag')
plt.ylabel('Partial Autocorrelation')
plt.title('Partial Autocorrelation Function')
plt.show()

# Plot ACF
plot_acf(f2)
plt.xlabel('Lag')
plt.ylabel('Autocorrelation')
plt.title('Autocorrelation Function')
plt.show()
```



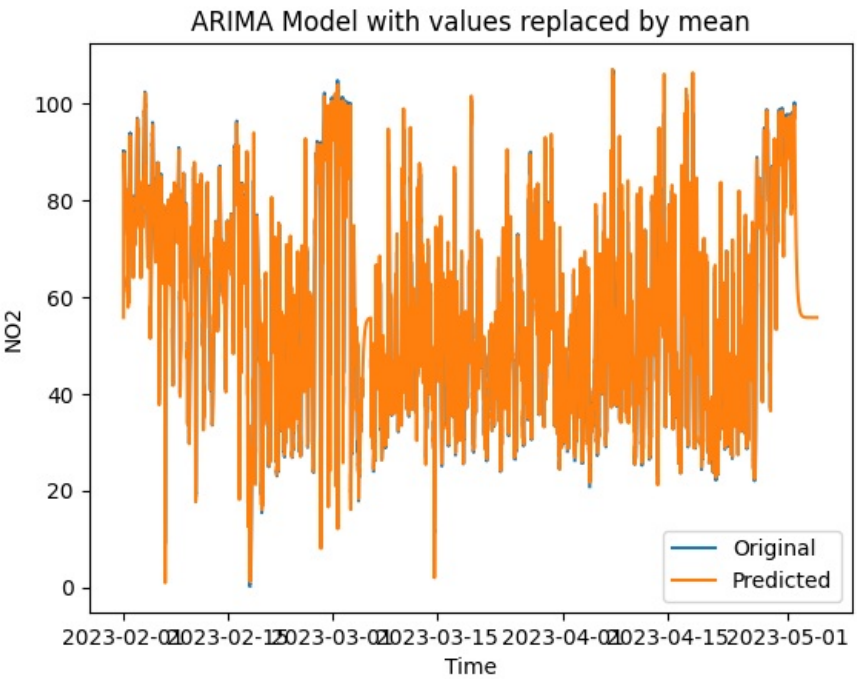

```
C:\Users\Vedant\AppData\Local\Programs\Python\Python310\lib\site-packages\matplotlib\axes\_base.py:2503: UserWa
rning: Warning: converting a masked element to nan.
  xys = np.asarray(xys)
```



In [65]:

```
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt

%matplotlib inline

f2= df['NO2'].copy()
data_series = f2
data_series = pd.Series(data_series)

model = sm.tsa.ARIMA(data_series, order=(2, 0, 0))
result = model.fit()

predictions = result.predict(start='2023-02-01', end='2023-05-05')

plt.plot(data_series, label='Original')
plt.plot(predictions, label='Predicted')
plt.xlabel('Time')
plt.ylabel('NO2')
plt.title('ARIMA Model with values replaced by mean')
plt.legend()
plt.show()
```

```
C:\Users\Vedant\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmodels\tsa\base\tsa_model.py:473
: ValueWarning: No frequency information was provided, so inferred frequency 15T will be used.
  self._init_dates(dates, freq)
C:\Users\Vedant\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmodels\tsa\base\tsa_model.py:473
: ValueWarning: No frequency information was provided, so inferred frequency 15T will be used.
  self._init_dates(dates, freq)
C:\Users\Vedant\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmodels\tsa\base\tsa_model.py:473
: ValueWarning: No frequency information was provided, so inferred frequency 15T will be used.
  self._init_dates(dates, freq)
```



3) Interpolation

1. Linear Interpolation:

Linear interpolation is a simple and commonly used method for filling missing data points. In this method, missing values are estimated by assuming a linear relationship between adjacent data points. The missing value is calculated by taking the average of the values at the nearest available points. Linear interpolation assumes a constant rate of change between data points.

2. Polynomial Interpolation:

Polynomial interpolation is a more advanced interpolation method that uses polynomial functions to estimate missing values. It fits a polynomial curve through the available data points and uses the curve to estimate the missing values. The degree of the polynomial determines the complexity of the interpolation. Higher-degree polynomials can closely fit the data but may be prone to overfitting.

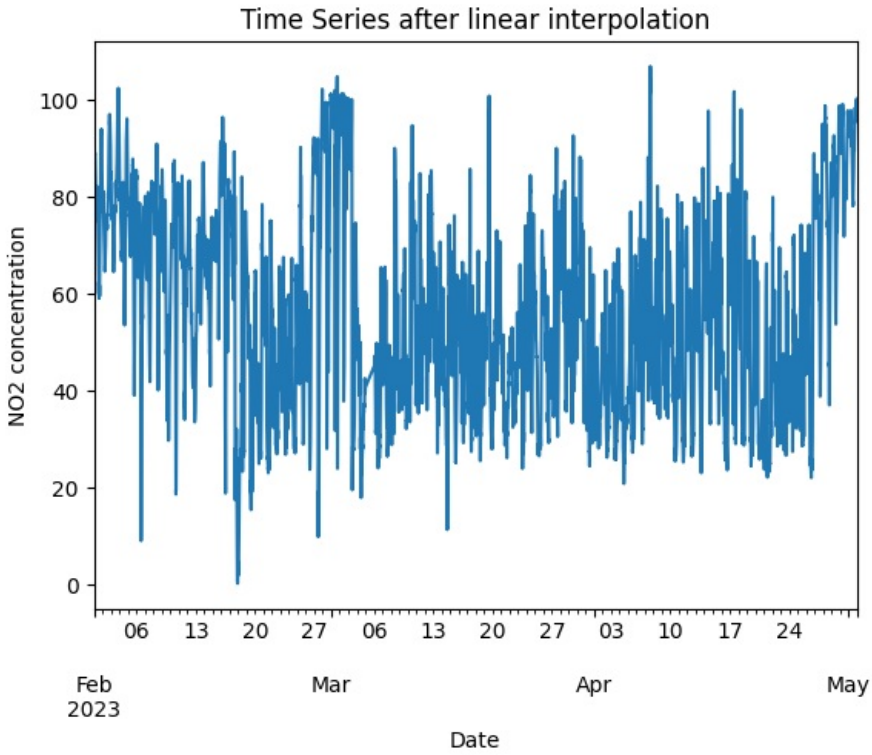
3. Spline Interpolation:

Spline interpolation is a smooth interpolation method that uses piecewise-defined polynomial functions called splines. It fits a separate polynomial curve between each pair of adjacent data points, ensuring smoothness at the connecting points. Spline interpolation can provide more accurate results than linear or polynomial interpolation, especially when the data has complex patterns or abrupt changes.

1) Linear Interpolation

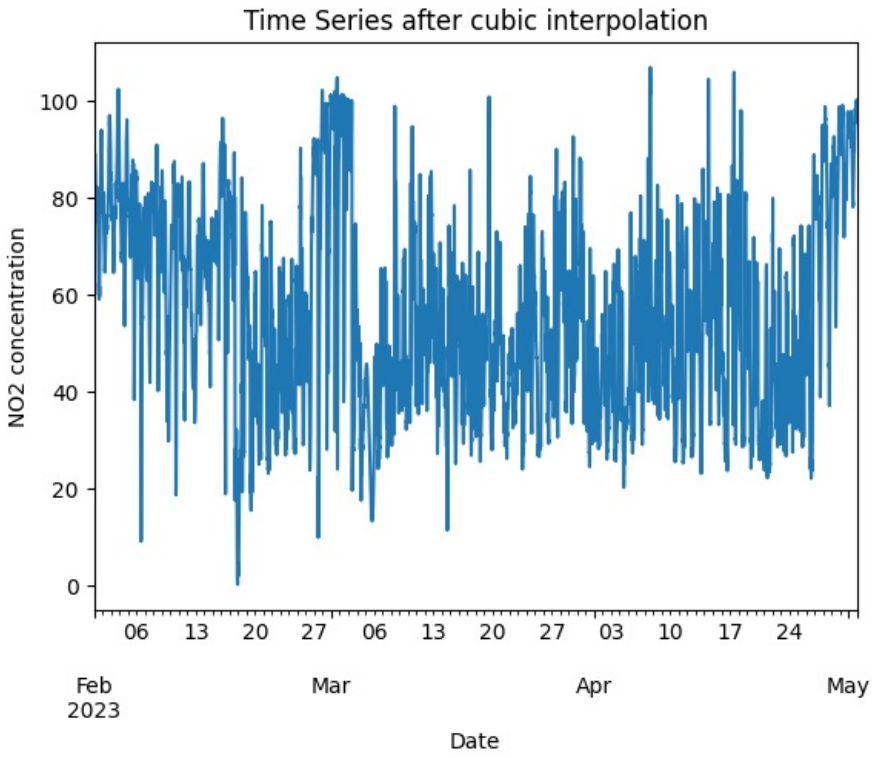
In [66]:
```
# Create a copy of the 'NO' column
d1 = df['NO2'].copy()

# Resample the time series
# s3 = s3.resample('D')

# Perform interpolation
d1 = d1.interpolate()
```

```
# Fill remaining missing values using forward fill and backward fill
d1.fillna(method='ffill', inplace=True)
d1.fillna(method='bfill', inplace=True)

# Plot the interpolated and filled time series
d1.plot()
plt.xlabel('Date')
plt.ylabel('NO2 concentration')
plt.title('Time Series after linear interpolation')

# Show the plot
plt.show()
```



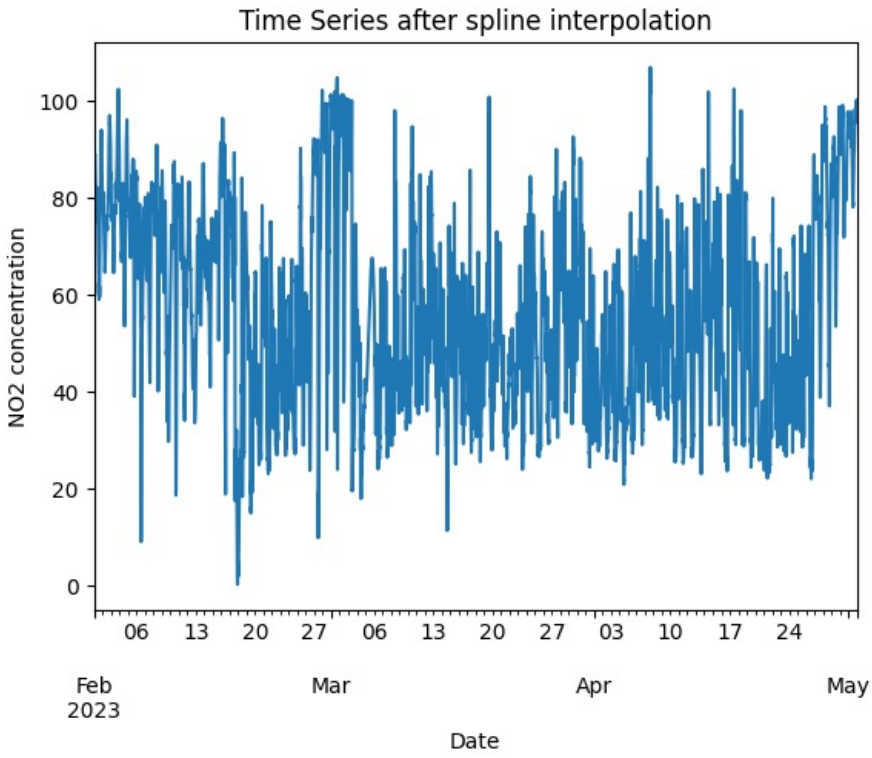
2) Cubic interpolation

In [67]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Create a copy of the 'NO' column
d2 = df['NO2'].copy()

# Resample the time series to a different frequency (e.g., from hourly to daily)
# s4 = s4.resample('D')

# Perform cubic interpolation
d2 = d2.interpolate(method='cubic')

# Fill remaining missing values using forward fill and backward fill
d2.fillna(method='ffill', inplace=True)
d2.fillna(method='bfill', inplace=True)

# Plot the interpolated and filled time series
d2.plot()
plt.xlabel('Date')
plt.ylabel('NO2 concentration')
plt.title('Time Series after cubic interpolation')

# Show the plot
plt.show()
```



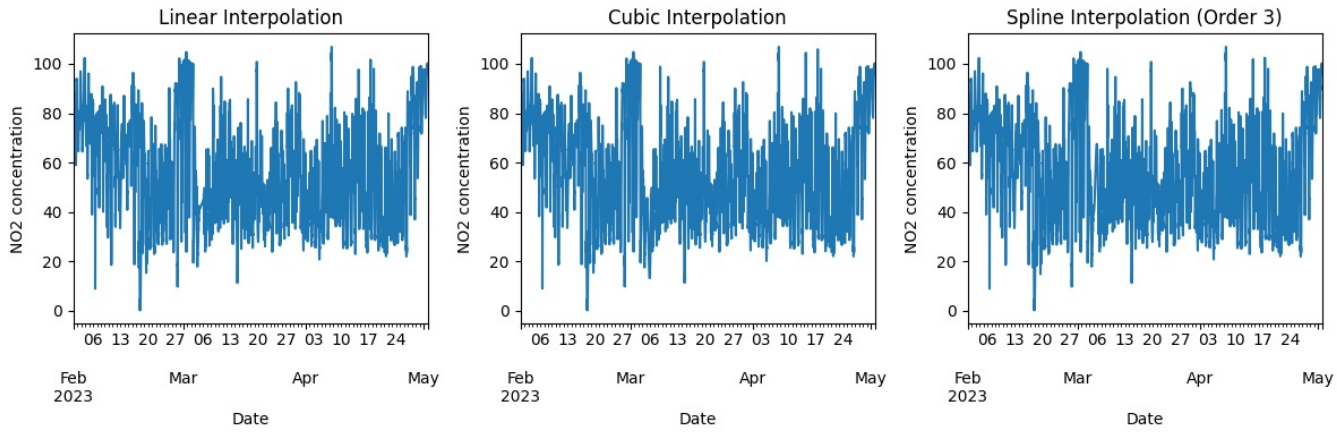
3) Spline Interpolation

In [68]:
```
# Create  copy of the 'NO' column
d3 = df['NO2'].copy()

# Resample the time series to a different frequency (e.g., from hourly to daily)
# sS = sS.resample('D')

# Perform spline interpolation with order 3
d3 = d3.interpolate(method='spline', order=3)
# Fill remaining missing values using forward fill and backward fill
d3.fillna(method='ffill', inplace=True)
d3.fillna(method='bfill', inplace=True)

# Plot the interpolated and filled time series
d3.plot()
plt.xlabel('Date')
plt.ylabel('NO2 concentration')
plt.title('Time Series after spline interpolation')

# Show the plot
plt.show()
```



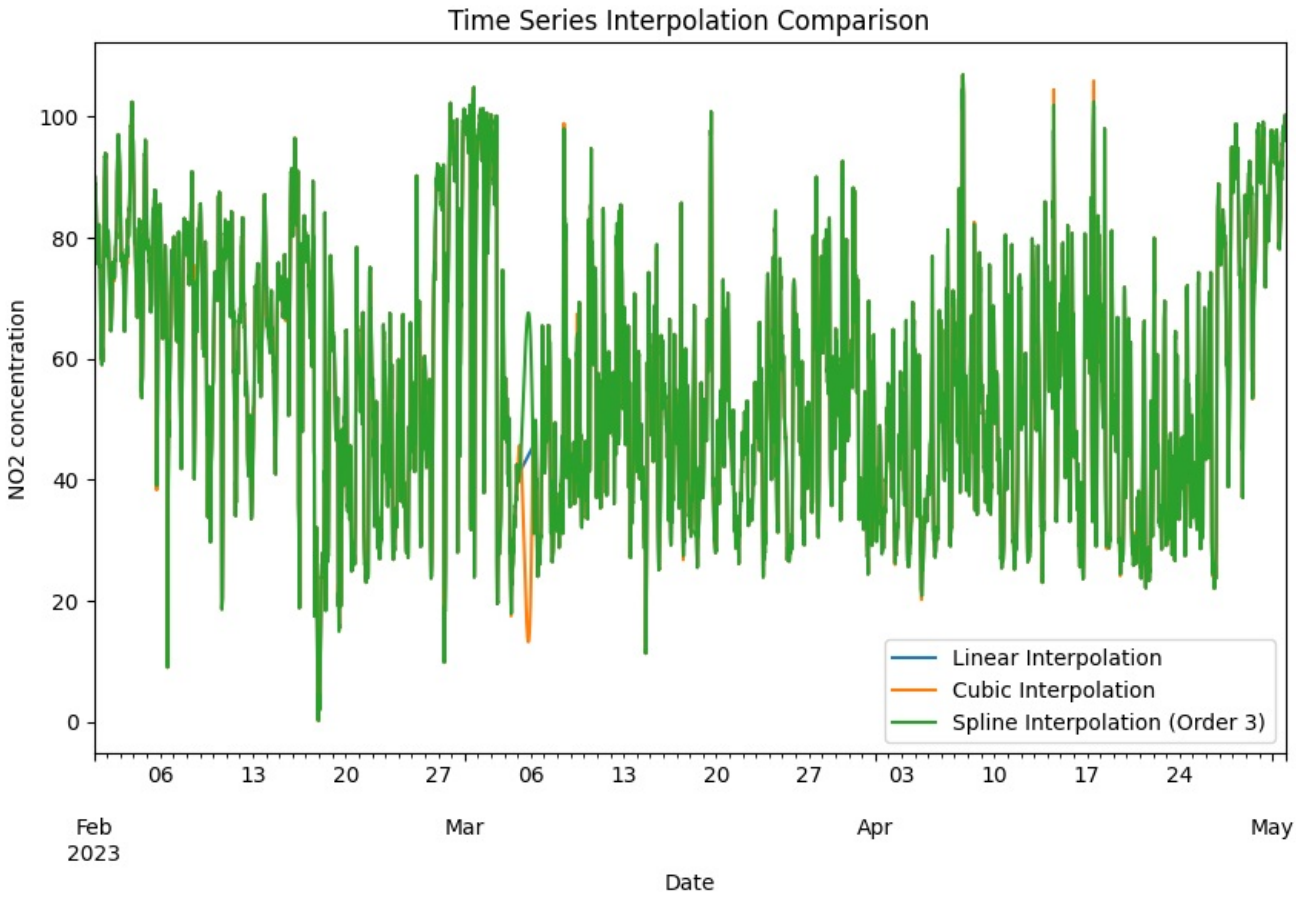
Now we compare all the interpolations together to find which is the best.

In [69]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Create a copy of the 'NO2' column
d1 = df['NO2'].copy()
d2 = df['NO2'].copy()
d3 = df['NO2'].copy()

# Perform linear interpolation
d1_interpolated = d1.interpolate(method='linear')
d2_interpolated = d2.interpolate(method='cubic')
d3_interpolated = d3.interpolate(method='spline', order=3)

# Fill remaining missing values using forward fill and backward fill
d1.fillna(method='ffill', inplace=True)
d1.fillna(method='bfill', inplace=True)
d2.fillna(method='ffill', inplace=True)
d2.fillna(method='bfill', inplace=True)
d3.fillna(method='ffill', inplace=True)
d3.fillna(method='bfill', inplace=True)

# Create subplots
fig, axes = plt.subplots(nrows=1, ncols=3, figsize=(12, 4))

# Plot the interpolated and filled time series on each subplot
d1_interpolated.plot(ax=axes[0])
axes[0].set_title('Linear Interpolation')

d2_interpolated.plot(ax=axes[1])
axes[1].set_title('Cubic Interpolation')

d3_interpolated.plot(ax=axes[2])
axes[2].set_title('Spline Interpolation (Order 3)')

# Set common labels for x-axis and y-axis
for ax in axes:
    ax.set_xlabel('Date')
    ax.set_ylabel('NO2 concentration')

# Adjust the layout to prevent overlapping of subplots
plt.tight_layout()

# Show the plot
plt.show()
```



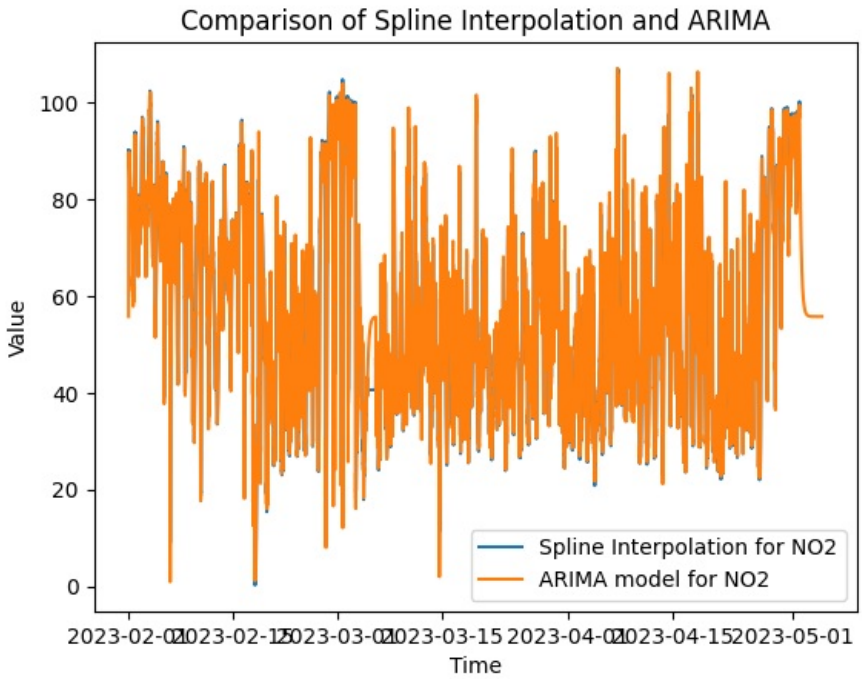
In [70]:

```
import pandas as pd
import matplotlib.pyplot as plt

# Create a copy of the 'NO2' column
d1 = df['NO2'].copy()
d2 = df['NO2'].copy()
d3 = df['NO2'].copy()

# Perform linear interpolation
d1_interpolated = d1.interpolate(method='linear')
d2_interpolated = d2.interpolate(method='cubic')
d3_interpolated = d3.interpolate(method='spline', order=3)

# Fill remaining missing values using forward fill and backward fill
d1.fillna(method='ffill', inplace=True)
d1.fillna(method='bfill', inplace=True)
d2.fillna(method='ffill', inplace=True)
d2.fillna(method='bfill', inplace=True)
d3.fillna(method='ffill', inplace=True)
d3.fillna(method='bfill', inplace=True)

# Create a new figure and axes
fig, ax = plt.subplots(figsize=(10, 6))

# Plot the interpolated and filled time series on the same axes
d1_interpolated.plot(ax=ax, label='Linear Interpolation')
d2_interpolated.plot(ax=ax, label='Cubic Interpolation')
d3_interpolated.plot(ax=ax, label='Spline Interpolation (Order 3)')

# Set labels and title
ax.set_xlabel('Date')
ax.set_ylabel('NO2 concentration')
ax.set_title('Time Series Interpolation Comparison')

# Display a legend
ax.legend()

# Show the plot
plt.show()
```



Now we do a comaprison between the ARIMA and interpolation

In [71]:
```
import matplotlib.pyplot as plt

# Plot the Spline Interpolation (Order 3) data
plt.plot(d3, label='Spline Interpolation for NO2')

# Plot the ARIMA model predictions for NO2
plt.plot(predictions, label='ARIMA model for NO2')

# Set x and y labels, and plot title
plt.xlabel('Time')
plt.ylabel('Value')
plt.title('Comparison of Spline Interpolation and ARIMA')

# Add a legend
plt.legend()

# Display the plot
plt.show()
```

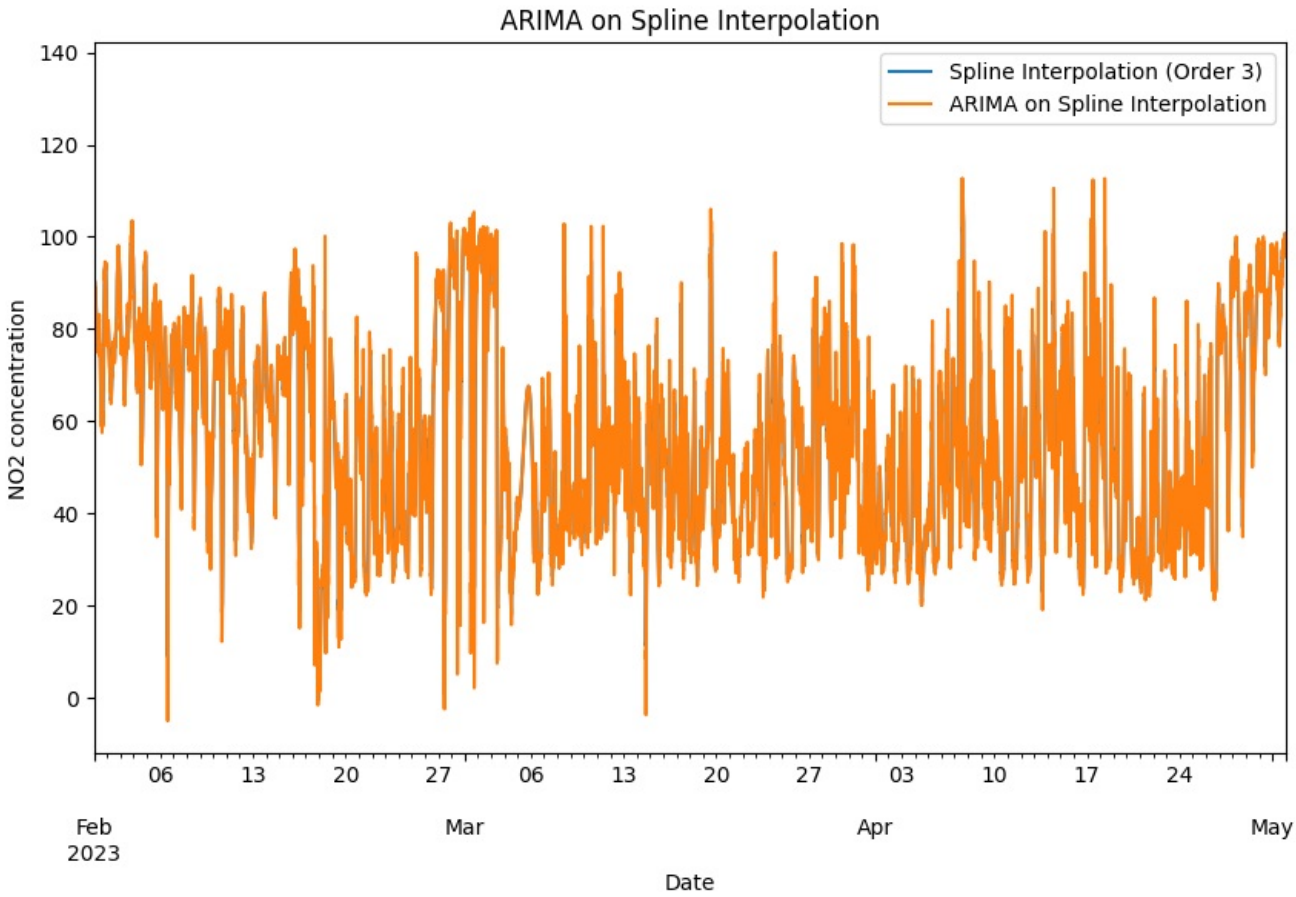


PLOT OF ARIMA ON SPLINE INTERPOLATION

Very low p-value indicates stationarity

In [72]:
```
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm

# Perform spline interpolation on the 'NO2' column
interpolated = df['NO2'].interpolate(method='spline', order=3)

# Fit ARIMA model on the interpolated data
model = sm.tsa.ARIMA(interpolated, order=(3, 2, 0))
arima_results = model.fit()

# Generate ARIMA predictions
arima_predictions = arima_results.predict(start=interpolated.index[0], end=interpolated.index[-1])

# Create a new figure and axes
fig, ax = plt.subplots(figsize=(10, 6))

# Plot the interpolated data
interpolated.plot(ax=ax, label='Spline Interpolation (Order 3)')

# Plot the ARIMA predictions
arima_predictions.plot(ax=ax, label='ARIMA on Spline Interpolation')

# Set labels and title
ax.set_xlabel('Date')
ax.set_ylabel('NO2 concentration')
ax.set_title('ARIMA on Spline Interpolation')

# Display a legend
ax.legend()

# Show the plot
plt.show()
```

```
C:\Users\Vedant\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmodels\tsa\base\tsa_model.py:473
: ValueWarning: No frequency information was provided, so inferred frequency 15T will be used.
  self._init_dates(dates, freq)
C:\Users\Vedant\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmodels\tsa\base\tsa_model.py:473
: ValueWarning: No frequency information was provided, so inferred frequency 15T will be used.
  self._init_dates(dates, freq)
C:\Users\Vedant\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmodels\tsa\base\tsa_model.py:473
: ValueWarning: No frequency information was provided, so inferred frequency 15T will be used.
  self._init_dates(dates, freq)
```



FORECASTING to study NO2 concentrations

Analysis for NO2 data

We will use arima model to predict the furute trends.

In [73]:
```
df_2=df.copy()
```

We will use interpolation to fill the missing values in the data

In [74]:
```
# interpolating
NO2 = df_2['NO2']
df_2['NO2'] = df_2['NO2'].interpolate(method='spline',order=3)
df_2['NO2'].fillna(method='ffill', inplace=True) # Fill missing values forward
df_2['NO2'].fillna(method='bfill', inplace=True) # Fill missing values backward
```

In [75]:
```
from pmdarima import auto_arima
stepwise_fit = auto_arima(df_2['NO2'], trace=True,
suppress_warnings=True)
```

```
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=inf, Time=5.35 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=46272.527, Time=0.09 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=43763.987, Time=0.25 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=44143.150, Time=0.37 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=46270.528, Time=0.08 sec
 ARIMA(2,1,0)(0,0,0)[0] intercept   : AIC=43755.164, Time=0.31 sec
 ARIMA(3,1,0)(0,0,0)[0] intercept   : AIC=43755.293, Time=0.45 sec
 ARIMA(2,1,1)(0,0,0)[0] intercept   : AIC=inf, Time=2.38 sec
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=43755.631, Time=0.24 sec
 ARIMA(3,1,1)(0,0,0)[0] intercept   : AIC=43759.163, Time=0.55 sec
 ARIMA(2,1,0)(0,0,0)[0]             : AIC=43753.164, Time=0.16 sec
 ARIMA(1,1,0)(0,0,0)[0]             : AIC=43761.987, Time=0.11 sec
 ARIMA(3,1,0)(0,0,0)[0]             : AIC=43753.293, Time=0.26 sec
 ARIMA(2,1,1)(0,0,0)[0]             : AIC=inf, Time=1.02 sec
 ARIMA(1,1,1)(0,0,0)[0]             : AIC=43753.630, Time=0.22 sec
 ARIMA(3,1,1)(0,0,0)[0]             : AIC=43757.163, Time=0.33 sec

Best model:  ARIMA(2,1,0)(0,0,0)[0]
Total fit time: 12.172 seconds
```

In [76]:
```
# ADF test to check for stationarity
from statsmodels.tsa.stattools import adfuller
adf_test = adfuller(df_2['NO2'])
print(f'p-value: {adf_test[1]}')
```

```
p-value: 1.842957027586097e-15
```

very low p value indicates that there is stationarity

In [ ]:
```
We will divide gthe data equally into test and training data
```

In [77]:
```
print(df_2.shape)
train=df_2.iloc[:-4000]
test=df_2.iloc[-4000:]
print(train.shape,test.shape)
```

```
(8640, 10)
(4640, 10) (4000, 10)
```

In [78]:
```
 from statsmodels.tsa.arima.model import ARIMA
model = ARIMA(train['NO2'], order=(2,1,0))
model_fit = model.fit()
print(model_fit.summary())
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                    NO2   No. Observations:                 4640
Model:                 ARIMA(2, 1, 0)   Log Likelihood              -11911.352
Date:                Tue, 27 Jun 2023   AIC                          23828.705
Time:                        23:10:49   BIC                          23848.032
Sample:                    02-01-2023   HQIC                         23835.504
                         - 03-21-2023
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.5025      0.004    114.376      0.000       0.494       0.511
ar.L2         -0.0363      0.008     -4.540      0.000      -0.052      -0.021
sigma2         9.9482      0.033    305.171      0.000       9.884      10.012
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):           1240812.61
Prob(Q):                              0.97   Prob(JB):                         0.00
Heteroskedasticity (H):               1.61   Skew:                            -4.39
Prob(H) (two-sided):                  0.00   Kurtosis:                        82.64
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

```
C:\Users\Vedant\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmodels\tsa\base\tsa_model.py:473
: ValueWarning: No frequency information was provided, so inferred frequency 15T will be used.
  self._init_dates(dates, freq)
C:\Users\Vedant\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmodels\tsa\base\tsa_model.py:473
: ValueWarning: No frequency information was provided, so inferred frequency 15T will be used.
  self._init_dates(dates, freq)
C:\Users\Vedant\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmodels\tsa\base\tsa_model.py:473
: ValueWarning: No frequency information was provided, so inferred frequency 15T will be used.
  self._init_dates(dates, freq)
```

In [79]:
```
start = len(train)
end = len(train) + len(test) - 1

# Estimate the model parameters
model_fit = model.fit()
```

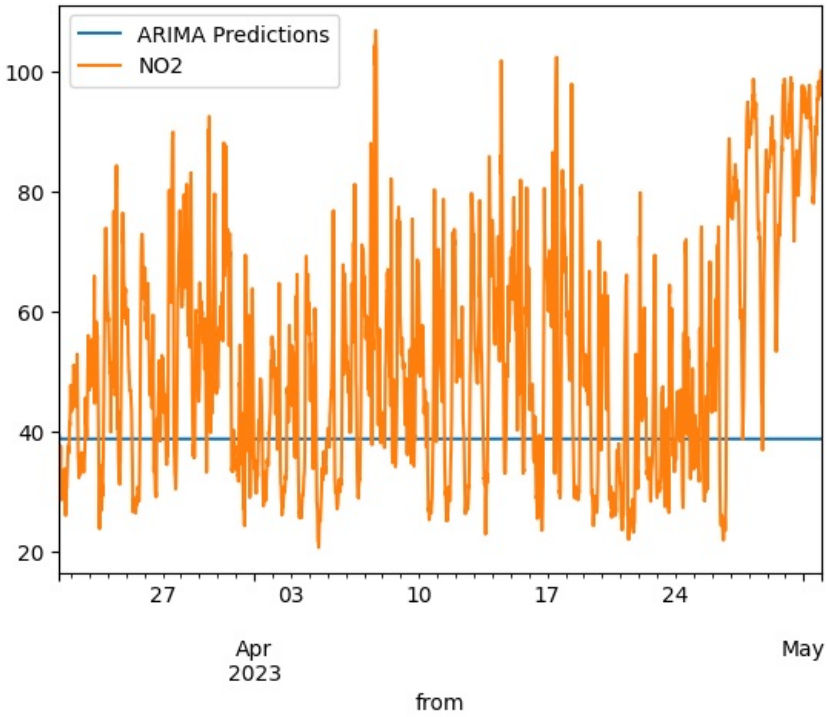
```
# Generate predictions
pred = model_fit.predict(start=start, end=end, typ='levels').rename('ARIMA Predictions')
print(pred)
```

```
2023-03-21 08:00:00    39.678314
2023-03-21 08:15:00    39.205572
2023-03-21 08:30:00    39.008804
2023-03-21 08:45:00    38.927117
2023-03-21 09:00:00    38.893224
                         ...
2023-05-01 22:45:00    38.869194
2023-05-01 23:00:00    38.869194
2023-05-01 23:15:00    38.869194
2023-05-01 23:30:00    38.869194
2023-05-01 23:45:00    38.869194
Freq: 15T, Name: ARIMA Predictions, Length: 4000, dtype: float64
```

```
C:\Users\Vedant\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmodels\tsa\statespace\representa
tion.py:374: FutureWarning: Unknown keyword arguments: dict_keys(['typ']).Passing unknown keyword arguments wil
l raise a TypeError beginning in version 0.15.
  warnings.warn(msg, FutureWarning)
```

In [80]:
```
start = len(train)
end = len(train) + len(test) - 1

# Fit the model
model_fit = model.fit()

# Generate predictions
pred = model_fit.predict(start=start, end=end, typ='levels', dynamic=False).rename('ARIMA Predictions')
pred.plot(legend=True)
test['NO2'].plot(legend=True)
plt.show()
```

```
C:\Users\Vedant\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmodels\tsa\statespace\representa
tion.py:374: FutureWarning: Unknown keyword arguments: dict_keys(['typ']).Passing unknown keyword arguments wil
l raise a TypeError beginning in version 0.15.
  warnings.warn(msg, FutureWarning)
```



In [81]:
```
from sklearn.metrics import mean_squared_error
from math import sqrt
test['NO2'].mean()
rmse=sqrt(mean_squared_error(pred,test['NO2']))
print(rmse)
```

```
23.249513667188275
```

In [ ]:
```
we here get the mean squared errror .
```

In [82]:
```
test['NO2'].mean()
```

Out[82]:
```
52.10891374566813
```

In [ ]:
```
here we have the mean of the data .
```

Resampling the data and then again we will build an ARIMA model.

In [83]:
```
df_resample=df.copy()
```

In [84]:
```
df_resample = df_resample.resample('D').mean()
# interpolating
NO2 = df_resample['NO2']
df_resample = df_resample.interpolate(method='spline', order=3)
df_resample.fillna(method='ffill', inplace=True)  # Fill missing values forward
df_resample.fillna(method='bfill', inplace=True)  # Fill missing values backward
```

In [85]:
```
df_resample.head()
```

Out[85]:

from	PM10	PM2.5	NO	NO2	NOX	CO	SO2	NH3	Ozone	Benzene
2023-02-01	114.739583	35.145833	10.622222	79.180645	48.821505	441.182796	10.244681	22.080851	30.076744	0.232292
2023-02-02	177.458333	52.020833	10.622222	79.286957	53.986957	1303.152174	10.244681	22.266667	25.331183	0.120000
2023-02-03	171.270833	52.916667	10.622222	82.408602	56.936559	1211.075269	10.244681	23.105319	27.535106	0.165625
2023-02-04	222.552941	74.651685	10.622222	76.781319	53.721978	1146.222222	10.244681	25.094565	26.340659	0.184444
2023-02-05	271.354430	86.987952	10.622222	74.712048	66.445783	890.357143	10.244681	25.571429	21.824706	0.214286

In [86]:
```
 # just checking code ADF test to check for stationarity
from statsmodels.tsa.stattools import adfuller
adf_test = adfuller(df_resample['NO2'])
print(f'p-value: {adf_test[1]}')
```

```
p-value: 0.01984267879976745
```

very low p value indicates stationarity

In [87]:
```
from pmdarima import auto_arima
stepwise_fit = auto_arima(df_resample['NO2'], trace=True,
suppress_warnings=True)
```

```
Performing stepwise search to minimize aic
 ARIMA(2,0,2)(0,0,0)[0] intercept   : AIC=652.267, Time=0.14 sec
 ARIMA(0,0,0)(0,0,0)[0] intercept   : AIC=741.072, Time=0.01 sec
 ARIMA(1,0,0)(0,0,0)[0] intercept   : AIC=647.335, Time=0.03 sec
 ARIMA(0,0,1)(0,0,0)[0] intercept   : AIC=683.724, Time=0.03 sec
 ARIMA(0,0,0)(0,0,0)[0]             : AIC=986.706, Time=0.01 sec
 ARIMA(2,0,0)(0,0,0)[0] intercept   : AIC=648.337, Time=0.06 sec
 ARIMA(1,0,1)(0,0,0)[0] intercept   : AIC=648.525, Time=0.06 sec
 ARIMA(2,0,1)(0,0,0)[0] intercept   : AIC=650.405, Time=0.07 sec
 ARIMA(1,0,0)(0,0,0)[0]             : AIC=inf, Time=0.01 sec

Best model:  ARIMA(1,0,0)(0,0,0)[0] intercept
Total fit time: 0.415 seconds
```

In [88]:
```
print(df.shape)
train=df.iloc[:-4000]
test=df.iloc[-4000:]
print(train.shape,test.shape)
```

```
(8640, 10)
(4640, 10) (4000, 10)
```

In [89]:
```
from statsmodels.tsa.arima.model import ARIMA
model=ARIMA(train['NO2'],order=(1,0,0))
model=model.fit()
model.summary()
```

```
C:\Users\Vedant\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmodels\tsa\base\tsa_model.py:473
: ValueWarning: No frequency information was provided, so inferred frequency 15T will be used.
  self._init_dates(dates, freq)
C:\Users\Vedant\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmodels\tsa\base\tsa_model.py:473
: ValueWarning: No frequency information was provided, so inferred frequency 15T will be used.
  self._init_dates(dates, freq)
C:\Users\Vedant\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmodels\tsa\base\tsa_model.py:473
: ValueWarning: No frequency information was provided, so inferred frequency 15T will be used.
  self._init_dates(dates, freq)
```

Out[89]:

SARIMAX Results

Dep. Variable:	NO2	**No. Observations:**	4640
Model:	ARIMA(1, 0, 0)	**Log Likelihood**	-11891.685
Date:	Tue, 27 Jun 2023	**AIC**	23789.371
Time:	23:11:29	**BIC**	23808.698
Sample:	02-01-2023	**HQIC**	23796.171
	- 03-21-2023		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
const	58.8977	3.490	16.876	0.000	52.057	65.738
ar.L1	0.9843	0.003	299.143	0.000	0.978	0.991
sigma2	13.5531	0.087	155.468	0.000	13.382	13.724

Ljung-Box (L1) (Q):	865.71	**Jarque-Bera (JB):**	514981.52
Prob(Q):	0.00	**Prob(JB):**	0.00
Heteroskedasticity (H):	1.70	**Skew:**	-3.38
Prob(H) (two-sided):	0.00	**Kurtosis:**	54.17

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

In [ ]:

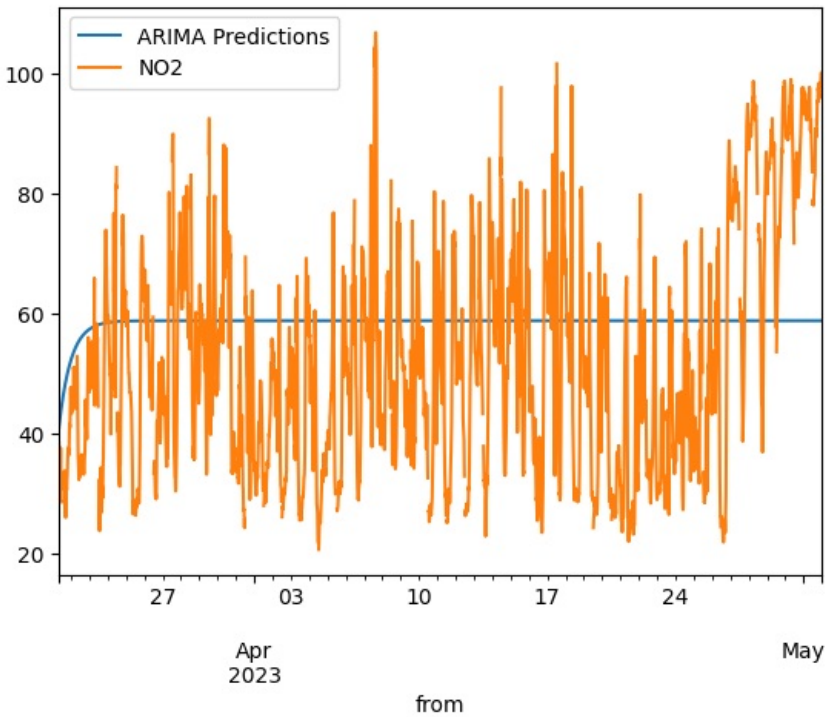
Here we get the ARIMA prediction for the analysis

In [98]:

```
start=len(train)
end=len(train)+len(test)-1
pred=model.predict(start=start,end=end,typ='levels').rename('ARIMA Predictions')
pred.plot(legend=True)
test['NO2'].plot(legend=True)
```

```
C:\Users\Vedant\AppData\Local\Programs\Python\Python310\lib\site-packages\statsmodels\tsa\statespace\representa
tion.py:374: FutureWarning: Unknown keyword arguments: dict_keys(['typ']).Passing unknown keyword arguments wil
l raise a TypeError beginning in version 0.15.
  warnings.warn(msg, FutureWarning)
```

Out[98]: <Axes: xlabel='from'>



In [102…

```
test['NO2'].mean()
```

Out[102]: 52.215448204598296

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

3) Statistical inference

```
import numpy as np
import pandas as pd
df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/Data Sets/Assignment/Data Sets/Assignment Data Sets/Beijing_PM25.csv')
pd.set_option('display.max_columns', 50)
pd.set_option('display.max_rows', 50)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 43824 entries, 0 to 43823
Data columns (total 13):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   No      43824 non-null  int64
 1   year    43824 non-null  int64
 2   month   43824 non-null  int64
 3   day     43824 non-null  int64
 4   hour    43824 non-null  int64
 5   pm2.5   41757 non-null  float64
 6   DEWP    43824 non-null  int64
 7   TEMP    43824 non-null  int64
 8   PRES    43824 non-null  int64
 9   cbwd    43824 non-null  object
 10  Iws     43824 non-null  int64
 11  Is      43824 non-null  int64
 12  Ir      43824 non-null  int64
dtype: float64(1), int64(11), object(1)
memory usage: 4.3+ MB
```

```
df.describe(include='object')
```

	cbwd
count	43824
unique	4
top	NW
freq	15262

```
df.describe(include='datetime')
```

```
ValueError: Cannot select dtypes: exclude=['datetime'] matches none of the select dtypes.
```

```
df.describe(exclude=['object'])
```

	No	year	month	day	hour	pm2.5	DEWP	TEMP	PRES	Iws	Is	Ir
count	43824.000000	43824.000000	43824.000000	43824.000000	43824.000000	41757.000000	43824.000000	43824.000000	43824.000000	43824.000000	43824.000000	43824.000000
mean	21912.500000	2012.500000	6.487210	15.727365	11.500000	98.613215	1.817246	12.448521	1016.447654	23.889140	0.052734	0.194966
std	12651.643719	0.866025	3.448572	8.799425	6.922281	92.050387	14.433440	12.914486	10.268698	50.010635	0.760375	1.415867
min	1.000000	2010.000000	1.000000	1.000000	0.000000	0.000000	-40.000000	-19.000000	991.000000	0.450000	0.000000	0.000000
25%	10956.750000	2011.750000	3.750000	8.000000	5.750000	29.000000	-10.000000	2.000000	1008.000000	1.790000	0.000000	0.000000
50%	21912.500000	2012.500000	7.000000	16.000000	11.500000	72.000000	2.000000	14.000000	1016.000000	5.370000	0.000000	0.000000
75%	32868.250000	2013.250000	10.000000	23.000000	17.250000	137.000000	15.000000	23.000000	1025.000000	21.910000	0.000000	0.000000
max	43824.000000	2014.000000	12.000000	31.000000	23.000000	994.000000	28.000000	42.000000	1046.000000	585.600000	27.000000	36.000000

```
df.describe(exclude=['object'])
```

	No	year	month	day	hour	pm2.5	DEWP	TEMP	PRES	Iws	Is	Ir
count	43824.000000	43824.000000	43824.000000	43824.000000	43824.000000	41757.000000	43824.000000	43824.000000	43824.000000	43824.000000	43824.000000	43824.000000
mean	21912.500000	2012.500000	6.487210	15.727365	11.500000	98.613215	1.817246	12.448521	1016.447654	23.889140	0.052734	0.194966
std	12651.643719	0.866025	3.448572	8.799425	6.922281	92.050387	14.433440	12.914486	10.268698	50.010635	0.760375	1.415867
min	1.000000	2010.000000	1.000000	1.000000	0.000000	0.000000	-40.000000	-19.000000	991.000000	0.450000	0.000000	0.000000
25%	10956.750000	2011.750000	3.750000	8.000000	5.750000	29.000000	-10.000000	2.000000	1008.000000	1.790000	0.000000	0.000000
50%	21912.500000	2012.500000	7.000000	16.000000	11.500000	72.000000	2.000000	14.000000	1016.000000	5.370000	0.000000	0.000000
75%	32868.250000	2013.250000	10.000000	23.000000	17.250000	137.000000	15.000000	23.000000	1025.000000	21.910000	0.000000	0.000000
max	43824.000000	2014.000000	12.000000	31.000000	23.000000	994.000000	28.000000	42.000000	1046.000000	585.600000	27.000000	36.000000

```
df.sample(5)
```

	No	year	month	day	hour	pm2.5	DEWP	TEMP	PRES	cbwd	Iws	Is	Ir
16	17	2010	1	2	16	13.0	-11	-5	1034	NW	79.23	0	0
17	18	2010	1	2	17	16.0	-12	-5	1034	NW	85.24	0	0
18	19	2010	1	2	18	12.0	-11	-5	1035	NW	91.32	0	0
19	20	2010	1	2	19	7.0	-10	-5	1036	NW	97.41	0	0
20	21	2010	1	2	20	3.0	-10	-5	1037	NW	103.49	0	0

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

```
No       0
year     0
month    0
day      0
hour     0
pm2.5    0
DEWP     0
TEMP     0
PRES     0
cbwd     0
Iws      0
Is       0
Ir       0
dtype: int64
```

```
df.isnull()
```

	No	year	month	day	hour	pm2.5	DEWP	TEMP	PRES	cbwd	Iws	Is	Ir
0	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...
43819	False	False	False	False	False	False	False	False	False	False	False	False	False
43820	False	False	False	False	False	False	False	False	False	False	False	False	False
43821	False	False	False	False	False	False	False	False	False	False	False	False	False
43822	False	False	False	False	False	False	False	False	False	False	False	False	False
43823	False	False	False	False	False	False	False	False	False	False	False	False	False

43824 rows × 13 columns

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

```
No       0
year     0
month    0
day      0
hour     0
pm2.5    0
DEWP     0
TEMP     0
PRES     0
cbwd     0
Iws      0
Is       0
Ir       0
dtype: int64
```

```
df.isnull().sum()/df.shape[0]*100
```

```
No       0.000000
year     0.000000
month    0.000000
day      0.000000
hour     0.000000
pm2.5    4.749087
DEWP     0.000000
TEMP     0.000000
PRES     0.000000
cbwd     0.000000
Iws      0.000000
Is       0.000000
Ir       0.000000
dtype: float64
```

Calculating mean,median and mode for all pollutants

```
import numpy as np
df['pm2.5'].mean()
df['pm2.5'].fillna(df['pm2.5'].mean(), inplace=True)
df['pm2.5'].isnull().sum()
```

```
0
```

```
df['pm2.5'].mean()
```

```
98.61321500000000
```

```
```

3) Exploratory Analysis of the Data and studying the correlation of NO2 with the rest of the gases .

Overview

Dataset statistics

Number of variables	13
Number of observations	8643
Missing cells	13031
Missing cells (%)	11.6%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	877.9 KiB
Average record size in memory	104.0 B

Variable types

Numeric	11
Text	1
DateTime	1

Alerts

# is highly overall correlated with Singrauli, Surya Kiran Bhawan Dudhichua NH3 (碌g/m3)	High correlation
Singrauli, Surya Kiran Bhawan Dudhichua PM10 (碌g/m3) is highly overall correlated with Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (碌g/m3) and 1 other fields (Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua Benzene (碌g/m3))	High correlation
Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (碌g/m3) is highly overall correlated with Singrauli, Surya Kiran Bhawan Dudhichua PM10 (碌g/m3) and 4 other fields (Singrauli, Surya Kiran Bhawan Dudhichua PM10 (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua NO (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb), Singrauli, Surya Kiran Bhawan Dudhichua Ozone (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua Benzene (碌g/m3))	High correlation
Singrauli, Surya Kiran Bhawan Dudhichua NO (碌g/m3) is highly overall correlated with Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (碌g/m3) and 3 other fields (Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua NO2 (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb), Singrauli, Surya Kiran Bhawan Dudhichua Ozone (碌g/m3))	High correlation
Singrauli, Surya Kiran Bhawan Dudhichua NO2 (碌g/m3) is highly overall correlated with Singrauli, Surya Kiran Bhawan Dudhichua NO (碌g/m3) and 1 other fields (Singrauli, Surya Kiran Bhawan Dudhichua NO (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb))	High correlation
Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb) is highly overall correlated with Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (碌g/m3) and 3 other fields (Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua NO (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua NO2 (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua Ozone (碌g/m3))	High correlation
Singrauli, Surya Kiran Bhawan Dudhichua NH3 (碌g/m3) is highly overall correlated with #	High correlation
Singrauli, Surya Kiran Bhawan Dudhichua Ozone (碌g/m3) is highly overall correlated with Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (碌g/m3) and 2 other fields (Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua NO (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb))	High correlation
Singrauli, Surya Kiran Bhawan Dudhichua Benzene (碌g/m3) is highly overall correlated with Singrauli, Surya Kiran Bhawan Dudhichua PM10 (碌g/m3) and 1 other fields (Singrauli, Surya Kiran Bhawan Dudhichua PM10 (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (碌g/m3))	High correlation
Singrauli, Surya Kiran Bhawan Dudhichua PM10 (碌g/m3) has 1681 (19.4%) missing values	Missing
Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (碌g/m3) has 226 (2.6%) missing values	Missing
Singrauli, Surya Kiran Bhawan Dudhichua NO (碌g/m3) has 1369 (15.8%) missing values	Missing
Singrauli, Surya Kiran Bhawan Dudhichua NO2 (碌g/m3) has 416 (4.8%) missing values	Missing
Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb) has 415 (4.8%) missing values	Missing
Singrauli, Surya Kiran Bhawan Dudhichua CO (mg/m3) has 496 (5.7%) missing values	Missing
Singrauli, Surya Kiran Bhawan Dudhichua SO2 (碌g/m3) has 1451 (16.8%) missing values	Missing
Singrauli, Surya Kiran Bhawan Dudhichua NH3 (碌g/m3) has 326 (3.8%) missing values	Missing
Singrauli, Surya Kiran Bhawan Dudhichua Ozone (碌g/m3) has 453 (5.2%) missing values	Missing
Singrauli, Surya Kiran Bhawan Dudhichua Benzene (碌g/m3) has 6195 (71.7%) missing values	Missing
# is uniformly distributed	Uniform

# has unique values	Unique
From has unique values	Unique

Reproduction

Analysis started	2023-06-27 12:23:56.443253
Analysis finished	2023-06-27 12:24:26.751240
Duration	30.31 seconds
Software version	ydata-profiling vv4.3.1 (https://github.com/ydataai/ydata-profiling)
Download configuration	config.json (data:text/plain;charset=utf-8,%7B%22title%22%3A%20%22Pandas%20Profiling%20Report%22%2C%20%22dataset%22%3A%20%7B%22description%22%3A%20%22%22%2

Variables

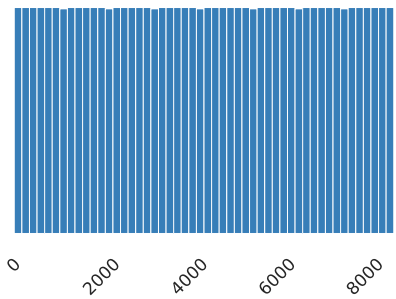
Select Columns

#

Real number (ℝ)

HIGH CORRELATION (This variable has a high overall correlation with 1 fields: Singrauli, Surya Kiran Bhawan Dudhichua NH3 (碌g/m3)) UNIFOR
UNIQUE

Distinct	8643	**Minimum**	1
Distinct (%)	100.0%	**Maximum**	8643
Missing	0	**Zeros**	0
Missing (%)	0.0%	**Zeros (%)**	0.0%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	4322	**Memory size**	67.6 KiB

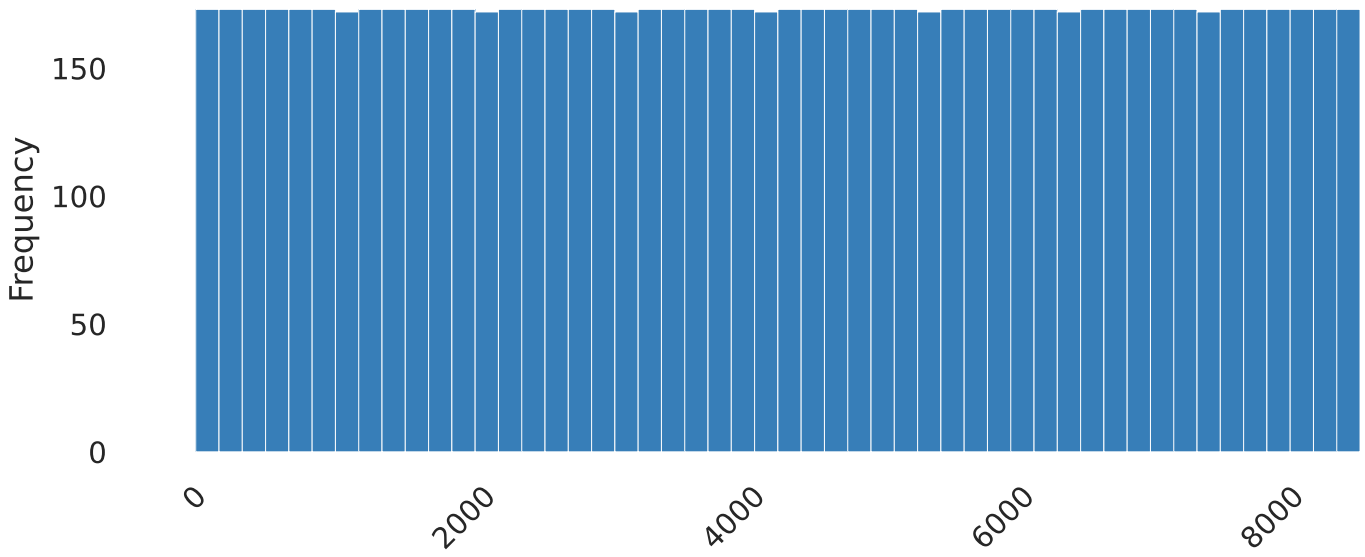


Quantile statistics

Minimum	1
5-th percentile	433.1
Q1	2161.5
median	4322
Q3	6482.5
95-th percentile	8210.9
Maximum	8643
Range	8642
Interquartile range (IQR)	4321

Descriptive statistics

Standard deviation	2495.1635
Coefficient of variation (CV)	0.57731687
Kurtosis	-1.2
Mean	4322
Median Absolute Deviation (MAD)	2161
Skewness	0
Sum	37355046
Variance	6225841
Monotonicity	Strictly increasing



Histogram with fixed size bins (bins=50)

Value	Count	Frequency
1	1	< 0.1
5758	1	< 0.1
5772	1	< 0.1
5771	1	< 0.1
5770	1	< 0.1
5769	1	< 0.1
5768	1	< 0.1
5767	1	< 0.1
5766	1	< 0.1
5765	1	< 0.1
Other values (8633)	8633	99.9%

Value	Count	Frequency (
1	1	< 0.1%
2	1	< 0.1%
3	1	< 0.1%
4	1	< 0.1%
5	1	< 0.1%
6	1	< 0.1%
7	1	< 0.1%
8	1	< 0.1%
9	1	< 0.1%
10	1	< 0.1%

Value	Count	Frequency (

8643	1	< 0.1%
8642	1	< 0.1%
8641	1	< 0.1%
8640	1	< 0.1%
8639	1	< 0.1%
8638	1	< 0.1%
8637	1	< 0.1%
8636	1	< 0.1%
8635	1	< 0.1%
8634	1	< 0.1%

From

Text

UNIQUE

Distinct	8643
Distinct (%)	100.0%
Missing	0
Missing (%)	0.0%
Memory size	67.6 KiB

Length

Max length	19
Median length	19
Mean length	18.57804
Min length	3

Characters and Unicode

Total characters	160570
Distinct characters	22
Distinct categories	6 (https://en.wikipedia.org/wiki/Unicode_character_property#General_Category) ?
Distinct scripts	2 (https://en.wikipedia.org/wiki/Script_(Unicode)#List_of_scripts_in_Unicode) ?
Distinct blocks	1 (https://en.wikipedia.org/wiki/Unicode_block) ?

The Unicode Standard assigns character properties to each code point, which can be used to analyse textual variables.

Unique

Unique	8643	?
Unique (%)	100.0%	

Sample

1st row	2023-02-01 0:00:00
2nd row	2023-02-01 0:15:00
3rd row	2023-02-01 0:30:00
4th row	2023-02-01 0:45:00
5th row	2023-02-01 1:00:00

Value	Count	Frequency (%)
2023-02-01	96	0.6%
2023-04-24	96	0.6%
2023-04-25	96	0.6%
2023-04-26	96	0.6%
2023-04-21	96	0.6%
2023-04-22	96	0.6%
2023-04-20	96	0.6%
2023-04-30	96	0.6%
2023-04-29	96	0.6%
2023-05-01	96	0.6%
Other values (179)	16323	94.4%

Most occurring characters

Value	Count	Frequency (%)

0	45576	28.4%
2	26136	16.3%
-	17280	10.8%
:	17280	10.8%
3	16008	10.0%
1	10776	6.7%
	8640	5.4%
4	6624	4.1%
5	6000	3.7%
6	1584	1.0%
Other values (12)	4666	2.9%

Most occurring categories

Value	Count	Frequency (%)
Decimal Number	117360	73.1%
Other Punctuation	17281	10.8%
Dash Punctuation	17280	10.8%
Space Separator	8640	5.4%
Lowercase Letter	6	< 0.1%
Uppercase Letter	3	< 0.1%

Most frequent character per category

Decimal Number

Value	Count	Frequency (%)
0	45576	38.8%
2	26136	22.3%
3	16008	13.6%
1	10776	9.2%
4	6624	5.6%
5	6000	5.1%
6	1584	1.3%
8	1584	1.3%
7	1584	1.3%
9	1488	1.3%

Lowercase Letter

Value	Count	Frequency (%)
a	1	16.7%

x	1	16.7%
i	1	16.7%
n	1	16.7%
v	1	16.7%
g	1	16.7%

Other Punctuation

Value	Count	Frequency (%)
:	17280	> 99.9%
.	1	< 0.1%

Uppercase Letter

Value	Count	Frequency (%)
M	2	66.7%
A	1	33.3%

Dash Punctuation

Value	Count	Frequency (%)
-	17280	100.0%

Space Separator

Value	Count	Frequency (%)
	8640	100.0%

Most occurring scripts

Value	Count	Frequency (%)
Common	160561	> 99.9%
Latin	9	< 0.1%

Most frequent character per script

Common

Value	Count	Frequency (%)
0	45576	28.4%
2	26136	16.3%
-	17280	10.8%
:	17280	10.8%
3	16008	10.0%
1	10776	6.7%
	8640	5.4%
4	6624	4.1%

5	6000	3.7%
6	1584	1.0%
Other values (4)	4657	2.9%

Latin

Value	Count	Frequency (%)
M	2	22.2%
a	1	11.1%
x	1	11.1%
i	1	11.1%
n	1	11.1%
A	1	11.1%
v	1	11.1%
g	1	11.1%

Most occurring blocks

Value	Count	Frequency (%)
ASCII	160570	100.0%

Most frequent character per block

ASCII

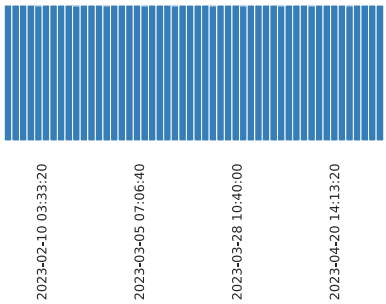
Value	Count	Frequency (%)
0	45576	28.4%
2	26136	16.3%
-	17280	10.8%
:	17280	10.8%
3	16008	10.0%
1	10776	6.7%
	8640	5.4%
4	6624	4.1%
5	6000	3.7%
6	1584	1.0%
Other values (12)	4666	2.9%

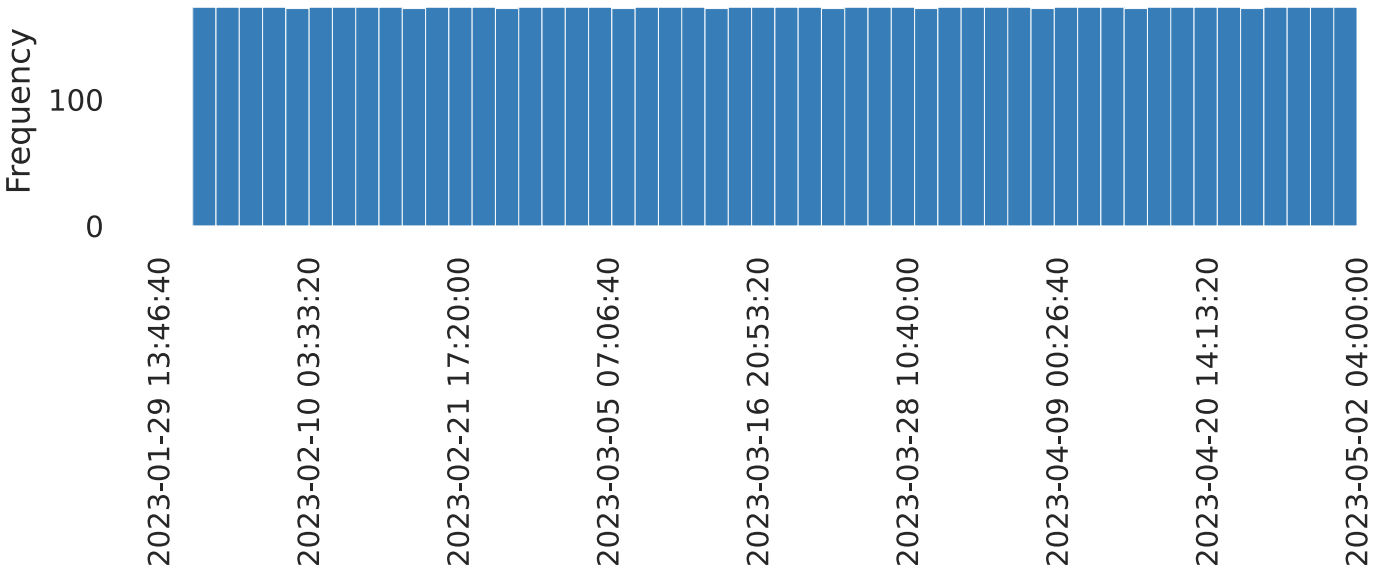
To (Interval: 15M)

Date

Distinct	8640
Distinct (%)	100.0%
Missing	3
Missing (%)	< 0.1%
Memory size	67.6 KiB

Minimum	2023-02-01 00:15:00
Maximum	2023-05-02 00:00:00





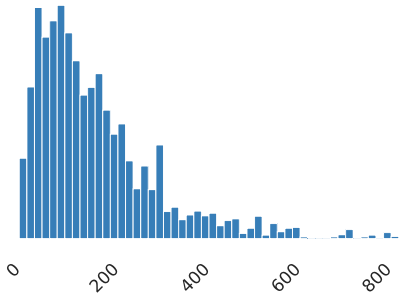
Histogram with fixed size bins (bins=50)

Singrauli, Surya Kiran Bhawan Dudhichua PM10 (碌g/m3)

Real number (ℝ)

`HIGH CORRELATION (This variable has a high overall correlation with 2 fields: Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua Benzene (碌g/m3))` `MISSING`

Distinct	376	**Minimum**	12
Distinct (%)	5.4%	**Maximum**	847
Missing	1681	**Zeros**	0
Missing (%)	19.4%	**Zeros (%)**	0.0%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	181.47995	**Memory size**	67.6 KiB

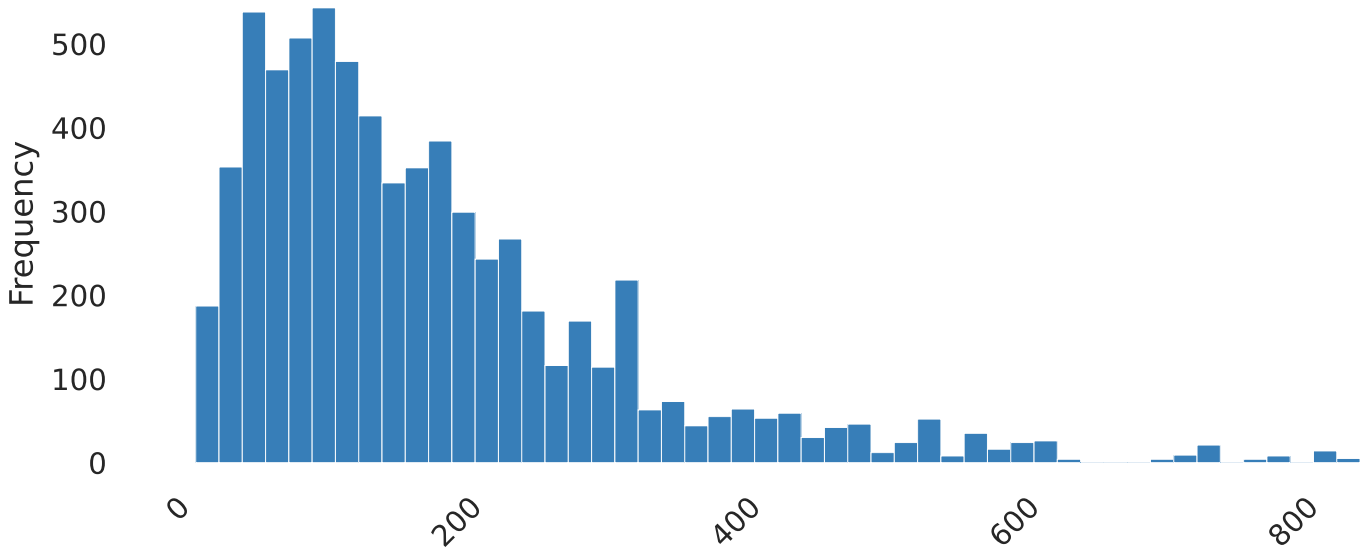


Quantile statistics

Minimum	12
5-th percentile	37
Q1	84
median	145
Q3	238
95-th percentile	469
Maximum	847
Range	835
Interquartile range (IQR)	154

Descriptive statistics

Standard deviation	136.23574
Coefficient of variation (CV)	0.75069308
Kurtosis	3.2784936
Mean	181.47995
Median Absolute Deviation (MAD)	72
Skewness	1.6269462
Sum	1263463.4
Variance	18560.177
Monotonicity	Not monotonic



Histogram with fixed size bins (bins=50)

Value	Count	Frequency
53	127	1.5
327	78	0.9
99	77	0.9
84	65	0.8

122	64	0.7
133	63	0.7
117	61	0.7
113	60	0.7
178	59	0.7
182	55	0.6
Other values (366)	6253	72.3%
(Missing)	1681	19.4

Value	Count	Frequency (
12	5	0.1%
13	8	0.1%
15	4	< 0.1%
18	30	0.3%
19	20	0.2%
20	8	0.1%
21	8	0.1%
22	12	0.1%
23	16	0.2%
24	20	0.2%

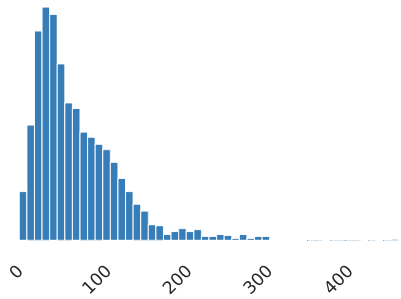
Value	Count	Frequency (
847	5	0.1%
823	14	0.2%
796	4	< 0.1%
793	4	< 0.1%
770	4	< 0.1%
743	4	< 0.1%
742	4	< 0.1%
741	9	0.1%
735	4	< 0.1%
726	4	< 0.1%

Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (碌g/m3)

Real number (ℝ)

`HIGH CORRELATION` (This variable has a high overall correlation with 5 fields: Singrauli, Surya Kiran Bhawan Dudhichua PM10 (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua NO (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb), Singrauli, Surya Kiran Bhawan Dudhichua Ozone (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua Benzene (碌g/m3)) `MISSING`

Distinct	245	**Minimum**	3
Distinct (%)	2.9%	**Maximum**	474
Missing	226	**Zeros**	0
Missing (%)	2.6%	**Zeros (%)**	0.0%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	75.729083	**Memory size**	67.6 KiB

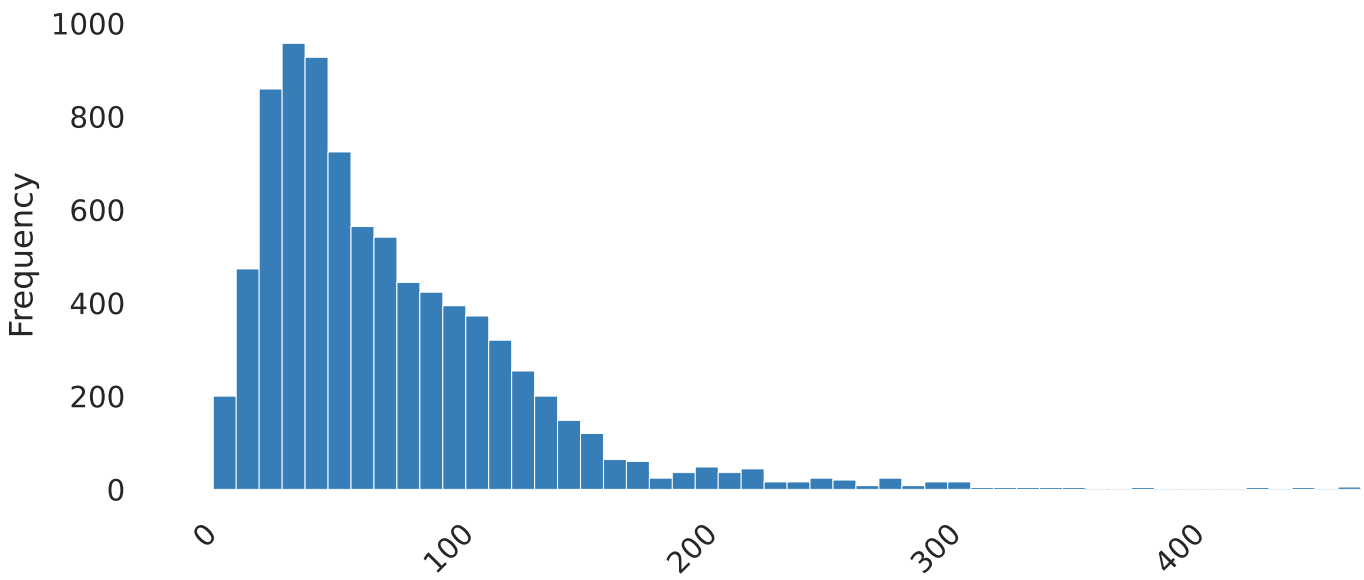


Quantile statistics

Minimum	3
5-th percentile	17
Q1	36
median	61
Q3	101
95-th percentile	175
Maximum	474
Range	471
Interquartile range (IQR)	65

Descriptive statistics

Standard deviation	55.41145
Coefficient of variation (CV)	0.73170633
Kurtosis	6.2350808
Mean	75.729083
Median Absolute Deviation (MAD)	29
Skewness	1.9454314
Sum	637411.69
Variance	3070.4287
Monotonicity	Not monotonic



Histogram with fixed size bins (bins=50)

Value	Count	Frequency
27	143	1.7
32	136	1.6
37	132	1.5
42	132	1.5

35	123	1.4
48	123	1.4
33	118	1.4
36	112	1.3
39	108	1.2
47	108	1.2
Other values (235)	7182	83.1%
(Missing)	226	2.6

Value	Count	Frequency (
3	5	0.1%
4	16	0.2%
5	16	0.2%
6	25	0.3%
7	12	0.1%
8	16	0.2%
9	20	0.2%
10	39	0.5%
11	36	0.4%
12	15	0.2%

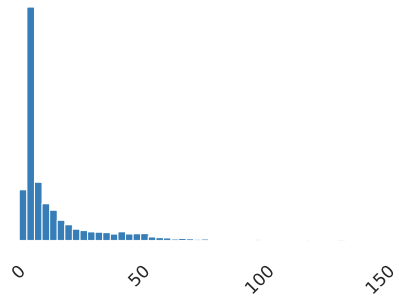
Value	Count	Frequency (
474	5	0.1%
448	4	< 0.1%
427	4	< 0.1%
389	4	< 0.1%
353	4	< 0.1%
343	4	< 0.1%
339	4	< 0.1%
327	4	< 0.1%
323	4	< 0.1%
312	4	< 0.1%

Singrauli, Surya Kiran Bhawan Dudhichua NO (碌g/m3)

Real number (ℝ)

`HIGH CORRELATION (This variable has a high overall correlation with 4 fields: Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua NO2 (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb), Singrauli, Surya Kiran Bhawan Dudhichua Ozone (碌g/m3))` `MISSING`

Distinct	775	**Minimum**	0.1
Distinct (%)	10.7%	**Maximum**	157.5
Missing	1369	**Zeros**	0
Missing (%)	15.8%	**Zeros (%)**	0.0%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	14.667274	**Memory size**	67.6 KiB

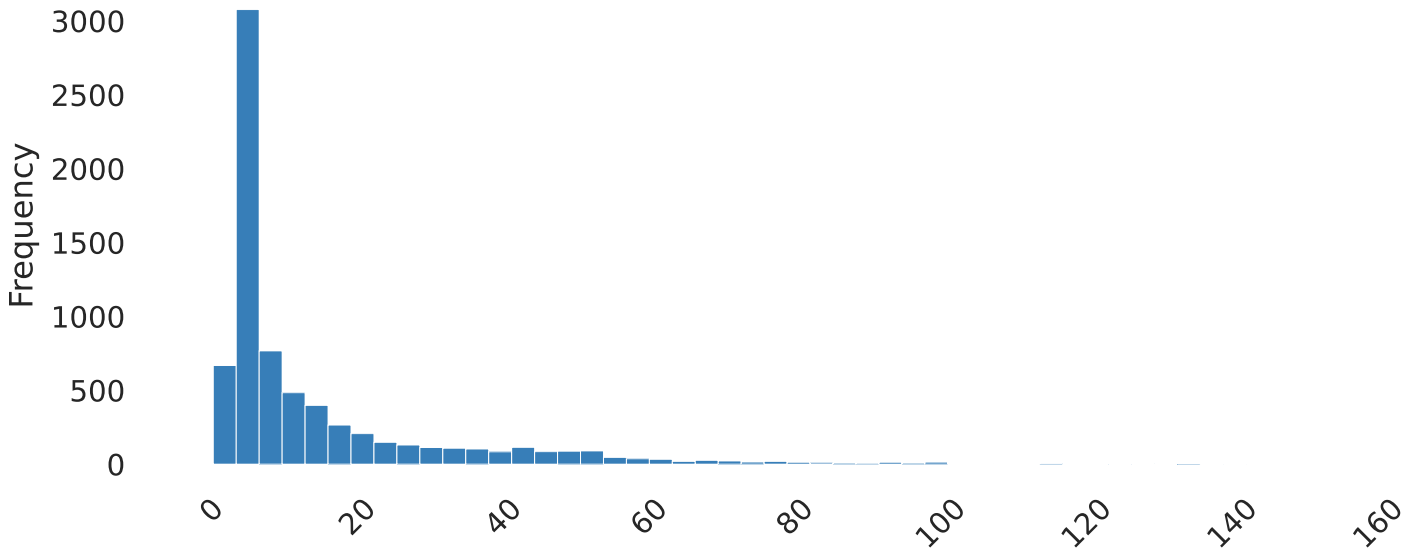


Quantile statistics

Minimum	0.1
5-th percentile	2.9
Q1	3.9
median	6.1
Q3	16.5
95-th percentile	52.8
Maximum	157.5
Range	157.4
Interquartile range (IQR)	12.6

Descriptive statistics

Standard deviation	19.291029
Coefficient of variation (CV)	1.315243
Kurtosis	10.746695
Mean	14.667274
Median Absolute Deviation (MAD)	2.8
Skewness	2.8731225
Sum	106689.75
Variance	372.14378
Monotonicity	Not monotonic



Histogram with fixed size bins (bins=50)

Value	Count	Frequency
3.8	225	2.6
3.9	194	2.2
3.5	183	2.1
3.6	183	2.1

4	179	2.1
3.7	174	2.0
4.2	171	2.0
3.4	170	2.0
4.1	160	1.9
3.3	153	1.8
Other values (765)	5482	63.4%
(Missing)	1369	15.8

Value	Count	Frequency (
0.1	2	< 0.1%
0.2	1	< 0.1%
0.4	3	< 0.1%
0.5	6	0.1%
0.6	1	< 0.1%
0.7	7	0.1%
0.8	4	< 0.1%
0.9	8	0.1%
1	12	0.1%
1.1	7	0.1%

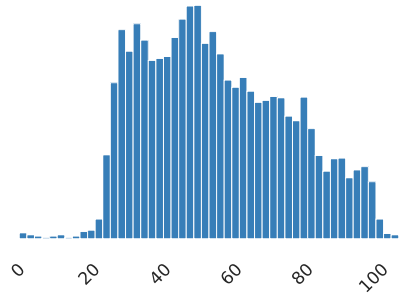
Value	Count	Frequency (
157.5	2	< 0.1%
157.1	1	< 0.1%
155	1	< 0.1%
154.7	1	< 0.1%
153.9	1	< 0.1%
151.5	1	< 0.1%
150.7	1	< 0.1%
150.6	1	< 0.1%
147.6	1	< 0.1%
145.9	1	< 0.1%

Singrauli, Surya Kiran Bhawan Dudhichua NO2 (碌g/m3)

Real number (ℝ)

HIGH CORRELATION (This variable has a high overall correlation with 2 fields: Singrauli, Surya Kiran Bhawan Dudhichua NO (碌g/m3), Singraul Surya Kiran Bhawan Dudhichua NOX (ppb)) MISSING

Distinct	861	**Minimum**	0.2
Distinct (%)	10.5%	**Maximum**	106.9
Missing	416	**Zeros**	0
Missing (%)	4.8%	**Zeros (%)**	0.0%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	55.756492	**Memory size**	67.6 KiB

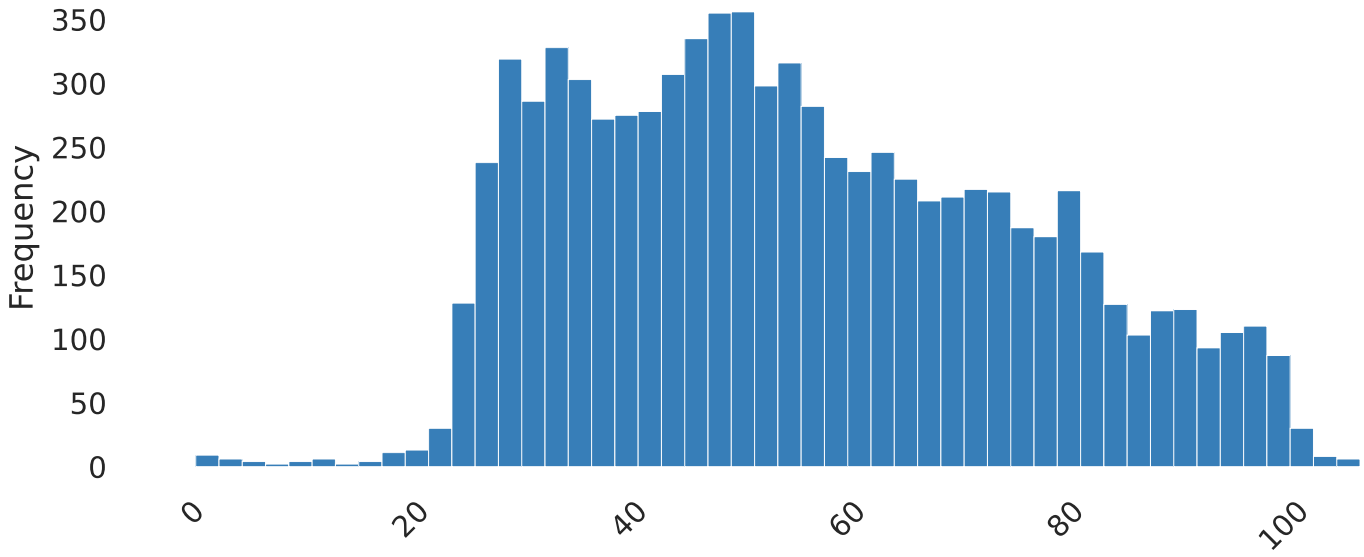


Quantile statistics

Minimum	0.2
5-th percentile	27.6
Q1	39.4
median	53.2
Q3	71.05
95-th percentile	92.6
Maximum	106.9
Range	106.7
Interquartile range (IQR)	31.65

Descriptive statistics

Standard deviation	20.244845
Coefficient of variation (CV)	0.36309395
Kurtosis	-0.7261858
Mean	55.756492
Median Absolute Deviation (MAD)	15.6
Skewness	0.32976986
Sum	458708.66
Variance	409.85374
Monotonicity	Not monotonic



Histogram with fixed size bins (bins=50)

Value	Count	Frequency
55.3	29	0.3
48.6	27	0.3
51.2	26	0.3
35.5	26	0.3

43.1	25	0.3
47.2	25	0.3
28	24	0.3
33.1	24	0.3
43.7	24	0.3
46.2	24	0.3
Other values (851)	7973	92.2%
(Missing)	416	4.8

Value	Count	Frequency (
0.2	2	< 0.1%
0.6	1	< 0.1%
0.7	1	< 0.1%
0.8	1	< 0.1%
0.9	1	< 0.1%
1.9	2	< 0.1%
2	1	< 0.1%
2.6	2	< 0.1%
2.8	1	< 0.1%
3	2	< 0.1%

Value	Count	Frequency (
106.9	2	< 0.1%
106.6	1	< 0.1%
105.8	1	< 0.1%
105.1	1	< 0.1%
104.8	1	< 0.1%
104.4	1	< 0.1%
104.3	1	< 0.1%
104.2	1	< 0.1%
104	1	< 0.1%
103.6	1	<

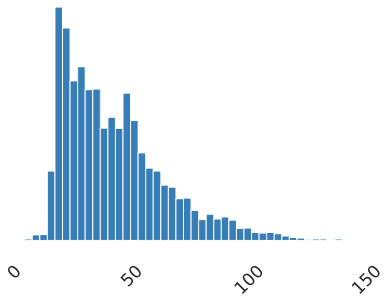
0.1%

Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb)

Real number (ℝ)

`HIGH CORRELATION (This variable has a high overall correlation with 4 fields: Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua NO (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua NO2 (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua Ozone (碌g/m3))` `MISSING`

Distinct	1009	**Minimum**	4.2
Distinct (%)	12.3%	**Maximum**	165.2
Missing	415	**Zeros**	0
Missing (%)	4.8%	**Zeros (%)**	0.0%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	42.682434	**Memory size**	67.6 KiB

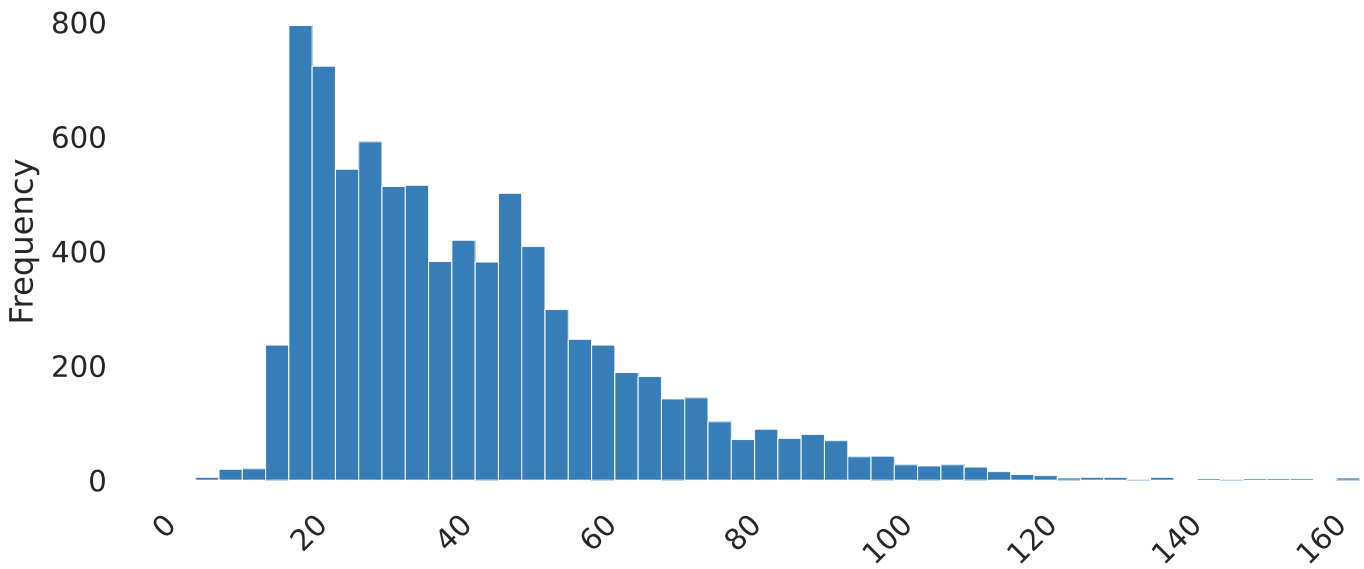


Quantile statistics

Minimum	4.2
5-th percentile	17.6
Q1	25
median	37.7
Q3	53.8
95-th percentile	87.8
Maximum	165.2
Range	161
Interquartile range (IQR)	28.8

Descriptive statistics

Standard deviation	22.475811
Coefficient of variation (CV)	0.52658222
Kurtosis	2.0823635
Mean	42.682434
Median Absolute Deviation (MAD)	14
Skewness	1.274659
Sum	351191.07
Variance	505.16208
Monotonicity	Not monotonic



Histogram with fixed size bins (bins=50)

Value	Count	Frequency
18.1	38	0.4
20.4	33	0.4
18.3	31	0.4
19.1	30	0.3

20.8	29	0.3
20.1	29	0.3
21.3	29	0.3
17.2	28	0.3
17.6	28	0.3
27.4	28	0.3
Other values (999)	7925	91.7%
(Missing)	415	4.8

Value	Count	Frequency (
4.2	2	< 0.1%
5.3	1	< 0.1%
6.7	1	< 0.1%
7.4	1	< 0.1%
7.5	1	< 0.1%
7.7	2	< 0.1%
7.9	2	< 0.1%
8.3	1	< 0.1%
8.5	2	< 0.1%
8.6	1	< 0.1%

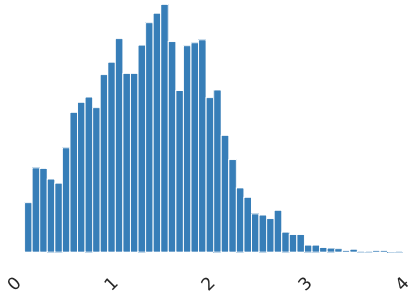
Value	Count	Frequency (
165.2	2	< 0.1%
163.3	1	< 0.1%
162.7	1	< 0.1%
159.6	1	< 0.1%
158.4	1	< 0.1%
157.8	1	< 0.1%
156.9	1	< 0.1%
155.3	2	< 0.1%
153.9	1	< 0.1%
151	1	< 0.1%

Singrauli, Surya Kiran Bhawan Dudhichua CO (mg/m3)

Real number (ℝ)

MISSING

Distinct	297	**Minimum**	0.1
Distinct (%)	3.6%	**Maximum**	4
Missing	496	**Zeros**	0
Missing (%)	5.7%	**Zeros (%)**	0.0%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	1.4086952	**Memory size**	67.6 KiB

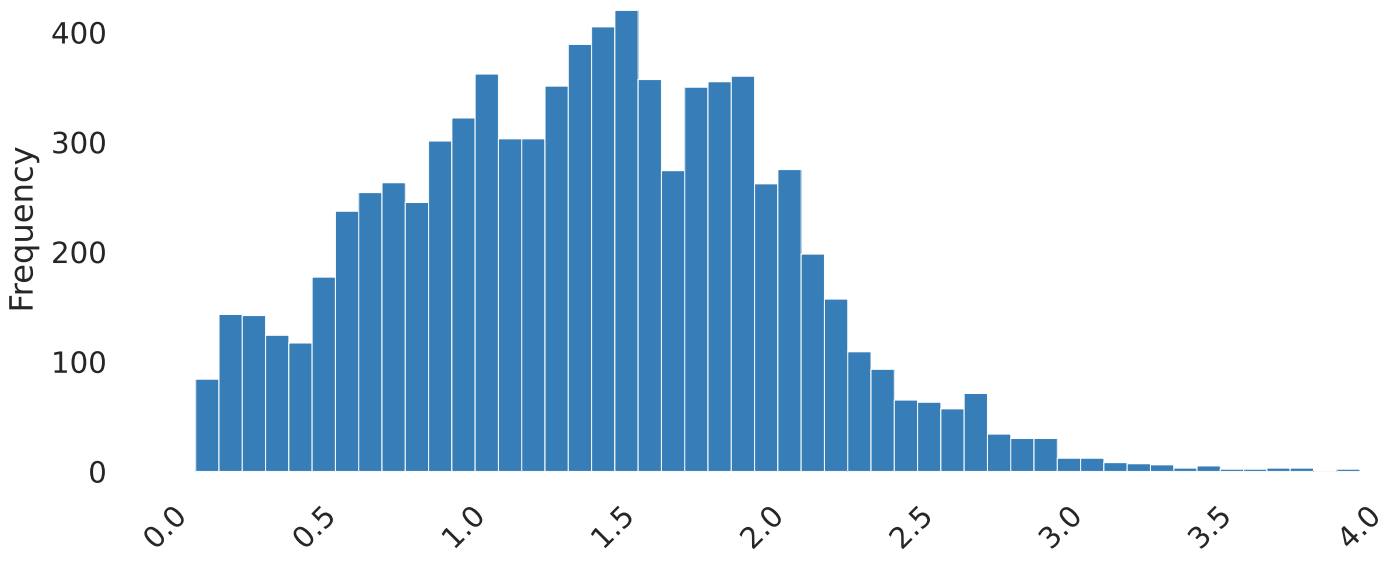


Quantile statistics

Minimum	0.1
5-th percentile	0.35
Q1	0.95
median	1.42
Q3	1.85
95-th percentile	2.44
Maximum	4
Range	3.9
Interquartile range (IQR)	0.9

Descriptive statistics

Standard deviation	0.63175921
Coefficient of variation (CV)	0.44847118
Kurtosis	-0.1761064
Mean	1.4086952
Median Absolute Deviation (MAD)	0.46
Skewness	0.19797051
Sum	11476.64
Variance	0.39911969
Monotonicity	Not monotonic



Histogram with fixed size bins (bins=50)

Value	Count	Frequency
1.49	72	0.8
1.51	71	0.8
1.48	70	0.8
1.4	69	0.8
1.59	68	0.8

1.05	66	0.8
1.58	65	0.8
1.82	64	0.7
1.98	63	0.7
1.23	62	0.7
Other values (287)	7477	86.5%
(Missing)	496	5.7

Value	Count	Frequency (
0.1	9	0.1%
0.11	9	0.1%
0.13	9	0.1%
0.14	12	0.1%
0.15	13	0.2%
0.16	17	0.2%
0.17	15	0.2%
0.18	11	0.1%
0.19	19	0.2%
0.21	17	0.2%

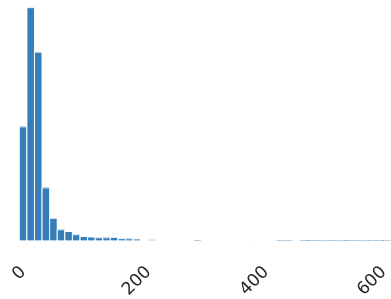
Value	Count	Frequency (
4	2	< 0.1%
3.82	1	< 0.1%
3.78	1	< 0.1%
3.77	1	< 0.1%
3.72	1	< 0.1%
3.71	1	< 0.1%
3.69	1	< 0.1%
3.63	1	< 0.1%
3.61	1	< 0.1%
3.56	1	< 0.1%

Singrauli, Surya Kiran Bhawan Dudhichua SO2 (碌g/m3)

Real number (ℝ)

MISSING

Distinct	1113	**Minimum**	0.1
Distinct (%)	15.5%	**Maximum**	645.6
Missing	1451	**Zeros**	0
Missing (%)	16.8%	**Zeros (%)**	0.0%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	34.312991	**Memory size**	67.6 KiB

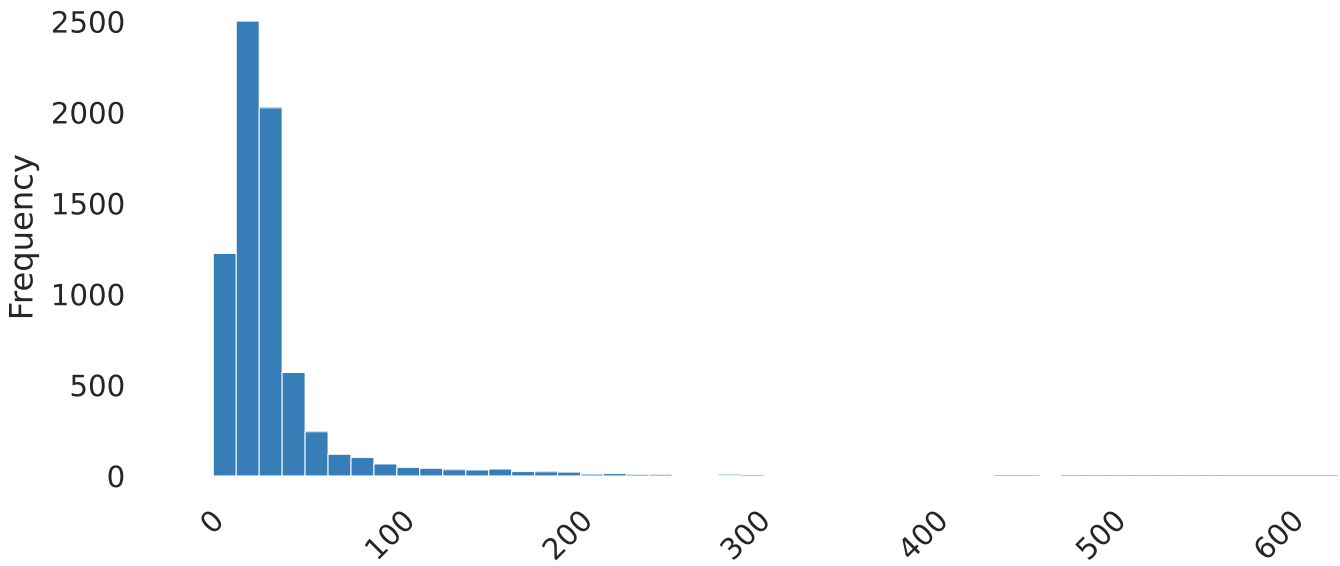


Quantile statistics

Minimum	0.1
5-th percentile	7.7
Q1	16.1
median	25.3
Q3	35.2
95-th percentile	100.825
Maximum	645.6
Range	645.5
Interquartile range (IQR)	19.1

Descriptive statistics

Standard deviation	40.099306
Coefficient of variation (CV)	1.1686334
Kurtosis	37.005787
Mean	34.312991
Median Absolute Deviation (MAD)	9.4
Skewness	4.9239013
Sum	246779.03
Variance	1607.9543
Monotonicity	Not monotonic



Histogram with fixed size bins (bins=50)

Value	Count	Frequency
18.9	36	0.4
26.1	35	0.4
19.2	35	0.4
27.2	34	0.4
27.7	33	0.4

17.6	32	0.4
29.6	31	0.4
27.9	29	0.3
29.8	28	0.3
27.5	28	0.3
Other values (1103)	6871	79.5%
(Missing)	1451	16.8

Value	Count	Frequency (
0.1	2	< 0.1%
0.2	2	< 0.1%
0.4	2	< 0.1%
0.5	1	< 0.1%
0.6	1	< 0.1%
0.8	3	< 0.1%
1.1	3	< 0.1%
1.2	4	< 0.1%
1.3	4	< 0.1%
1.5	4	< 0.1%

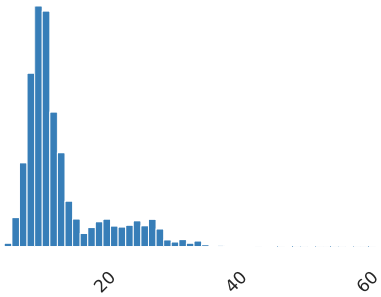
Value	Count	Frequency (
645.6	2	< 0.1%
475.1	1	< 0.1%
428.6	1	< 0.1%
419.4	1	< 0.1%
407.6	1	< 0.1%
393.2	1	< 0.1%
374.8	1	< 0.1%
373.6	1	< 0.1%
371.3	1	< 0.1%
368.1	1	< 0.1%

Singrauli, Surya Kiran Bhawan Dudhichua NH3 (碌g/m3)

Real number (ℝ)

HIGH CORRELATION (This variable has a high overall correlation with 1 fields: #) MISSING

Distinct	334	**Minimum**	4.6
Distinct (%)	4.0%	**Maximum**	62.4
Missing	326	**Zeros**	0
Missing (%)	3.8%	**Zeros (%)**	0.0%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	13.247534	**Memory size**	67.6 KiB

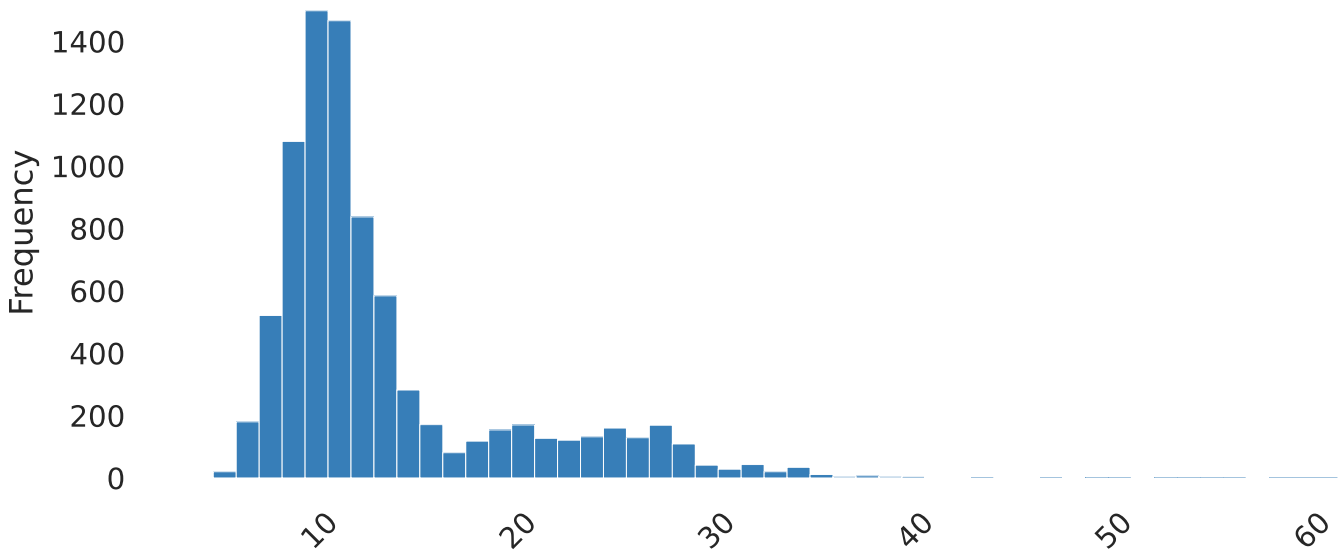


Quantile statistics

Minimum	4.6
5-th percentile	7.5
Q1	9.4
median	11
Q3	14
95-th percentile	27
Maximum	62.4
Range	57.8
Interquartile range (IQR)	4.6

Descriptive statistics

Standard deviation	6.1742296
Coefficient of variation (CV)	0.46606633
Kurtosis	3.5973951
Mean	13.247534
Median Absolute Deviation (MAD)	1.9
Skewness	1.7981363
Sum	110179.74
Variance	38.121111
Monotonicity	Not monotonic



Histogram with fixed size bins (bins=50)

Value	Count	Frequency
9.8	161	1.9
10.1	152	1.8
9.4	149	1.7
10.5	145	1.7
10	143	1.7

9.9	141	1.6
9.3	139	1.6
11.2	136	1.6
9.1	135	1.6
10.6	135	1.6
Other values (324)	6881	79.6%
(Missing)	326	3.8

Value	Count	Frequency (
4.6	2	< 0.1%
4.7	1	< 0.1%
4.8	1	< 0.1%
5	1	< 0.1%
5.1	1	< 0.1%
5.3	2	< 0.1%
5.4	3	< 0.1%
5.5	4	< 0.1%
5.6	4	< 0.1%
5.7	1	< 0.1%

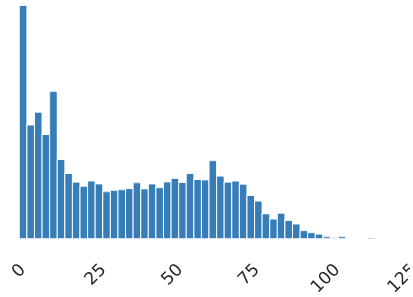
Value	Count	Frequency (
62.4	2	< 0.1%
56.9	1	< 0.1%
51.8	1	< 0.1%
51.3	1	< 0.1%
47.7	1	< 0.1%
47.4	1	< 0.1%
45.9	1	< 0.1%
44.5	1	< 0.1%
42.6	1	< 0.1%
42.3	1	< 0.1%

Singrauli, Surya Kiran Bhawan Dudhichua Ozone (碌g/m3)

Real number (ℝ)

HIGH CORRELATION (This variable has a high overall correlation with 3 fields: Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua NO (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb)) MISSING

Distinct	973	**Minimum**	0.1
Distinct (%)	11.9%	**Maximum**	123.8
Missing	453	**Zeros**	0
Missing (%)	5.2%	**Zeros (%)**	0.0%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	35.632958	**Memory size**	67.6 KiB

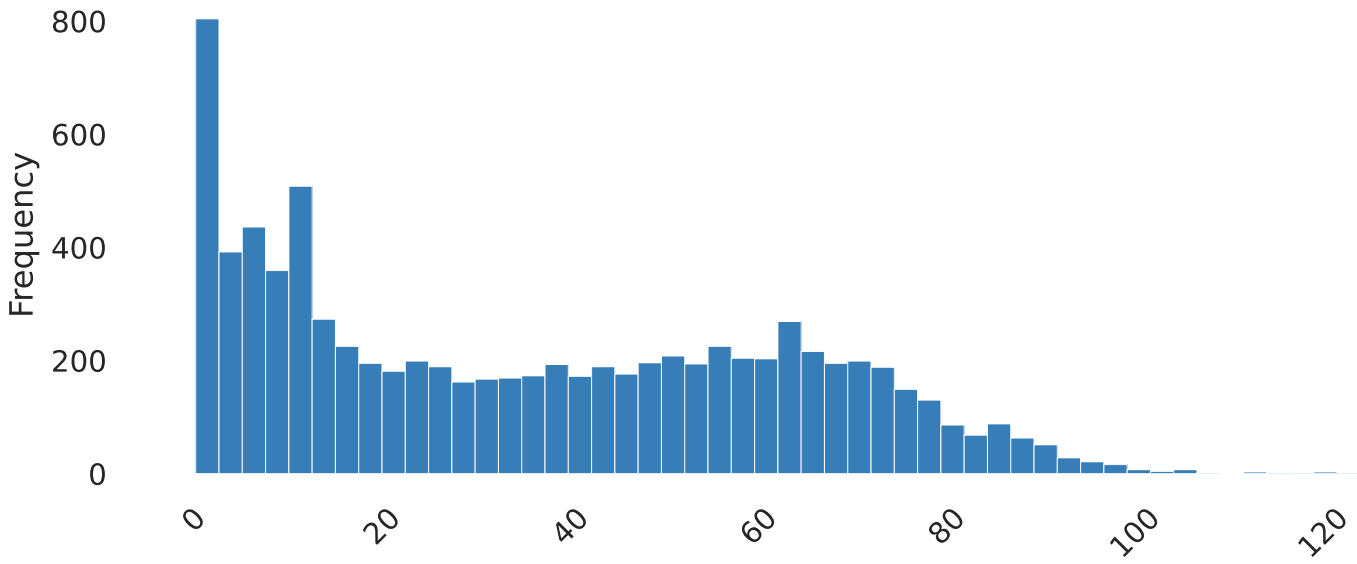


Quantile statistics

Minimum	0.1
5-th percentile	1.7
Q1	10.5
median	32.4
Q3	58.8
95-th percentile	80.7
Maximum	123.8
Range	123.7
Interquartile range (IQR)	48.3

Descriptive statistics

Standard deviation	27.03416
Coefficient of variation (CV)	0.75868412
Kurtosis	-1.0757415
Mean	35.632958
Median Absolute Deviation (MAD)	23.7
Skewness	0.36136861
Sum	291833.93
Variance	730.84579
Monotonicity	Not monotonic



Histogram with fixed size bins (bins=50)

Value	Count	Frequency
1.7	78	0.9
1.5	73	0.8
1.6	67	0.8
1.9	63	0.7

1.4	60	0.7
1.8	57	0.7
2	55	0.6
5.3	51	0.6
5.4	44	0.5
1.2	43	0.5
Other values (963)	7599	87.9%
(Missing)	453	5.2

Value	Count	Frequency (
0.1	7	0.1%
0.2	7	0.1%
0.3	3	< 0.1%
0.4	3	< 0.1%
0.5	8	0.1%
0.6	6	0.1%
0.7	5	0.1%
0.8	11	0.1%
0.9	14	0.2%
1	21	0.2%

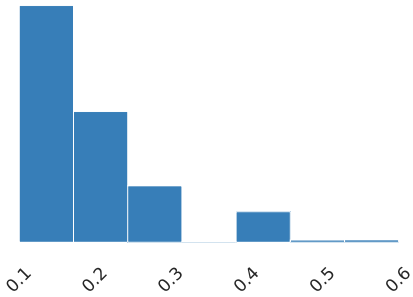
Value	Count	Frequency (
123.8	2	< 0.1%
120.5	1	< 0.1%
120.2	2	< 0.1%
118.1	1	< 0.1%
117.3	1	< 0.1%
113.6	1	< 0.1%
112.1	1	< 0.1%
112	1	< 0.1%
109	1	< 0.1%
108.8	1	< 0.1%

Singrauli, Surya Kiran Bhawan Dudhichua Benzene (碌g/m3)

Real number (ℝ)

HIGH CORRELATION (This variable has a high overall correlation with 2 fields: Singrauli, Surya Kiran Bhawan Dudhichua PM10 (碌g/m3), Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (碌g/m3)) MISSING

Distinct	7	**Minimum**	0.1
Distinct (%)	0.3%	**Maximum**	0.6
Missing	6195	**Zeros**	0
Missing (%)	71.7%	**Zeros (%)**	0.0%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	0.17764706	**Memory size**	67.6 KiB

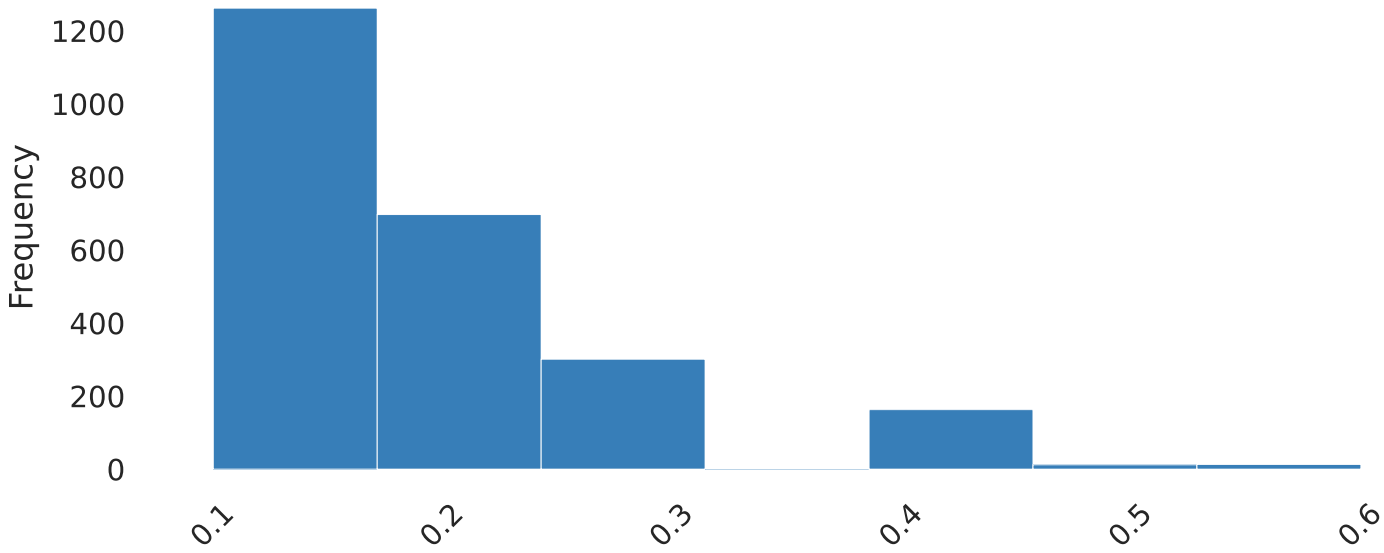


Quantile statistics

Minimum	0.1
5-th percentile	0.1
Q1	0.1
median	0.1
Q3	0.2
95-th percentile	0.4
Maximum	0.6
Range	0.5
Interquartile range (IQR)	0.1

Descriptive statistics

Standard deviation	0.099214831
Coefficient of variation (CV)	0.55849408
Kurtosis	1.457482
Mean	0.17764706
Median Absolute Deviation (MAD)	0
Skewness	1.3156533
Sum	434.88
Variance	0.0098435828
Monotonicity	Not monotonic



Histogram with fixed size bins (bins=7)

Value	Count	Frequency
0.1	1262	14.6
0.2	696	8.1
0.3	301	3.5
0.4	163	1.9

0.6	13	0.2
0.5	12	0.1
0.18	1	< 0.1
(Missing)	6195	71.7%

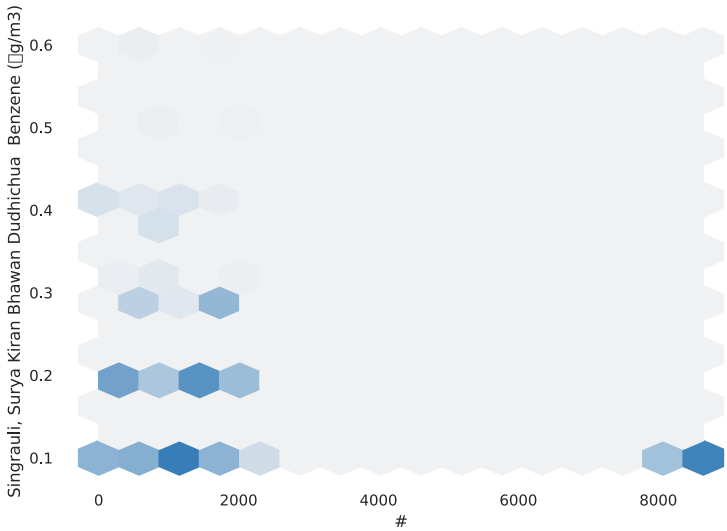
Value	Count	Frequency (
0.1	1262	14.6%
0.18	1	< 0.1%
0.2	696	8.1%
0.3	301	3.5%
0.4	163	1.9%
0.5	12	0.1%
0.6	13	0.2%

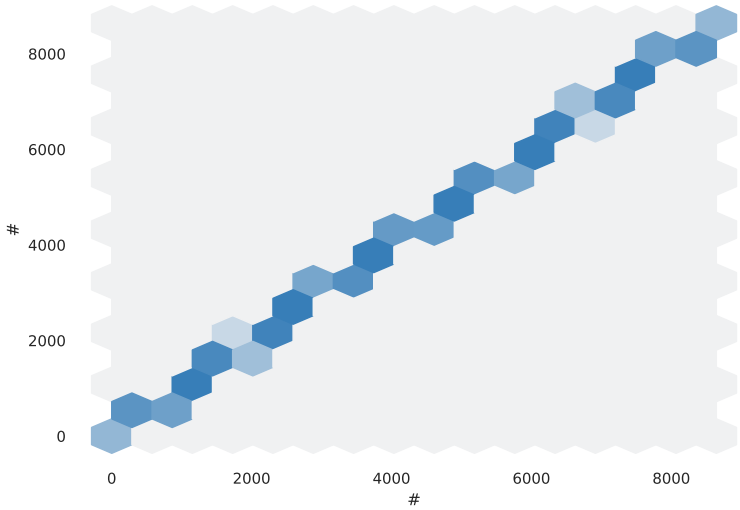
Value	Count	Frequency (
0.6	13	0.2%
0.5	12	0.1%
0.4	163	1.9%
0.3	301	3.5%
0.2	696	8.1%
0.18	1	< 0.1%
0.1	1262	14.6%

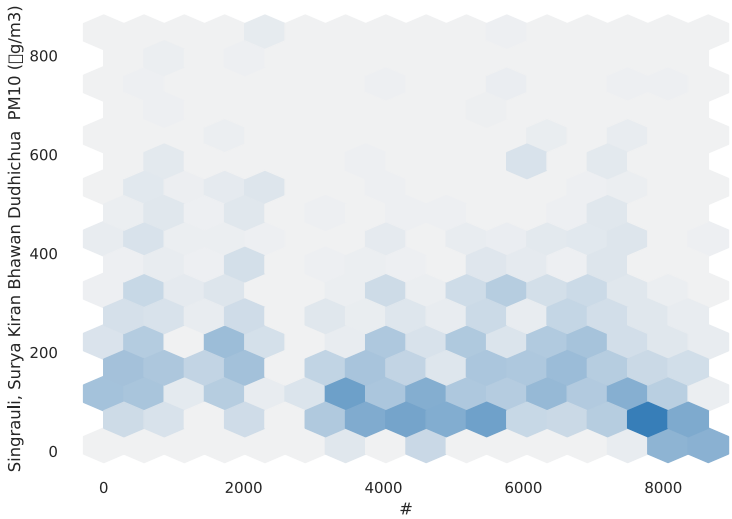
Interactions

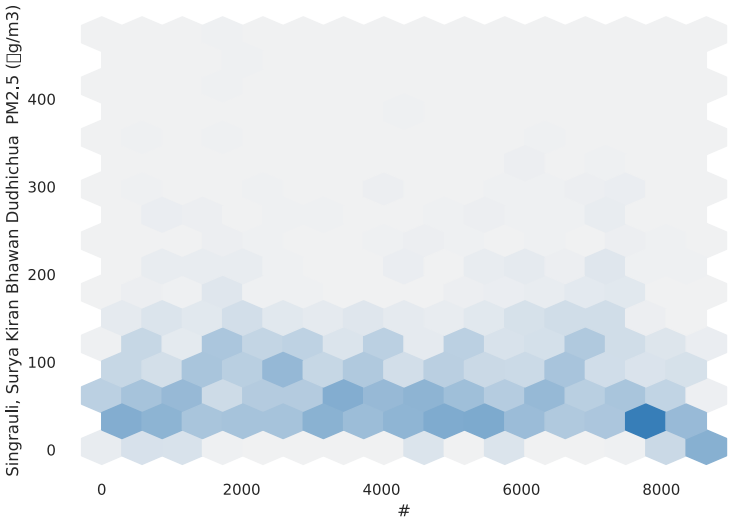
#
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha

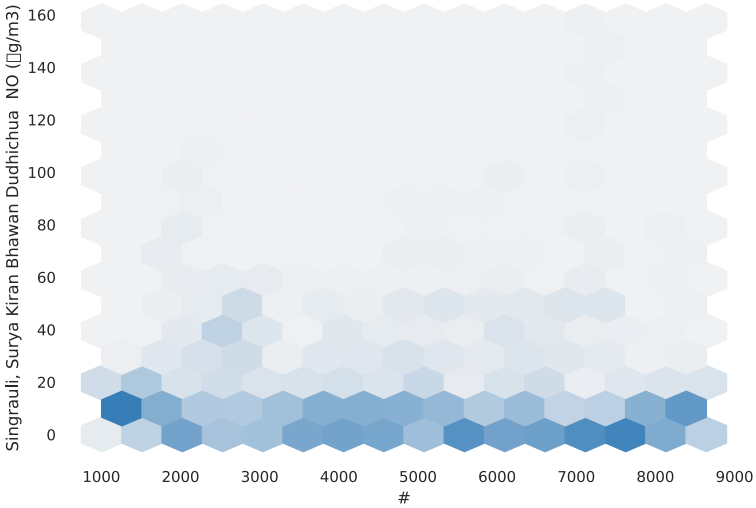
Singrauli, Surya Kiran Bha
#
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha

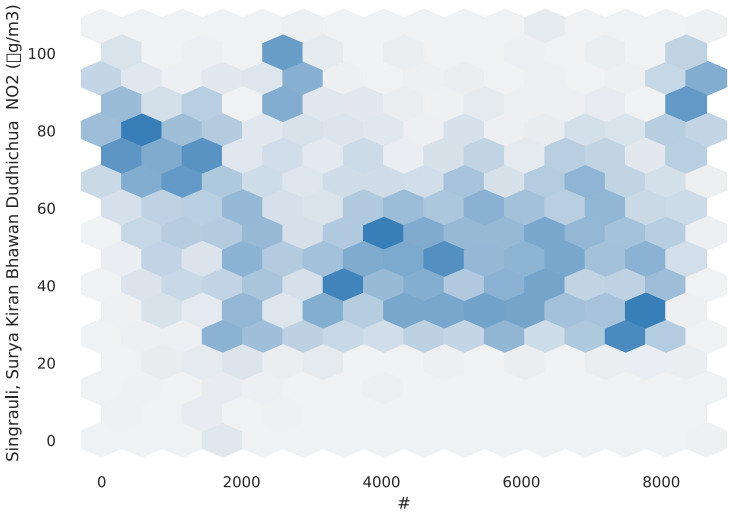


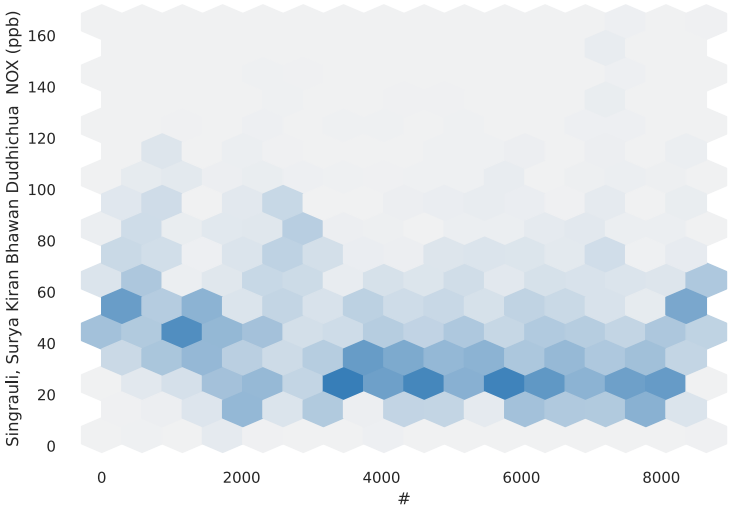


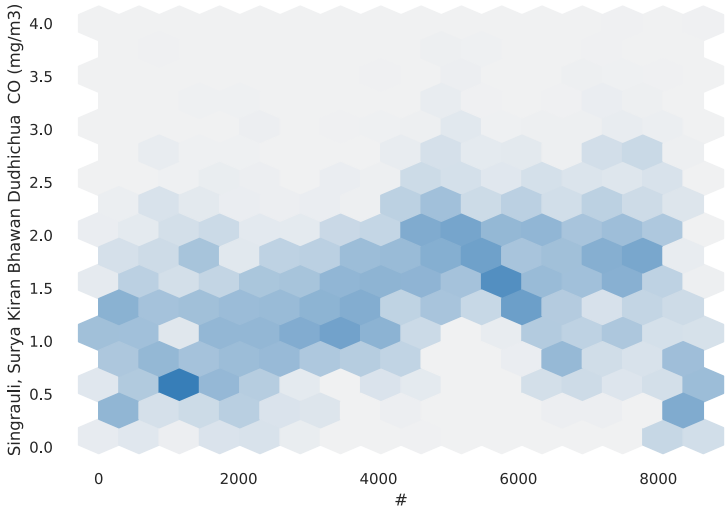


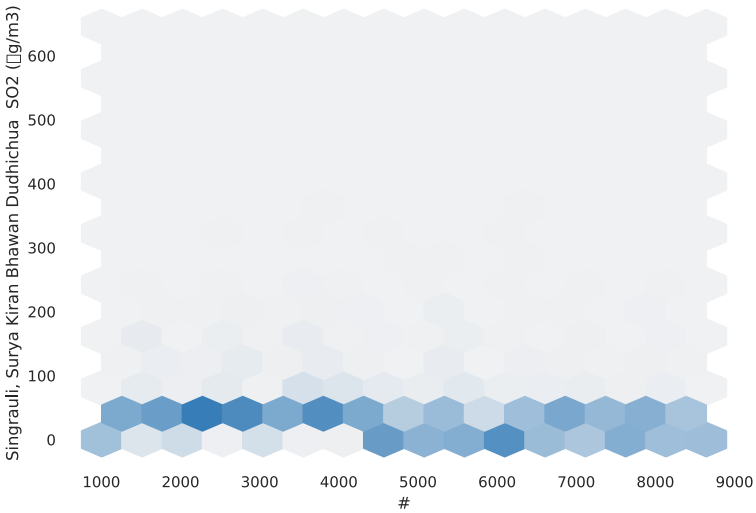

Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (g/m3)
400
300
200
100
0
0
2000
4000
6000
8000
#

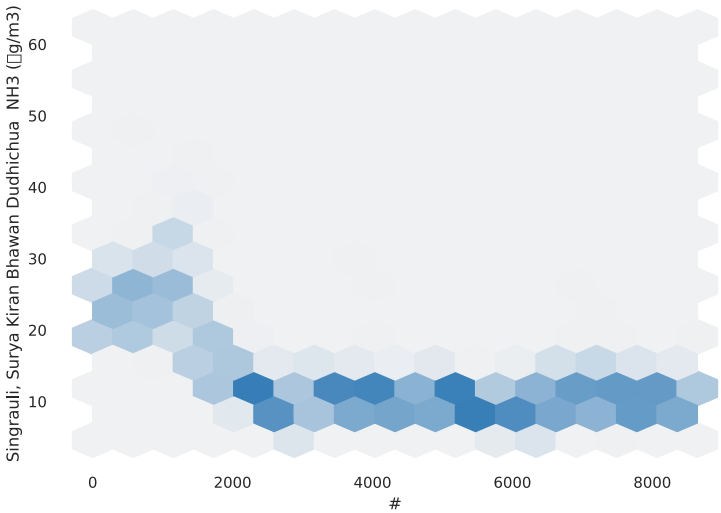

Singrauli, Surya Kiran Bhawan Dudhichua NO (g/m3)
160
140
120
100
80
60
40
20
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
#

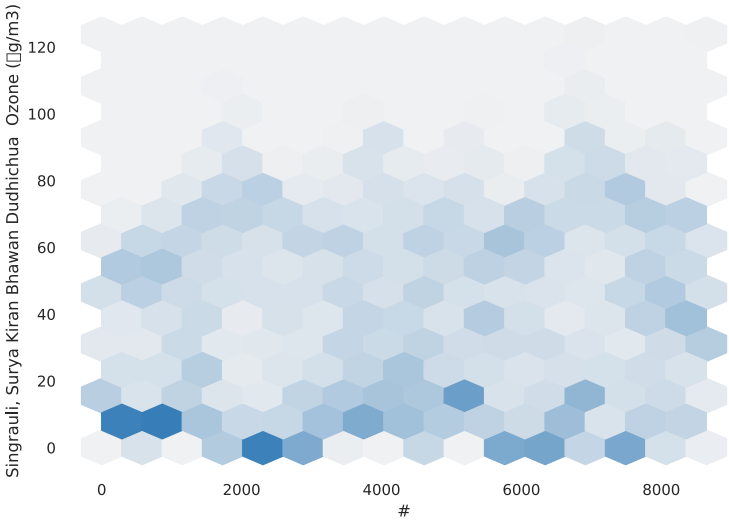

Singrauli, Surya Kiran Bhawan Dudhichua NO2 (g/m3)
100
80
60
40
20
0
0
2000
4000
6000
8000
#


Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb)
160
140
120
100
80
60
40
20
0
0
2000
4000
6000
8000
#

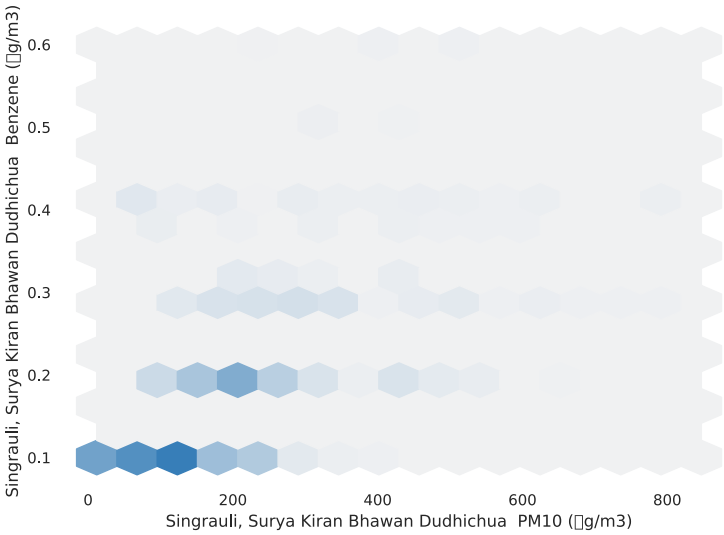

Singrauli, Surya Kiran Bhawan Dudhichua CO (mg/m3)
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
2000
4000
6000
8000
#

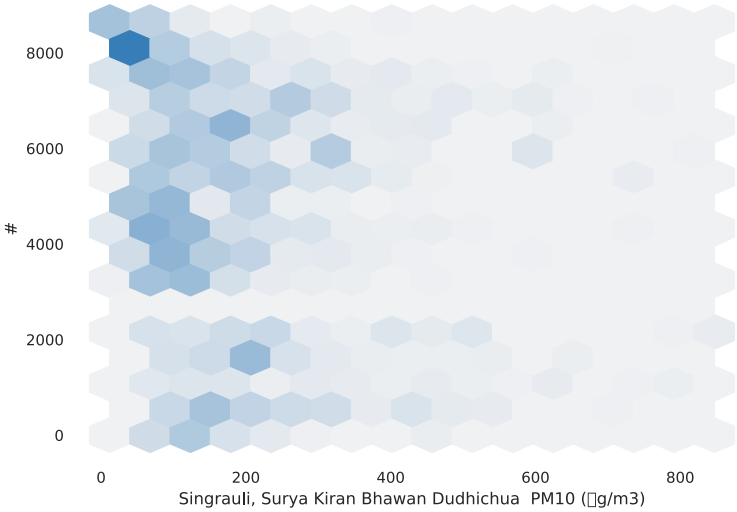

Singrauli, Surya Kiran Bhawan Dudhichua SO2 (µg/m3)
600
500
400
300
200
100
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
#

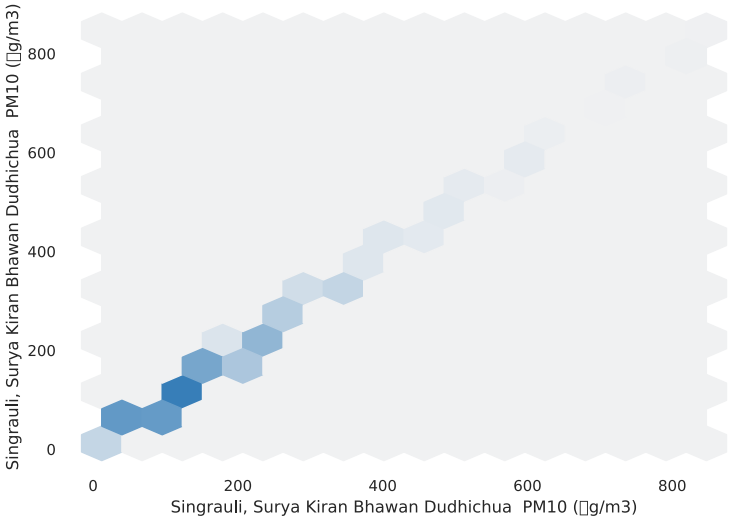


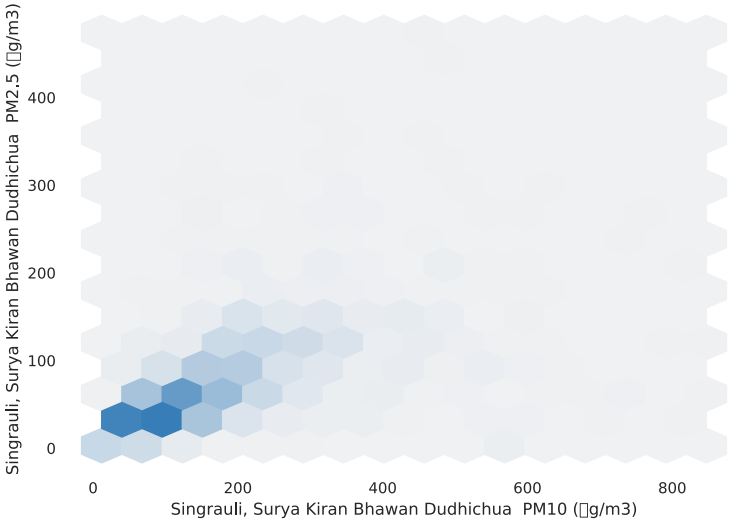


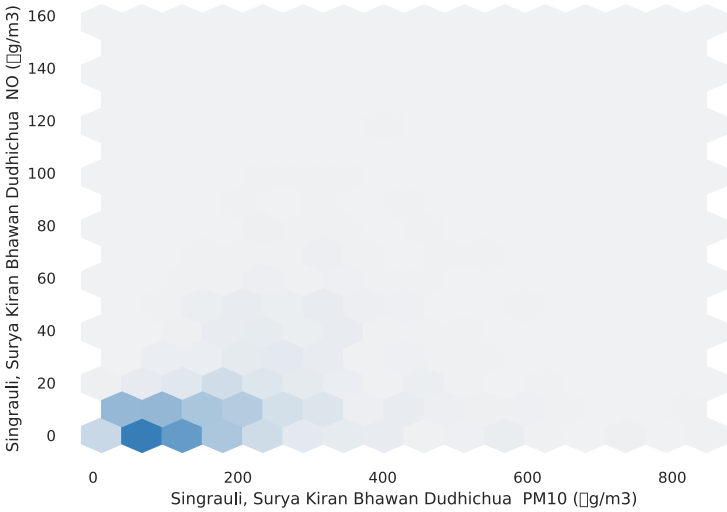
Singrauli, Surya Kiran Bha
#
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha

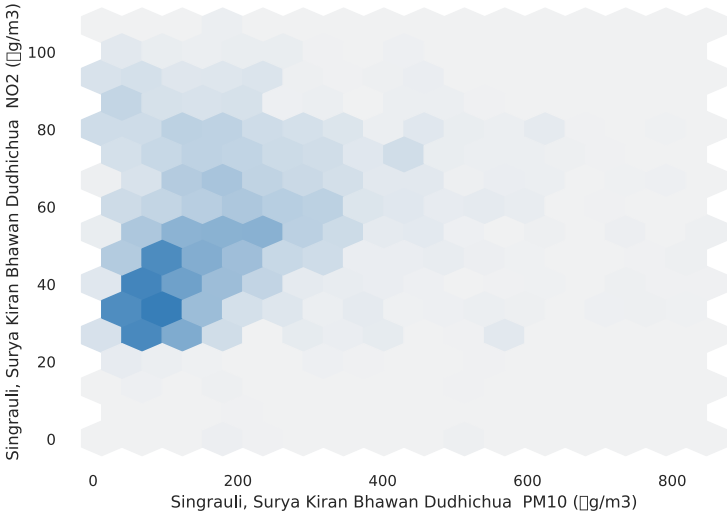


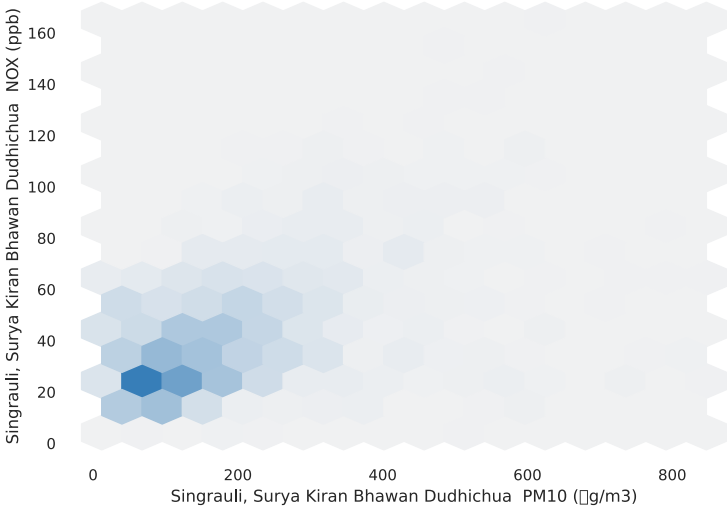

#
8000
6000
4000
2000
0
0
200
400
600
800
Singrauli, Surya Kiran Bhawan Dudhichua PM10 (□g/m3)

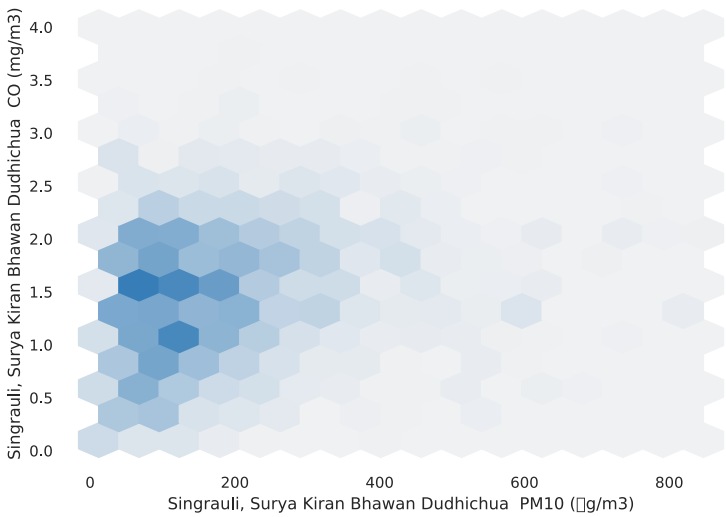

Singrauli, Surya Kiran Bhawan Dudhichua PM10 (□g/m3)
800
600
400
200
0
0
200
400
600
800
Singrauli, Surya Kiran Bhawan Dudhichua PM10 (□g/m3)

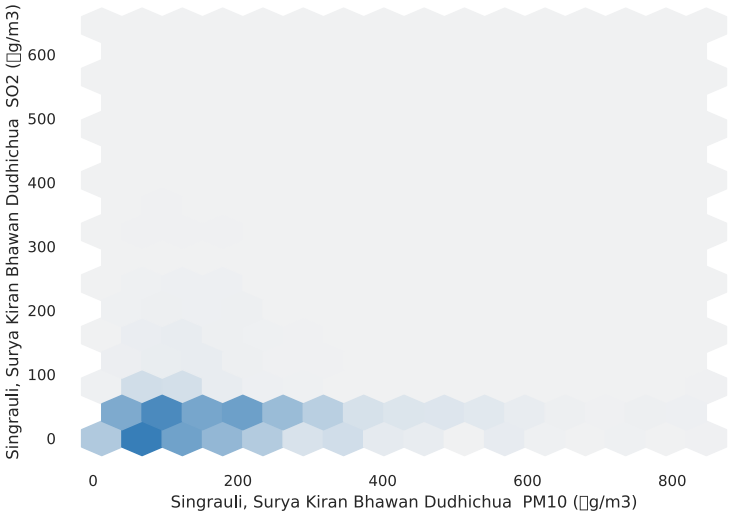

Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (□g/m3)
400
300
200
100
0
0
200
400
600
800
Singrauli, Surya Kiran Bhawan Dudhichua PM10 (□g/m3)

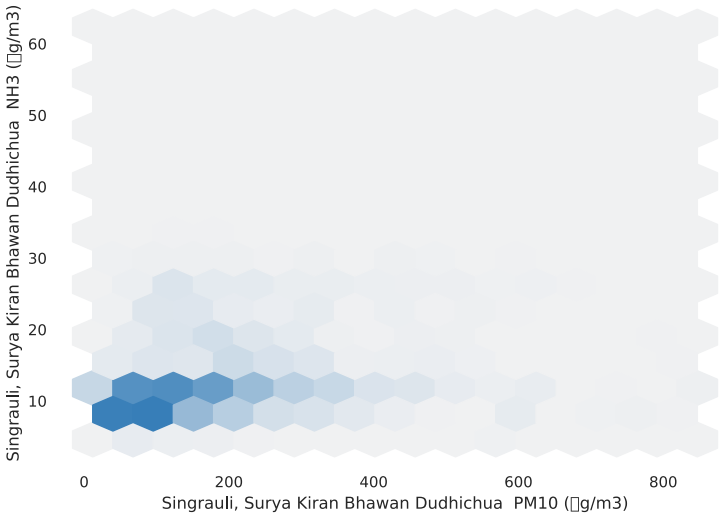

Singrauli, Surya Kiran Bhawan Dudhichua NO (□g/m3)
160
140
120
100
80
60
40
20
0
0
200
400
600
800
Singrauli, Surya Kiran Bhawan Dudhichua PM10 (□g/m3)

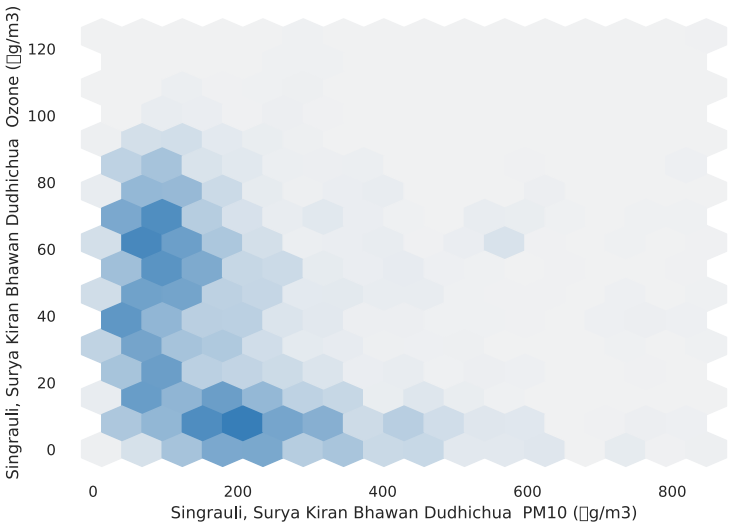

Singrauli, Surya Kiran Bhawan Dudhichua NO2 (□g/m3)
100
80
60
40
20
0
0
200
400
600
800
Singrauli, Surya Kiran Bhawan Dudhichua PM10 (□g/m3)


Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb)
160
140
120
100
80
60
40
20
0
0
200
400
600
800
Singrauli, Surya Kiran Bhawan Dudhichua PM10 (□g/m3)

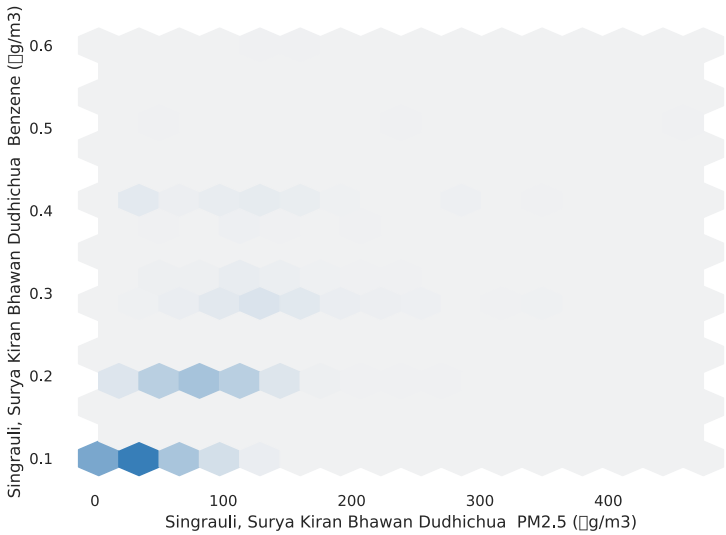

Singrauli, Surya Kiran Bhawan Dudhichua CO (mg/m3)
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
200
400
600
800
Singrauli, Surya Kiran Bhawan Dudhichua PM10 (□g/m3)

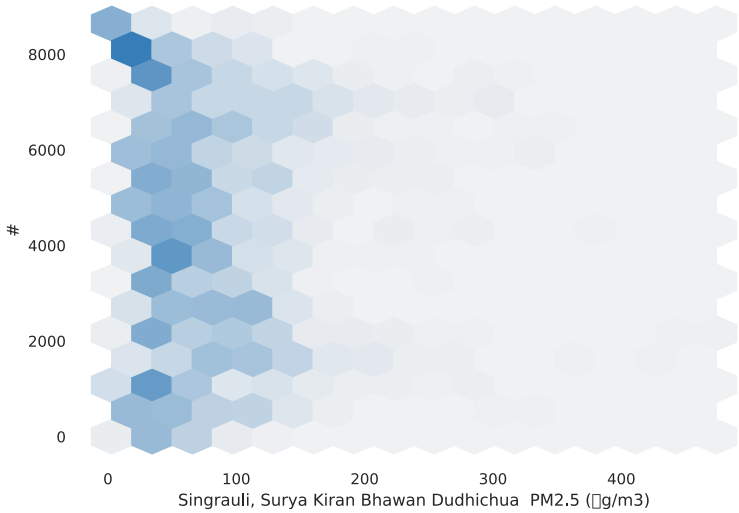

Singrauli, Surya Kiran Bhawan Dudhichua SO2 (□g/m3)
600
500
400
300
200
100
0
0
200
400
600
800
Singrauli, Surya Kiran Bhawan Dudhichua PM10 (□g/m3)

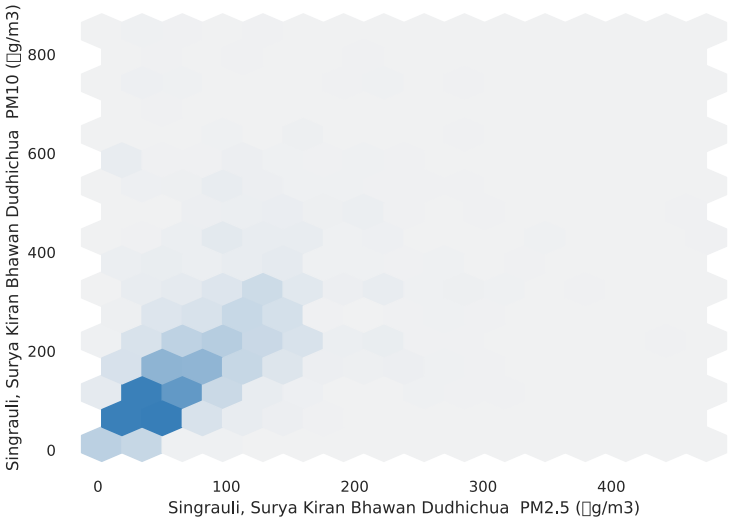

Singrauli, Surya Kiran Bhawan Dudhichua NH3 (□g/m3)
60
50
40
30
20
10
0
200
400
600
800
Singrauli, Surya Kiran Bhawan Dudhichua PM10 (□g/m3)

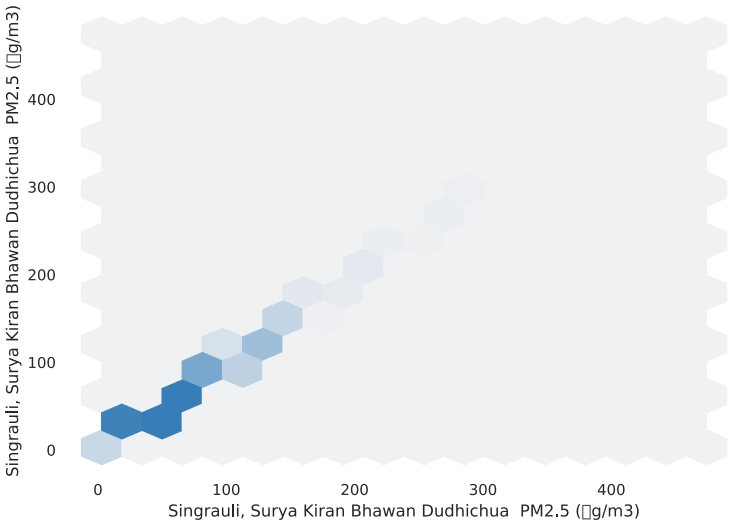


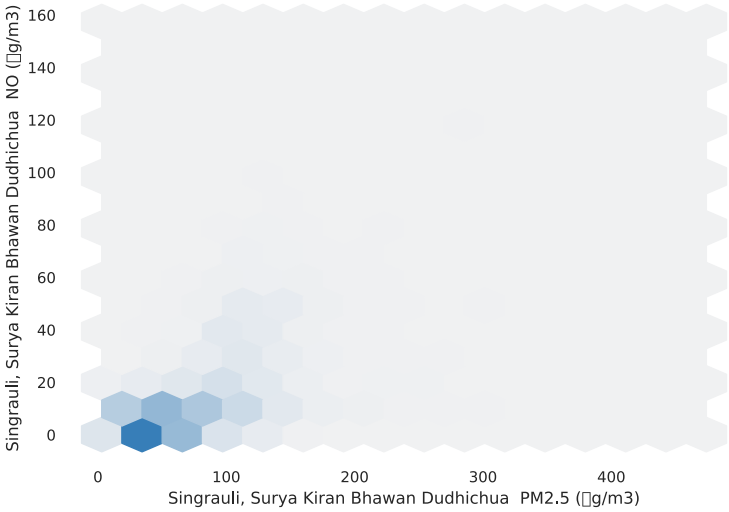
Singrauli, Surya Kiran Bha
#
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha

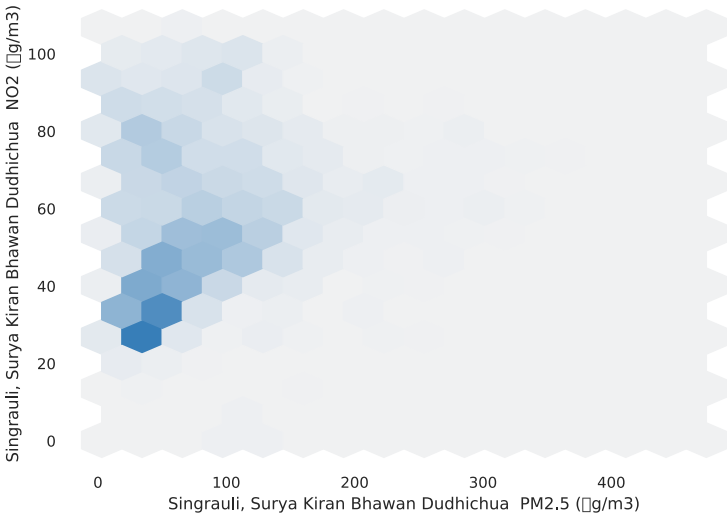


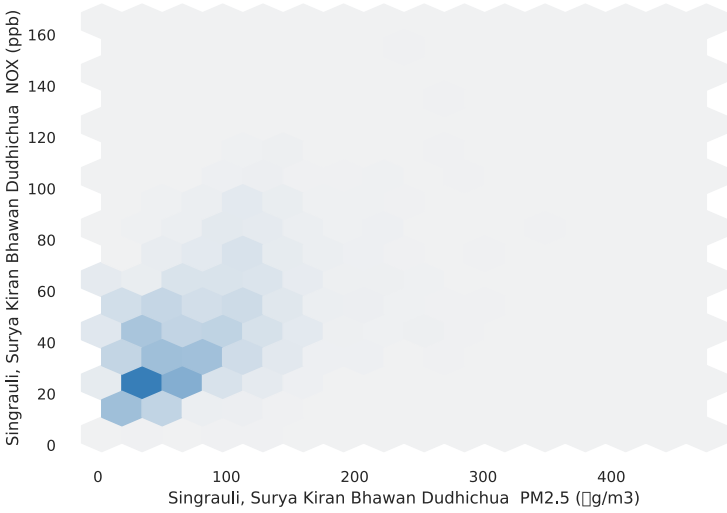


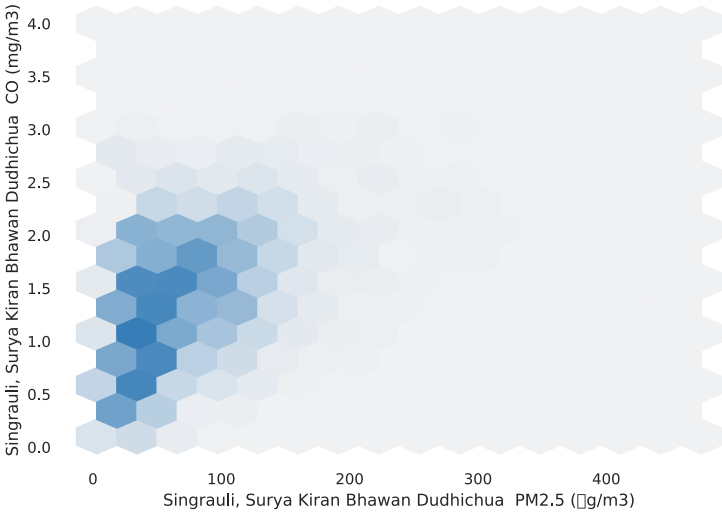

Singrauli, Surya Kiran Bhawan Dudhichua PM10 (g/m3)
800
600
400
200
0
0
100
200
300
400
Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (g/m3)

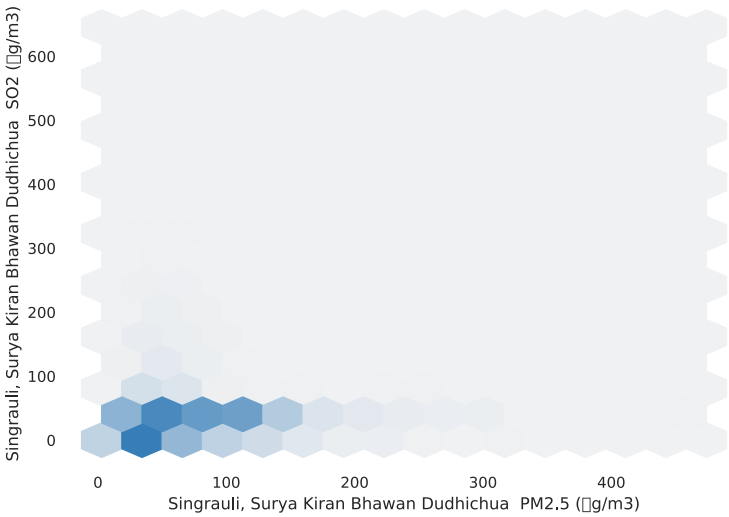

Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (g/m3)
400
300
200
100
0
0
100
200
300
400
Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (g/m3)

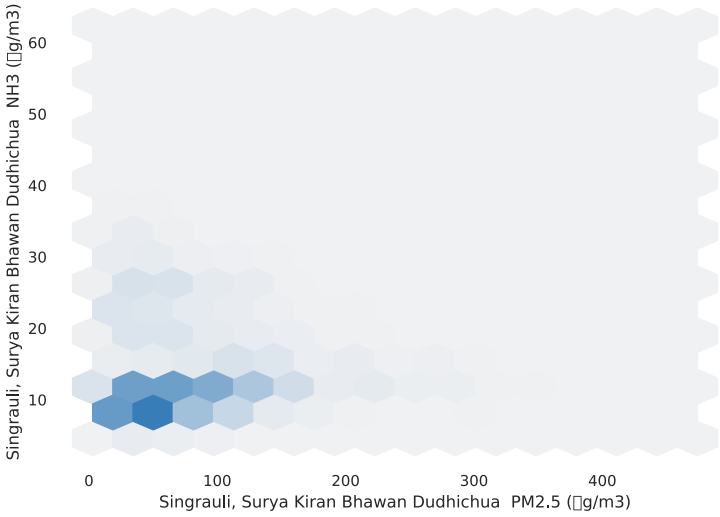

Singrauli, Surya Kiran Bhawan Dudhichua NO (g/m3)
160
140
120
100
80
60
40
20
0
0
100
200
300
400
Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (g/m3)

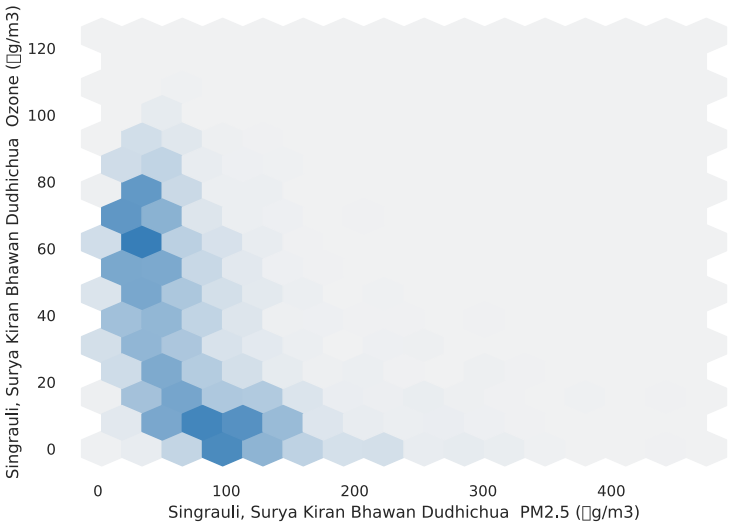

Singrauli, Surya Kiran Bhawan Dudhichua NO2 (g/m3)
0
20
40
60
80
100
0
100
200
300
400
Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (g/m3)


Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb)
0
20
40
60
80
100
120
140
160
0
100
200
300
400
Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (g/m3)


Singrauli, Surya Kiran Bhawan Dudhichua CO (mg/m3)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
100
200
300
400
Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (g/m3)

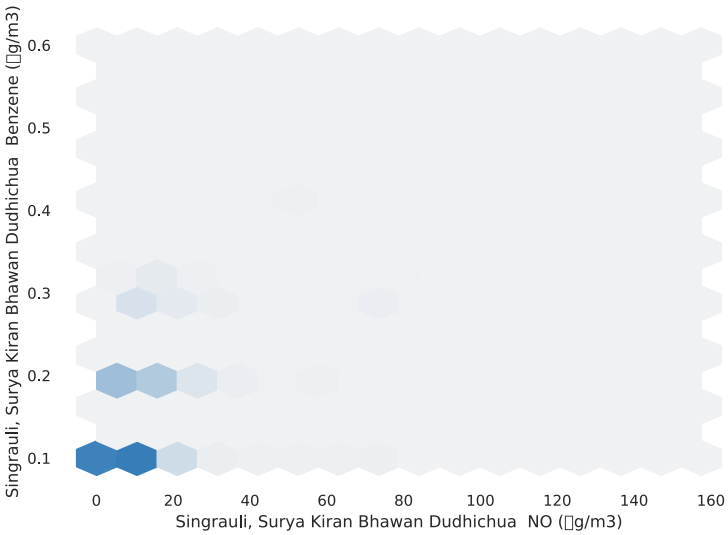


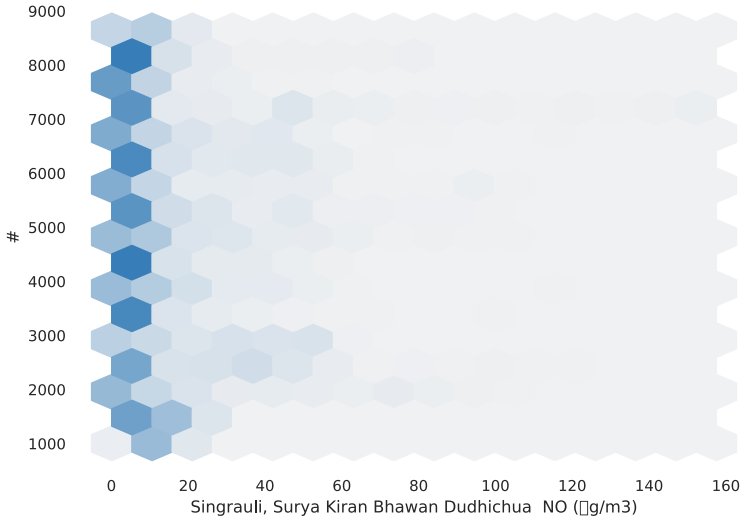


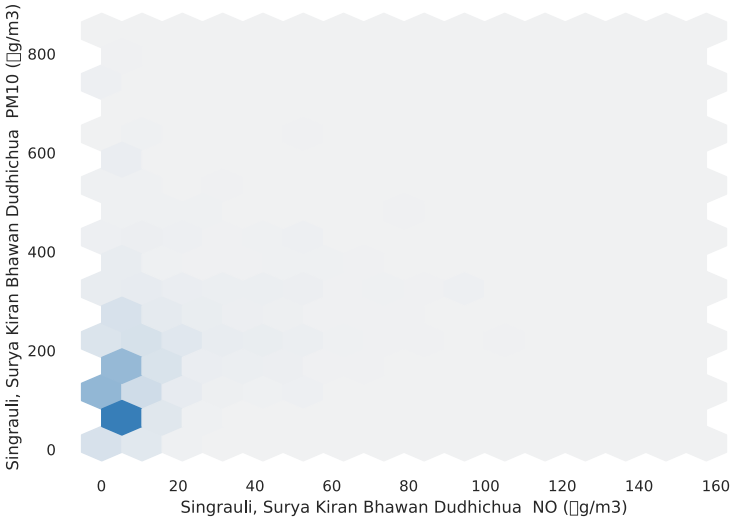


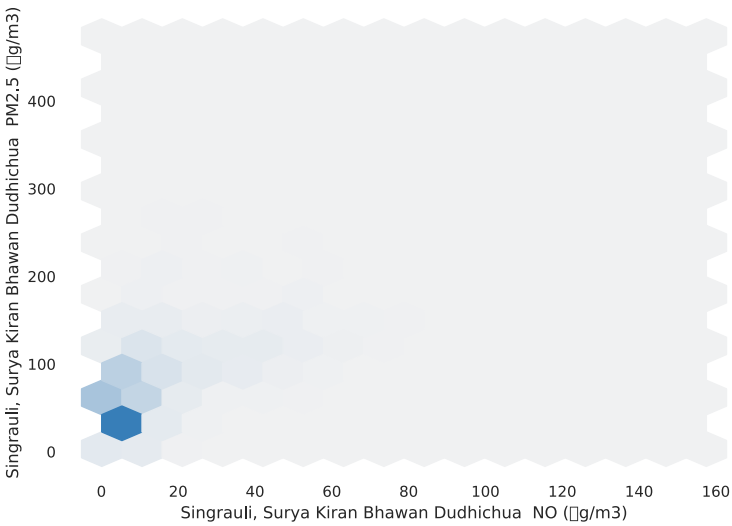
Singrauli, Surya Kiran Bha
#
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha

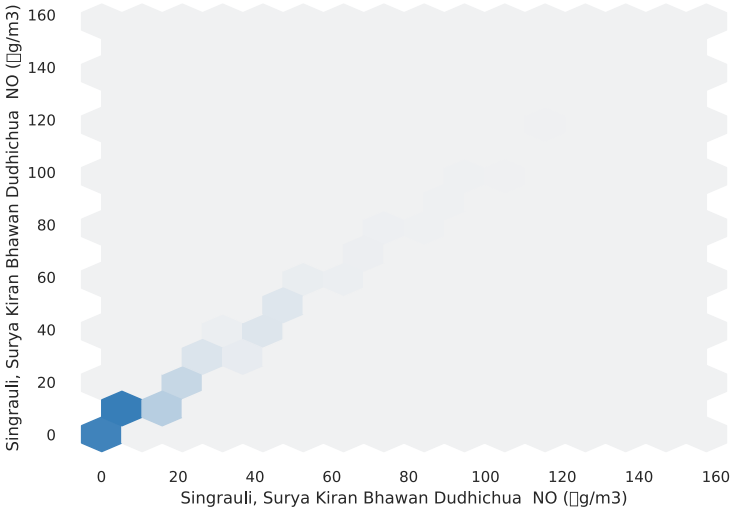
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha

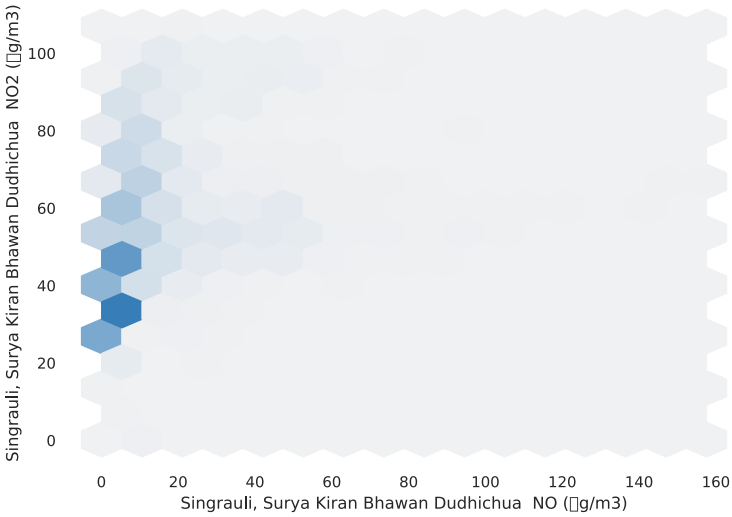


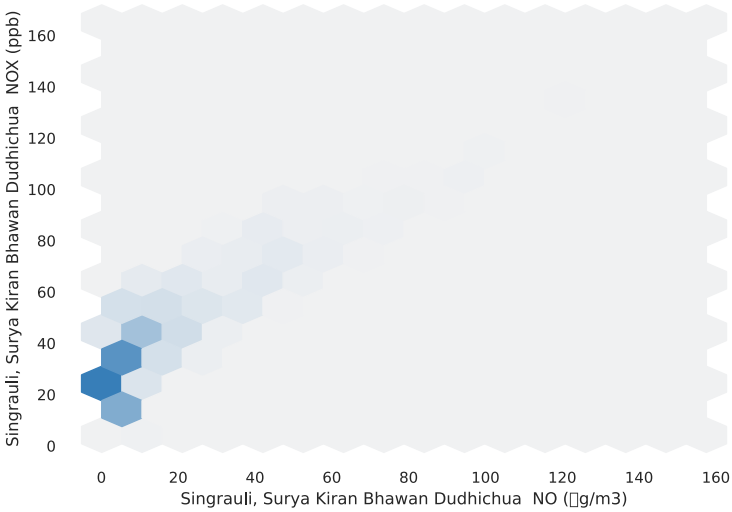


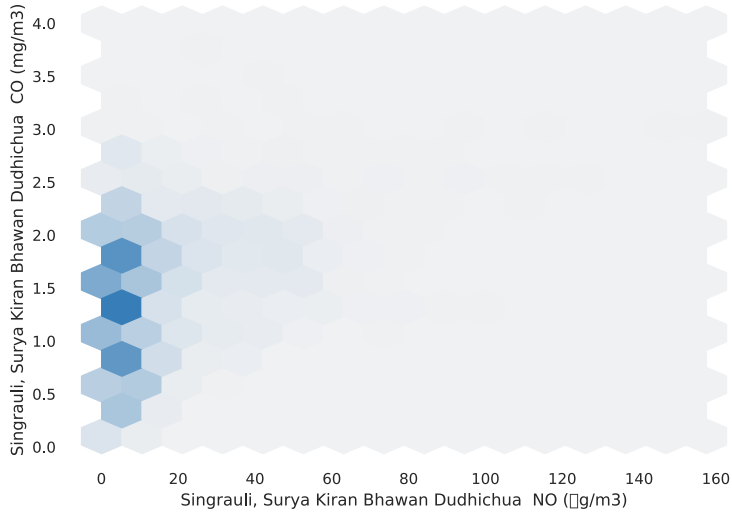


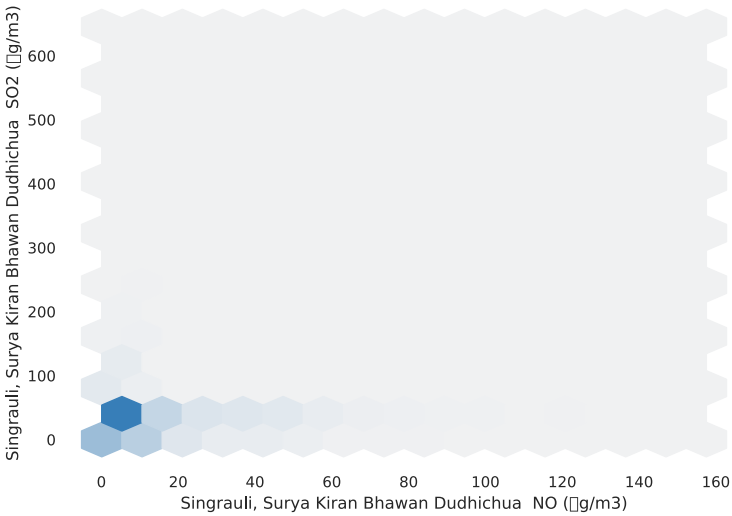

Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (g/m3)
0
100
200
300
400
0
20
40
60
80
100
120
140
160
Singrauli, Surya Kiran Bhawan Dudhichua NO (g/m3)

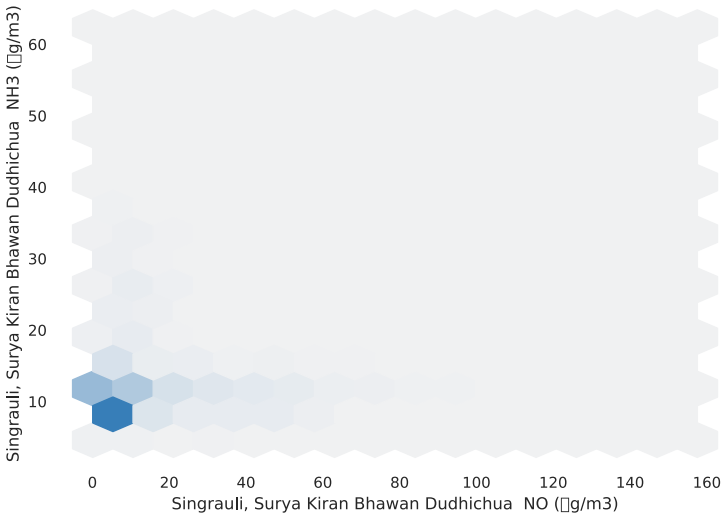

Singrauli, Surya Kiran Bhawan Dudhichua NO (g/m3)
0
20
40
60
80
100
120
140
160
0
20
40
60
80
100
120
140
160
Singrauli, Surya Kiran Bhawan Dudhichua NO (g/m3)

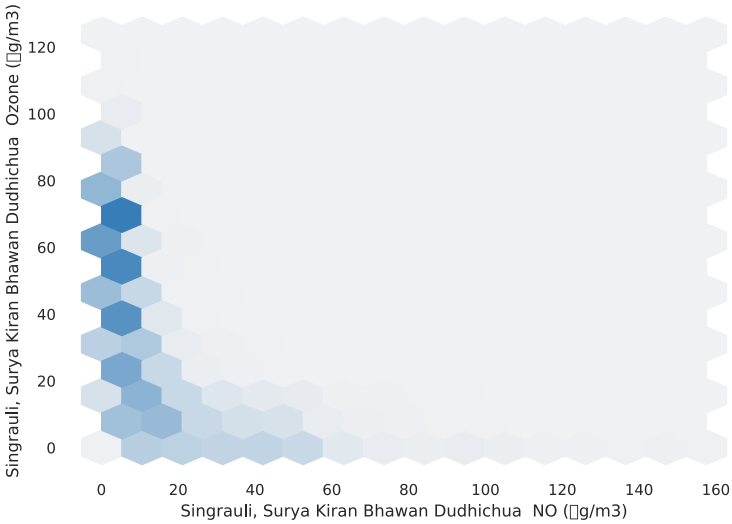

Singrauli, Surya Kiran Bhawan Dudhichua NO2 (g/m3)
0
20
40
60
80
100
0
20
40
60
80
100
120
140
160
Singrauli, Surya Kiran Bhawan Dudhichua NO (g/m3)


Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb)
0
20
40
60
80
100
120
140
160
0
20
40
60
80
100
120
140
160
Singrauli, Surya Kiran Bhawan Dudhichua NO (□g/m3)

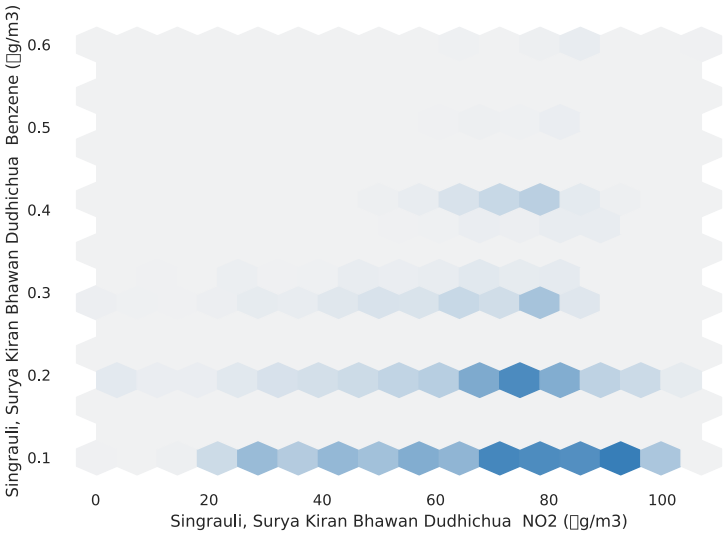

Singrauli, Surya Kiran Bhawan Dudhichua CO (mg/m3)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
20
40
60
80
100
120
140
160
Singrauli, Surya Kiran Bhawan Dudhichua NO (□g/m3)

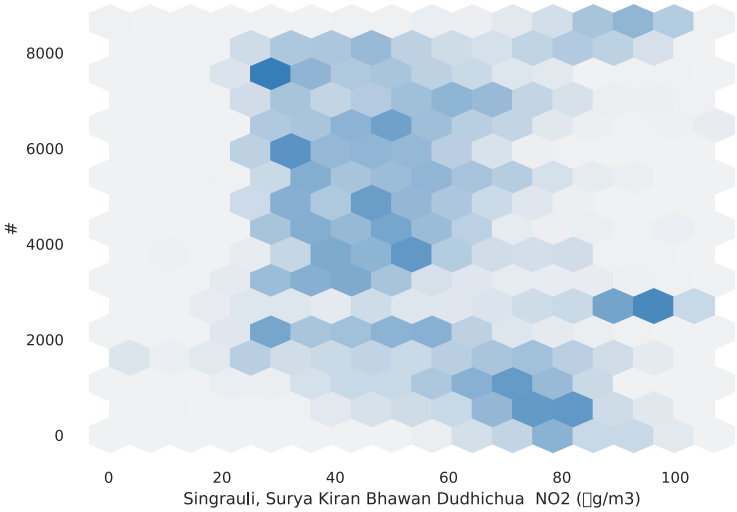

Singrauli, Surya Kiran Bhawan Dudhichua SO2 (□g/m3)
0
100
200
300
400
500
600
0
20
40
60
80
100
120
140
160
Singrauli, Surya Kiran Bhawan Dudhichua NO (□g/m3)

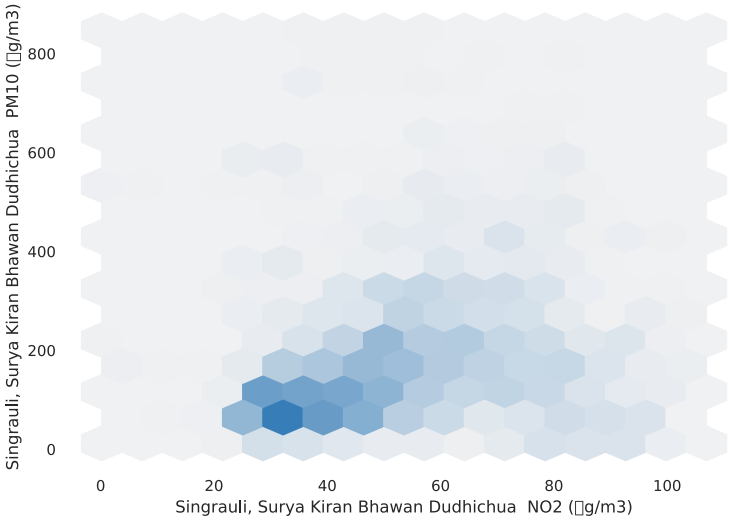


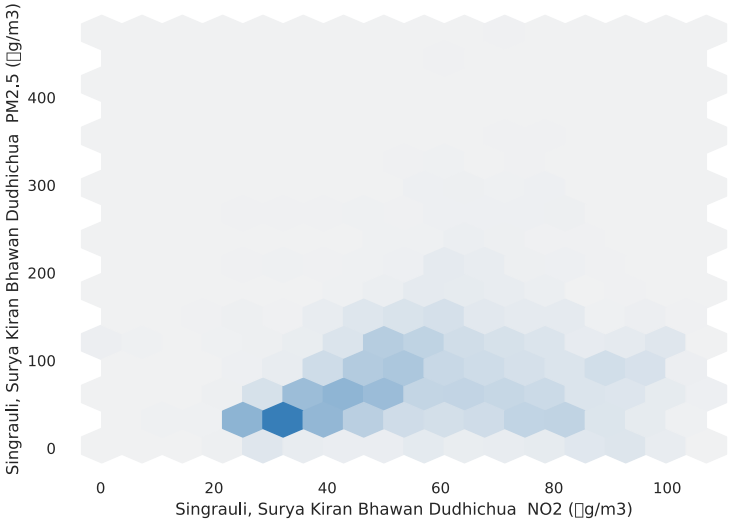


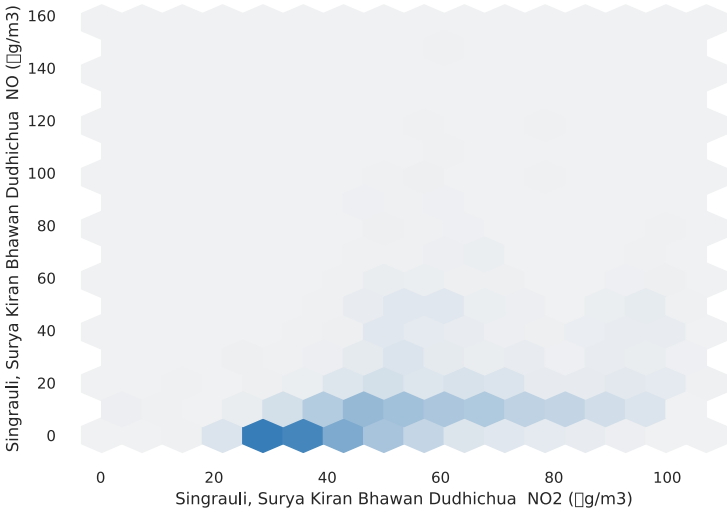
Singrauli, Surya Kiran Bha
#
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha

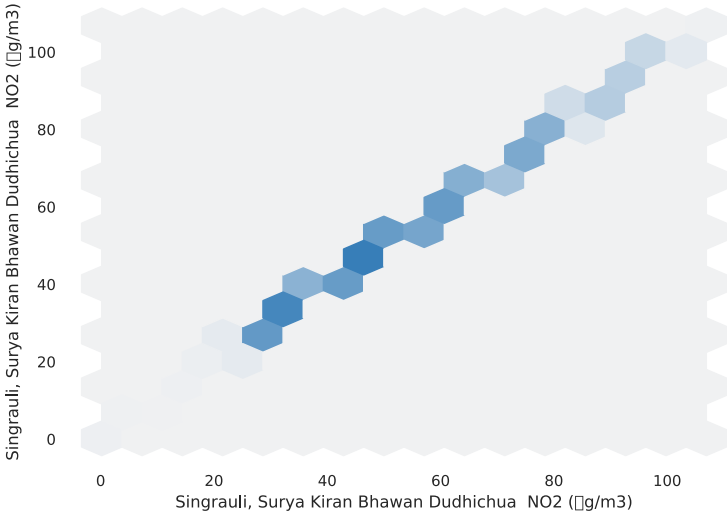


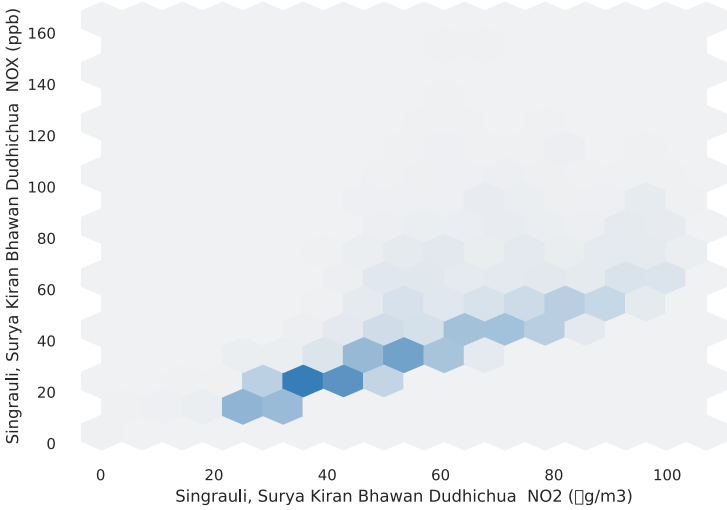

#
8000
6000
4000
2000
0
0
20
40
60
80
100
Singrauli, Surya Kiran Bhawan Dudhichua NO2 (□g/m3)

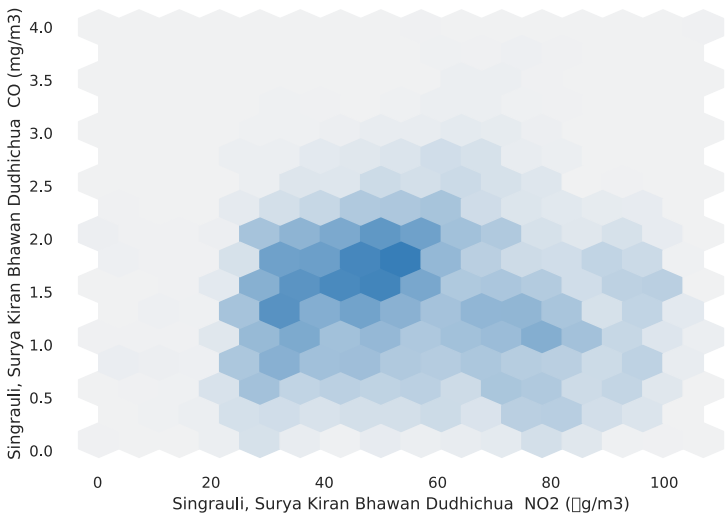

Singrauli, Surya Kiran Bhawan Dudhichua PM10 (□g/m3)
800
600
400
200
0
0
20
40
60
80
100
Singrauli, Surya Kiran Bhawan Dudhichua NO2 (□g/m3)

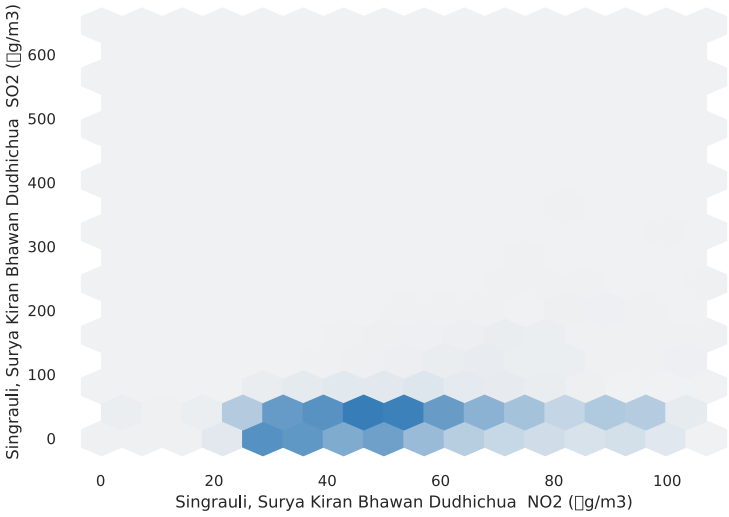

Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (□g/m3)
400
300
200
100
0
0
20
40
60
80
100
Singrauli, Surya Kiran Bhawan Dudhichua NO2 (□g/m3)

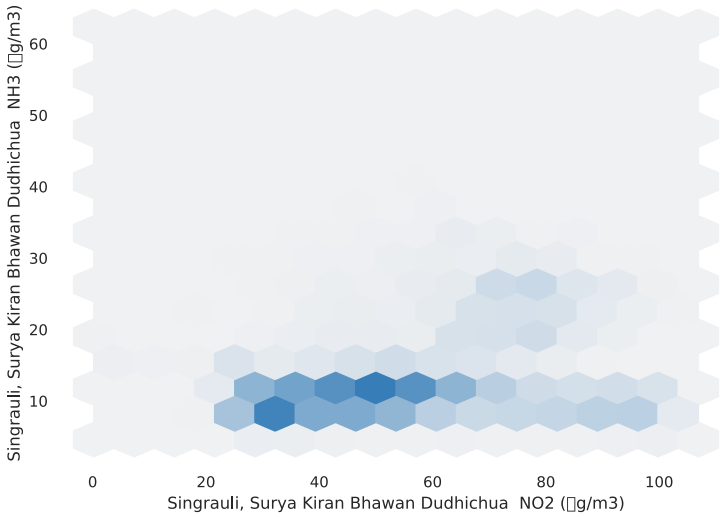

Singrauli, Surya Kiran Bhawan Dudhichua NO (□g/m3)
160
140
120
100
80
60
40
20
0
0
20
40
60
80
100
Singrauli, Surya Kiran Bhawan Dudhichua NO2 (□g/m3)

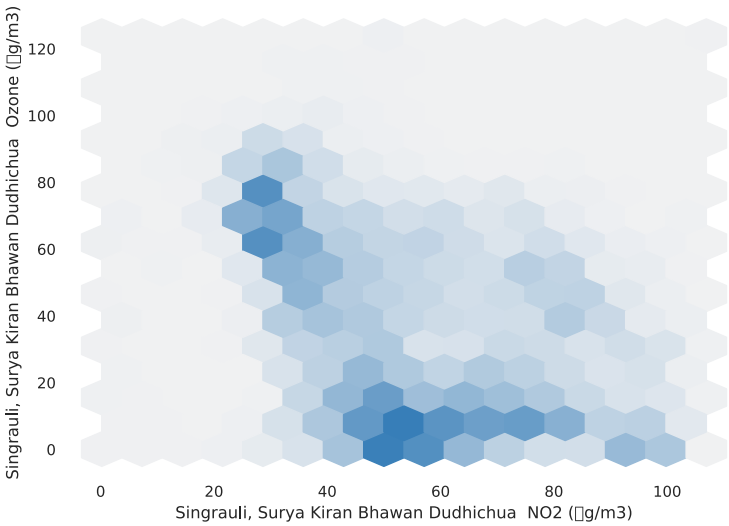

Singrauli, Surya Kiran Bhawan Dudhichua NO2 (□g/m3)
100
80
60
40
20
0
0
20
40
60
80
100
Singrauli, Surya Kiran Bhawan Dudhichua NO2 (□g/m3)


Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb)
160
140
120
100
80
60
40
20
0
0
20
40
60
80
100
Singrauli, Surya Kiran Bhawan Dudhichua NO2 (□g/m3)

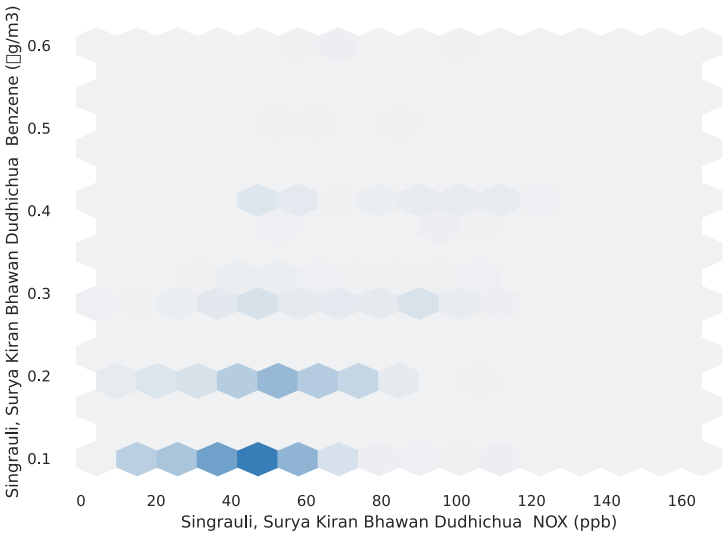

Singrauli, Surya Kiran Bhawan Dudhichua CO (mg/m3)
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
20
40
60
80
100
Singrauli, Surya Kiran Bhawan Dudhichua NO2 (g/m3)

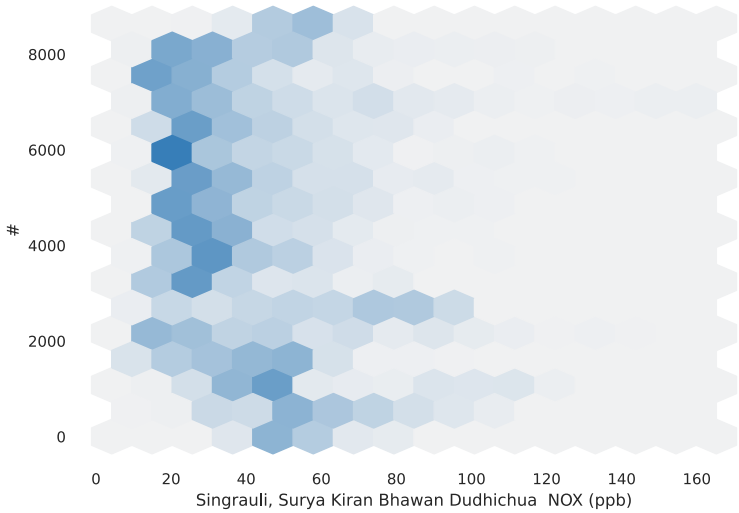

Singrauli, Surya Kiran Bhawan Dudhichua SO2 (g/m3)
600
500
400
300
200
100
0
0
20
40
60
80
100
Singrauli, Surya Kiran Bhawan Dudhichua NO2 (g/m3)

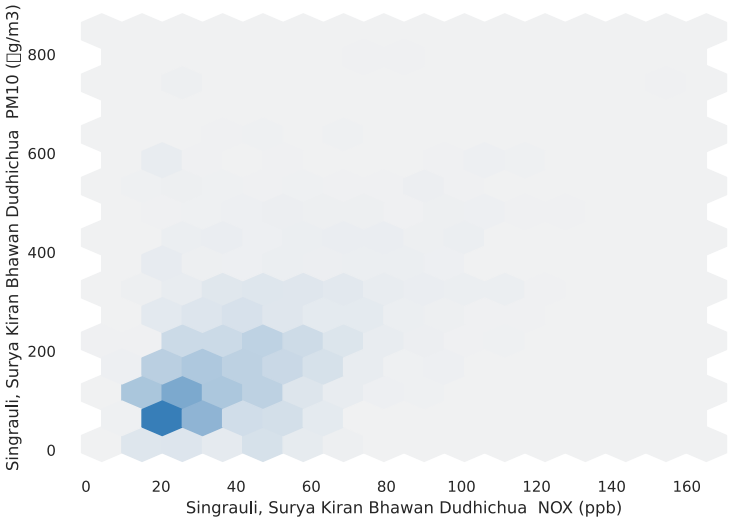

Singrauli, Surya Kiran Bhawan Dudhichua NH3 (g/m3)
60
50
40
30
20
10
0
20
40
60
80
100
Singrauli, Surya Kiran Bhawan Dudhichua NO2 (g/m3)

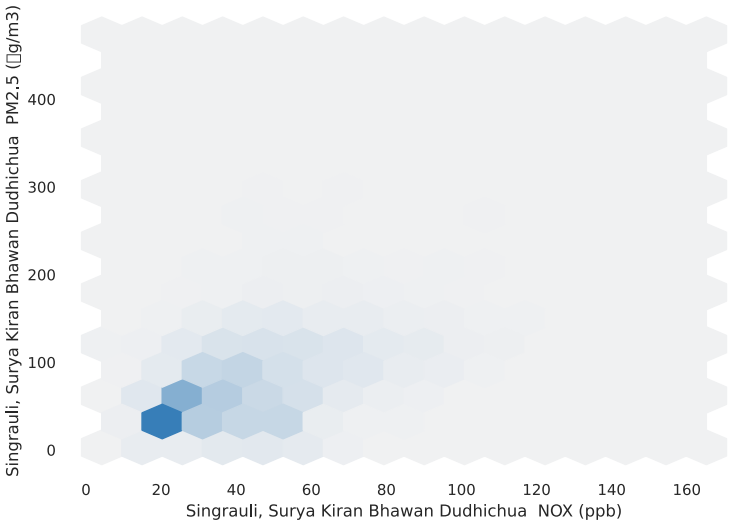


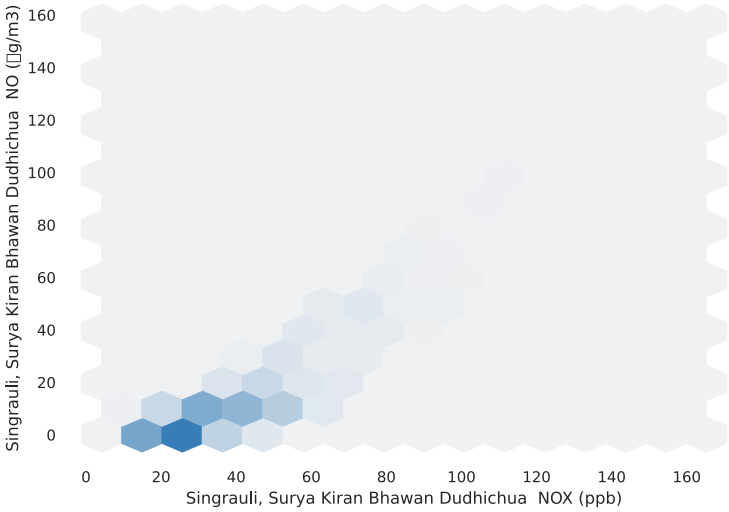
Singrauli, Surya Kiran Bha
#
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha

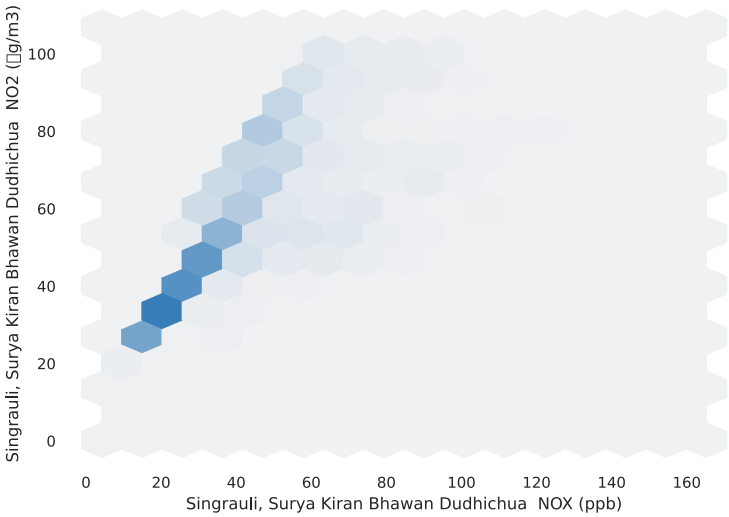


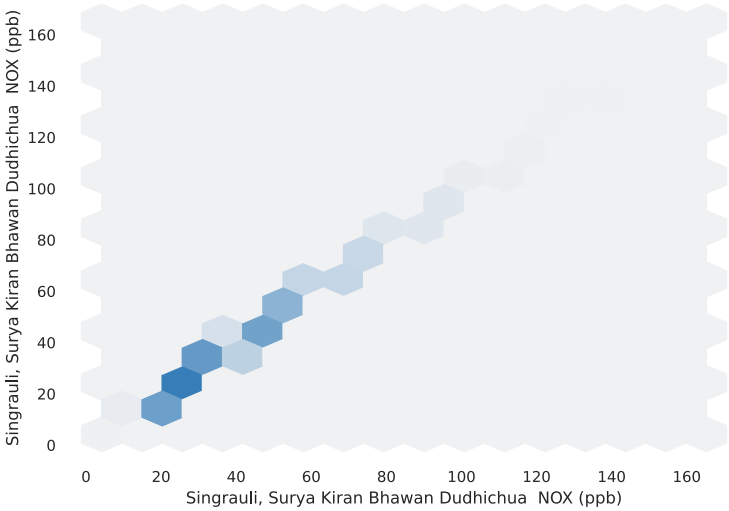


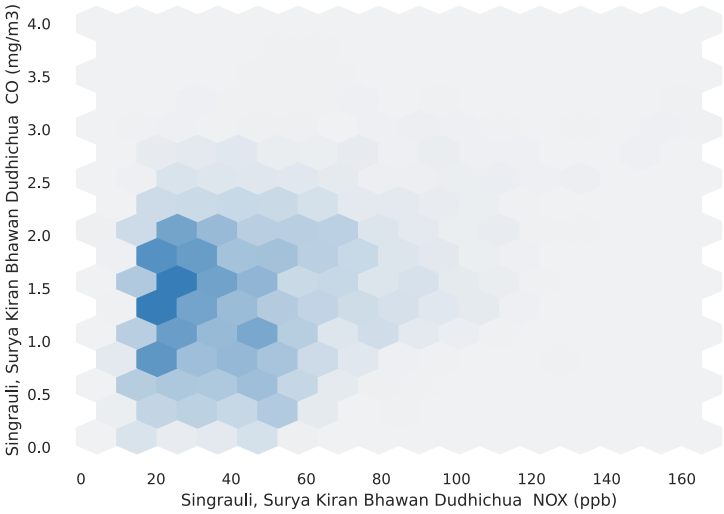

Singrauli, Surya Kiran Bhawan Dudhichua PM10 (g/m3)
0
200
400
600
800
0 20 40 60 80 100 120 140 160
Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb)

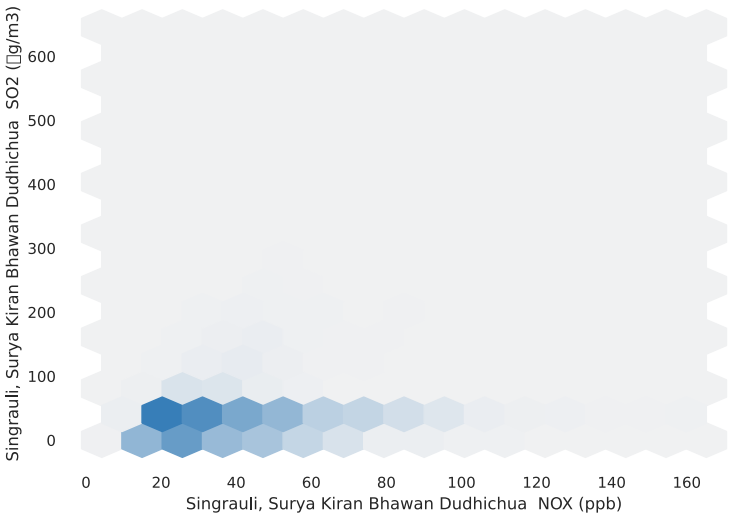

Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (g/m3)
0
100
200
300
400
0 20 40 60 80 100 120 140 160
Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb)

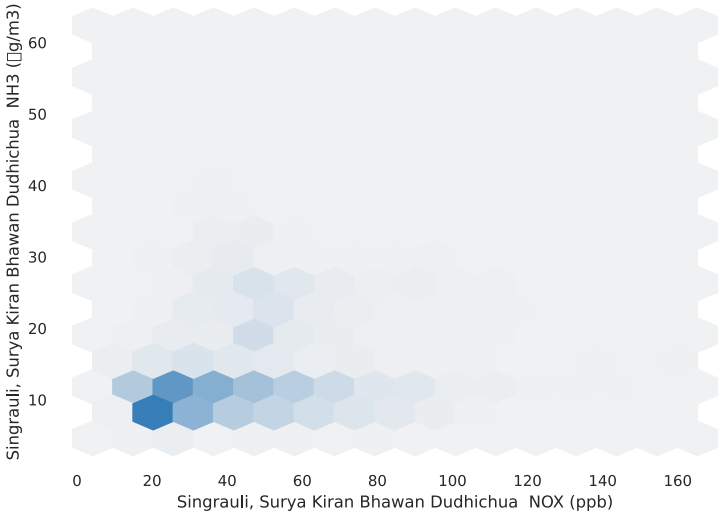

Singrauli, Surya Kiran Bhawan Dudhichua NO (g/m3)
0
20
40
60
80
100
120
140
160
0 20 40 60 80 100 120 140 160
Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb)

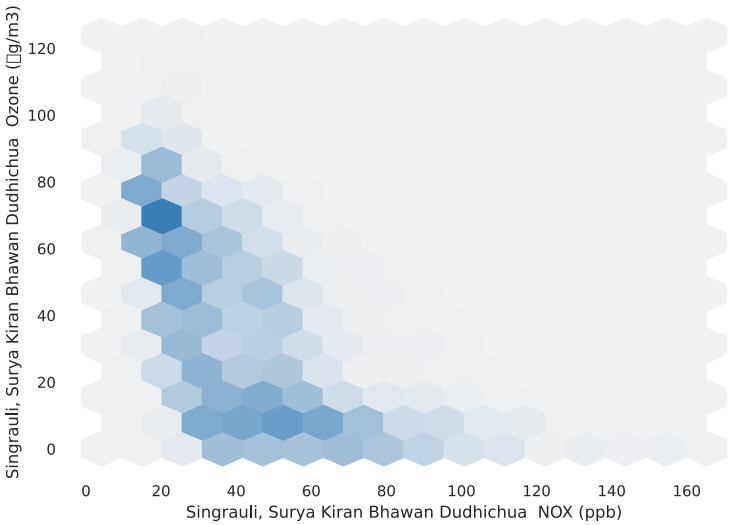

Singrauli, Surya Kiran Bhawan Dudhichua NO2 (µg/m3)
0
20
40
60
80
100
0
20
40
60
80
100
120
140
160
Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb)


Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb)
0
20
40
60
80
100
120
140
160
0
20
40
60
80
100
120
140
160
Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb)


Singrauli, Surya Kiran Bhawan Dudhichua CO (mg/m3)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
20
40
60
80
100
120
140
160
Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb)

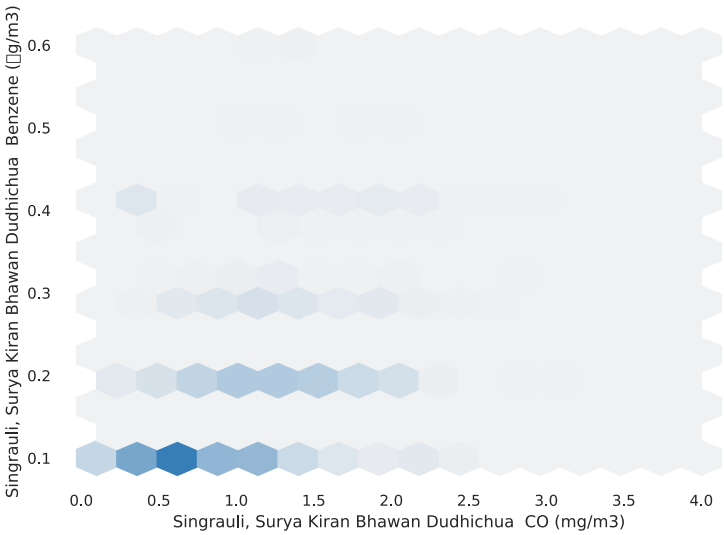


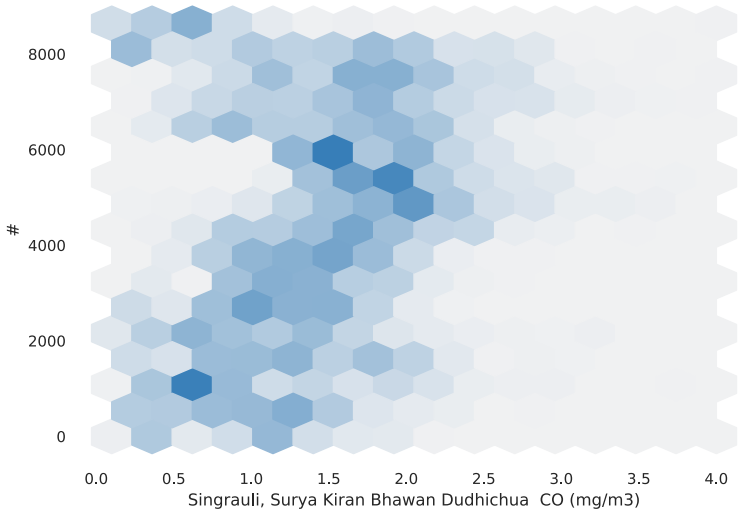


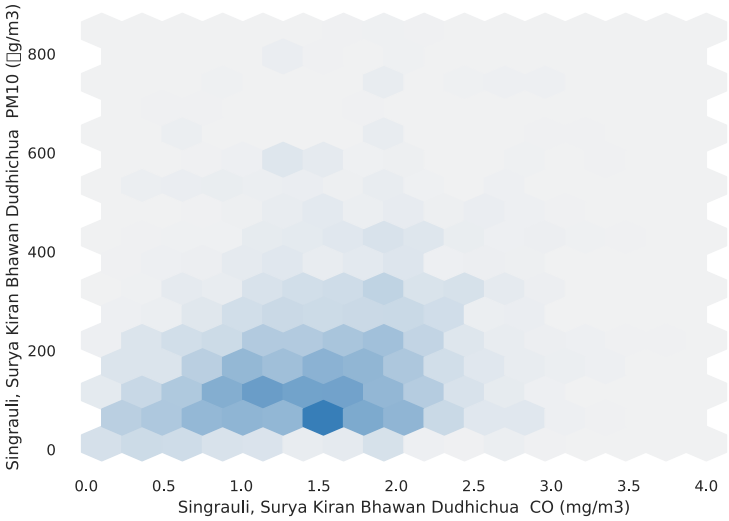


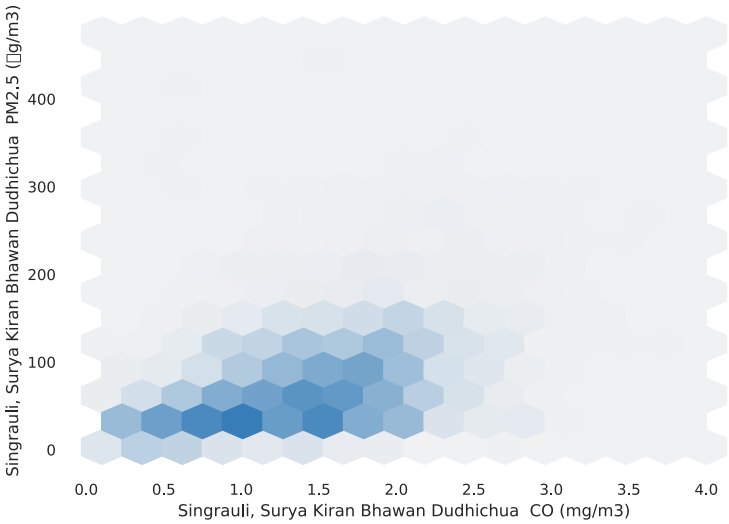
Singrauli, Surya Kiran Bha
#
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha

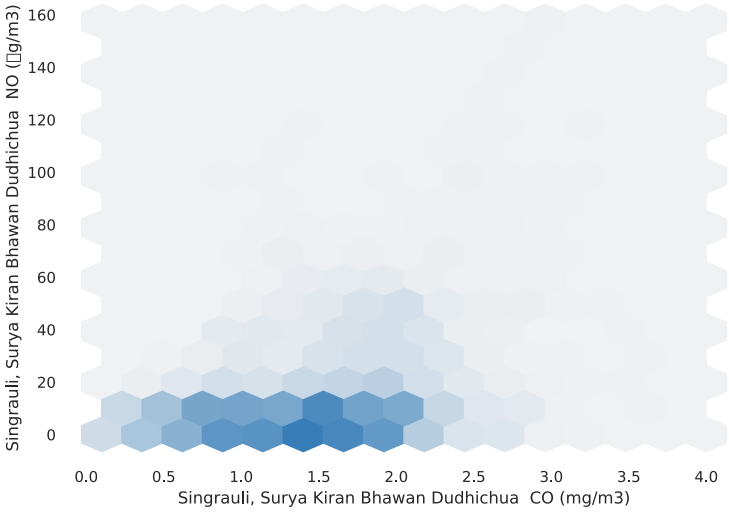
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha

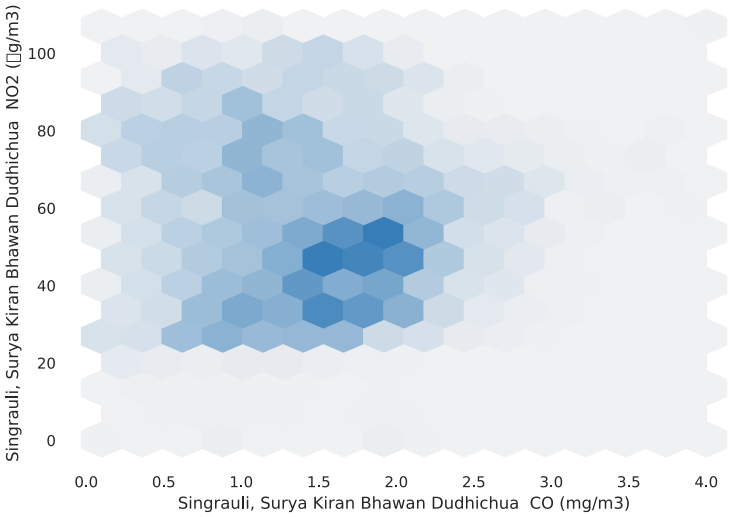


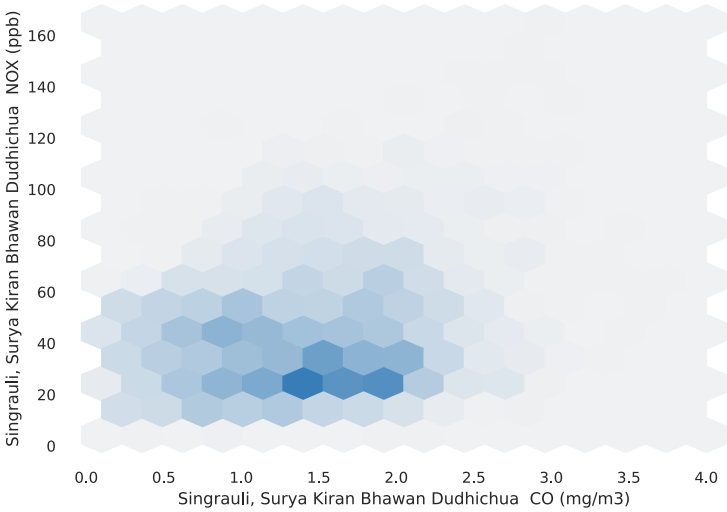


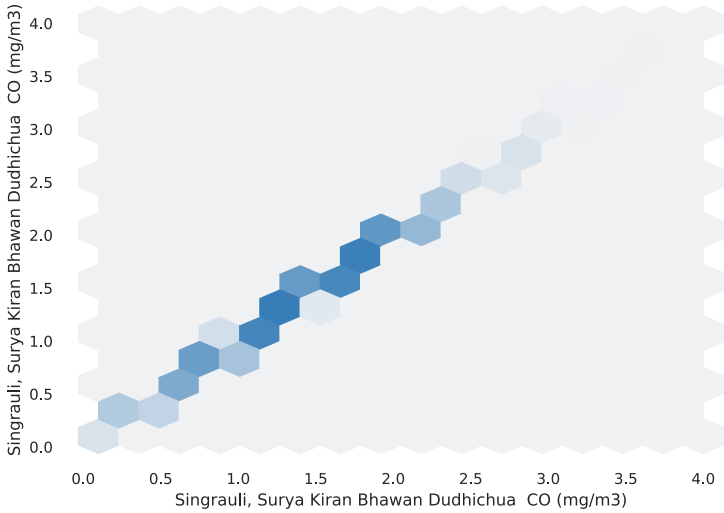


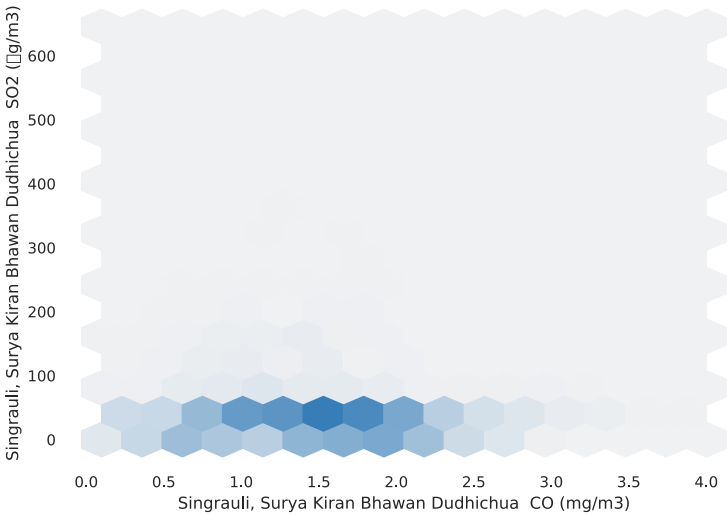

Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (g/m3)
0
100
200
300
400
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Singrauli, Surya Kiran Bhawan Dudhichua CO (mg/m3)

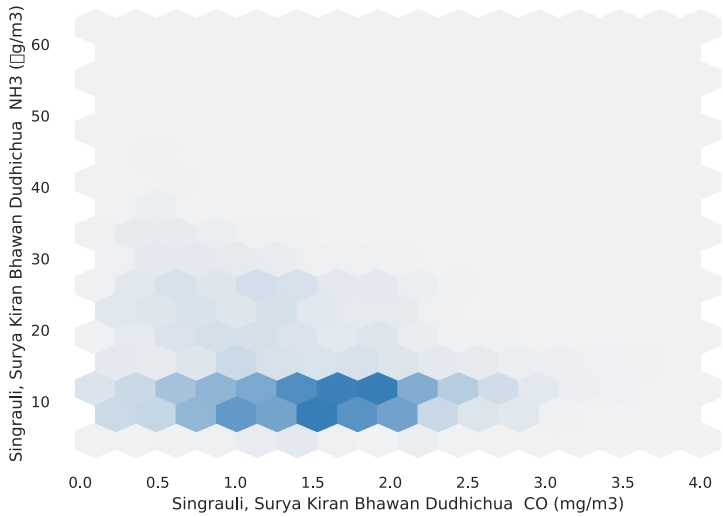

Singrauli, Surya Kiran Bhawan Dudhichua NO (g/m3)
0
20
40
60
80
100
120
140
160
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Singrauli, Surya Kiran Bhawan Dudhichua CO (mg/m3)

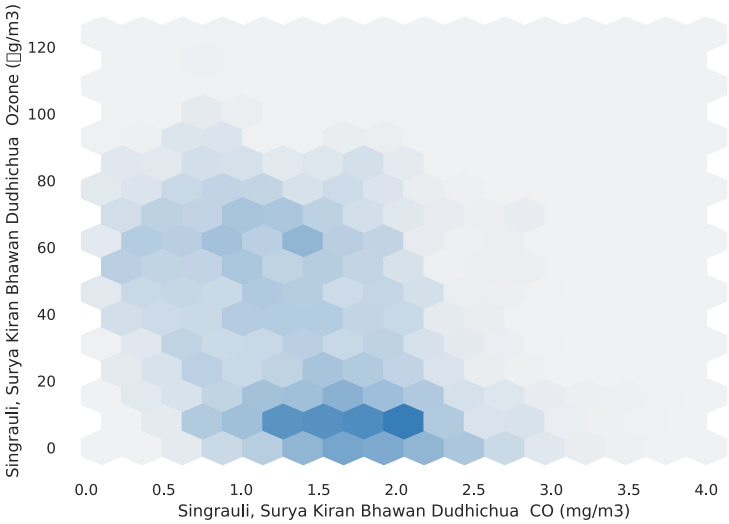

Singrauli, Surya Kiran Bhawan Dudhichua NO2 (g/m3)
0
20
40
60
80
100
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Singrauli, Surya Kiran Bhawan Dudhichua CO (mg/m3)


Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb)
0
20
40
60
80
100
120
140
160
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Singrauli, Surya Kiran Bhawan Dudhichua CO (mg/m3)

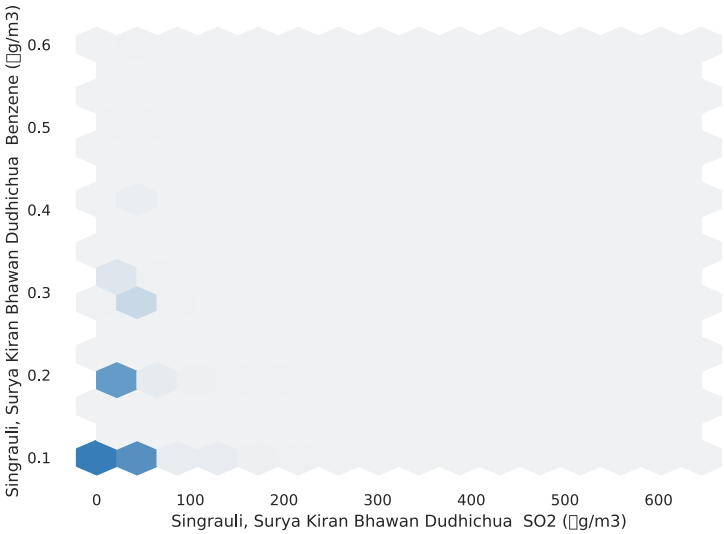

Singrauli, Surya Kiran Bhawan Dudhichua CO (mg/m3)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Singrauli, Surya Kiran Bhawan Dudhichua CO (mg/m3)

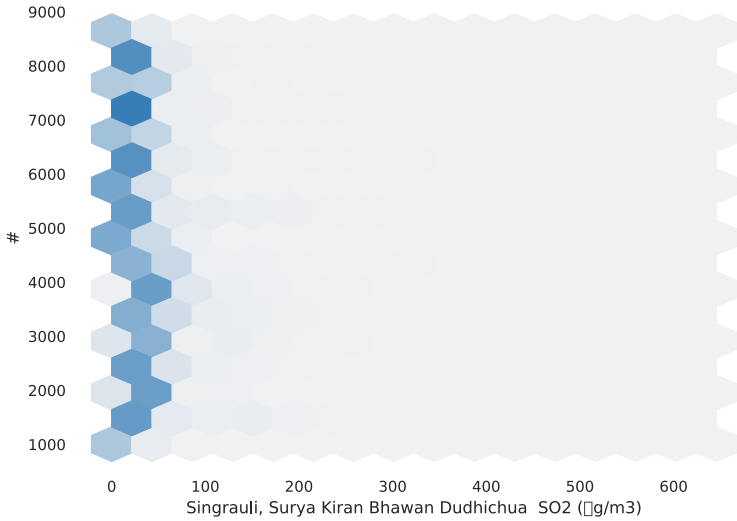

Singrauli, Surya Kiran Bhawan Dudhichua SO2 (□g/m3)
0
100
200
300
400
500
600
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Singrauli, Surya Kiran Bhawan Dudhichua CO (mg/m3)

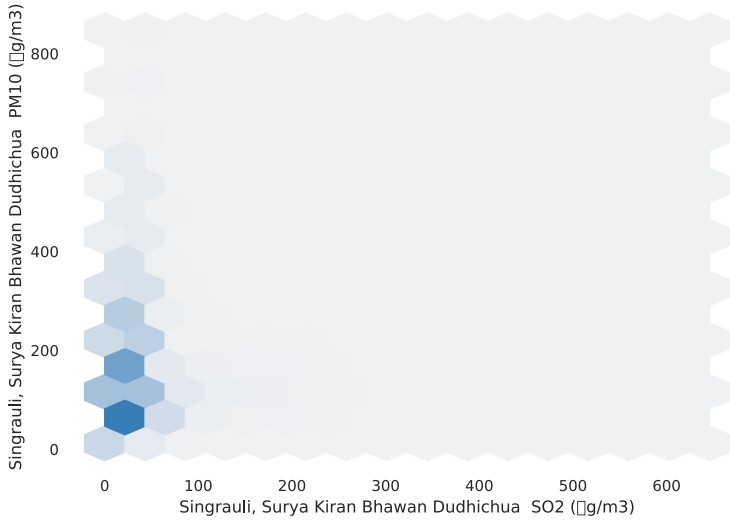


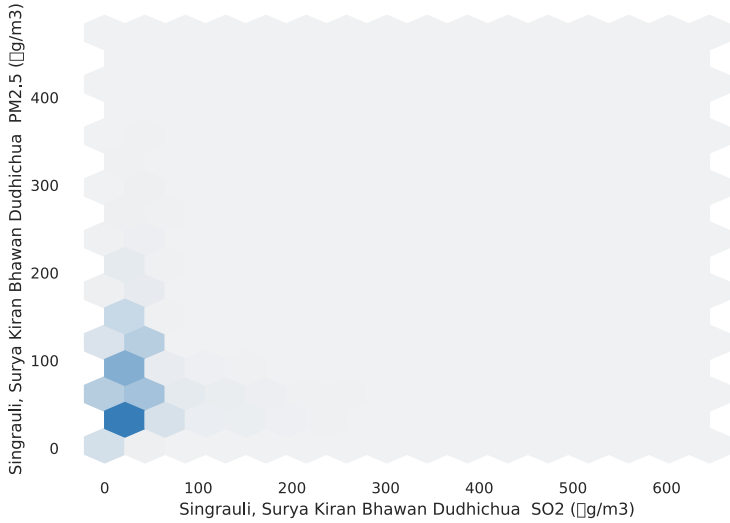


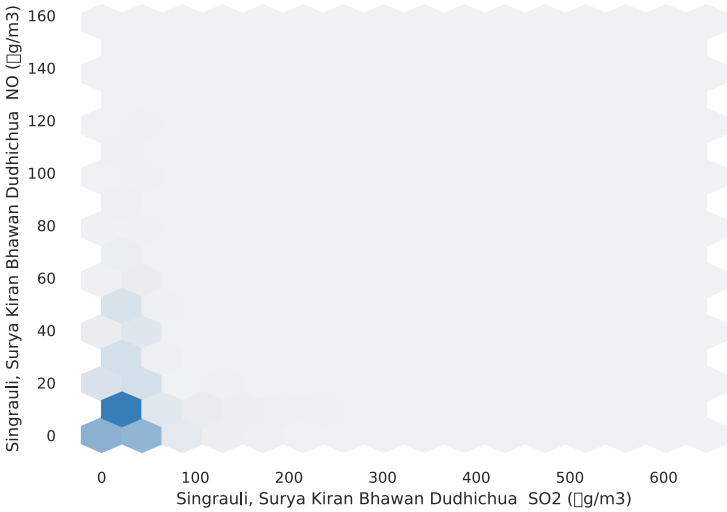
Singrauli, Surya Kiran Bha
#
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha

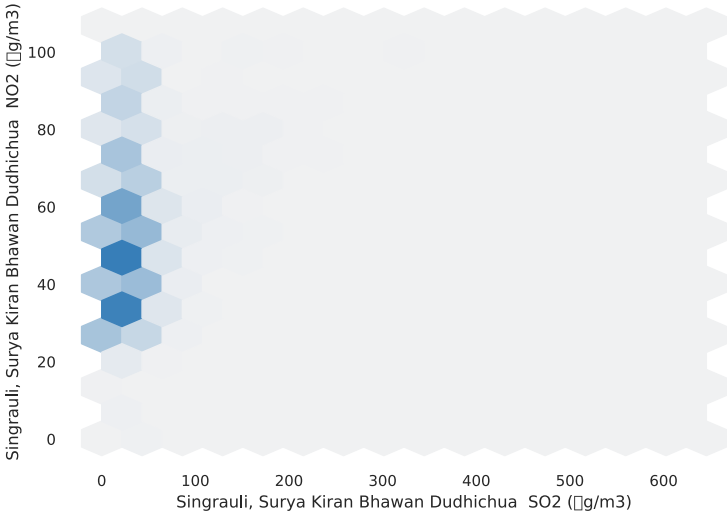


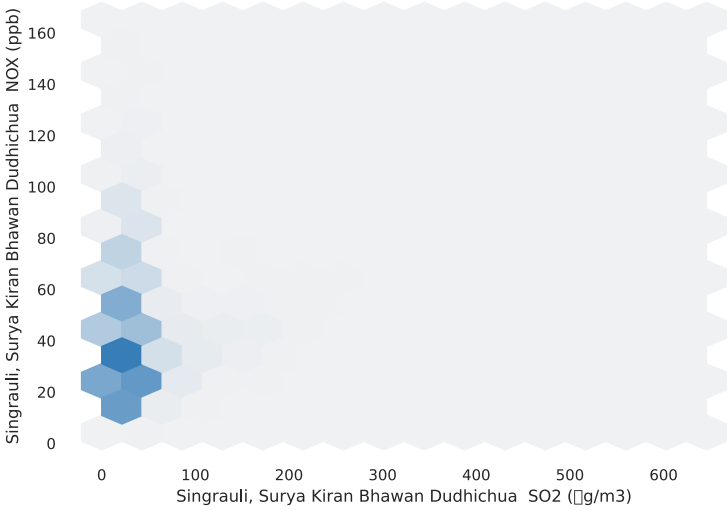

9000
8000
7000
6000
5000
4000
3000
2000
1000
#
0
100
200
300
400
500
600
Singrauli, Surya Kiran Bhawan Dudhichua SO2 (g/m3)

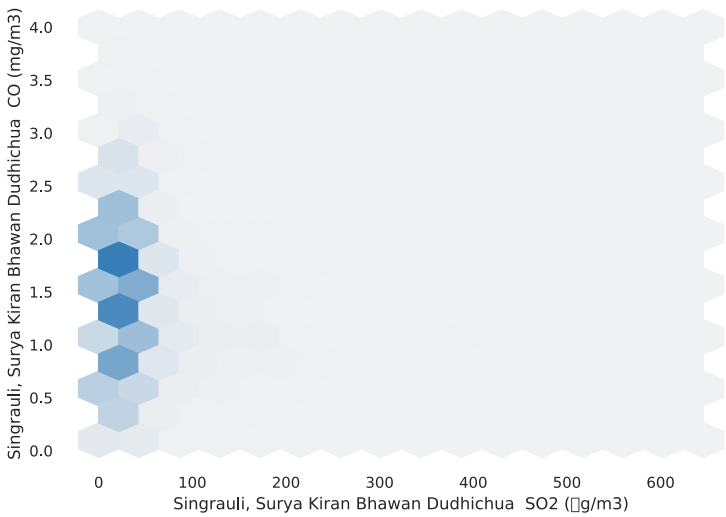

800
600
400
200
0
Singrauli, Surya Kiran Bhawan Dudhichua PM10 (g/m3)
0
100
200
300
400
500
600
Singrauli, Surya Kiran Bhawan Dudhichua SO2 (g/m3)

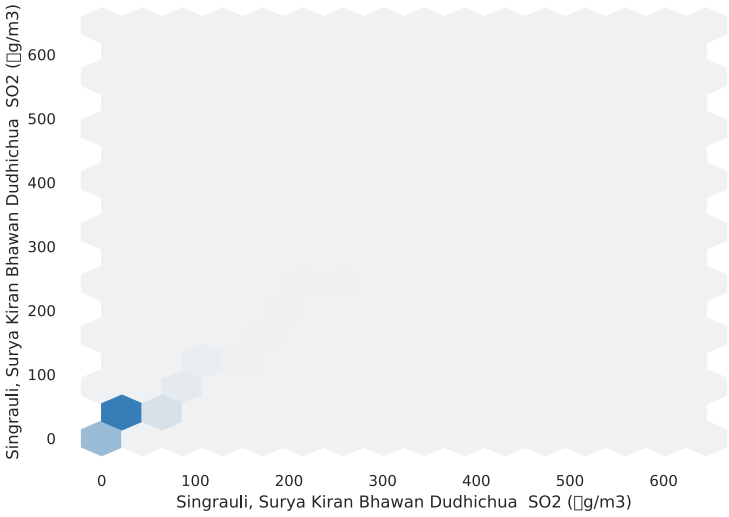

400
300
200
100
0
Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (g/m3)
0
100
200
300
400
500
600
Singrauli, Surya Kiran Bhawan Dudhichua SO2 (g/m3)

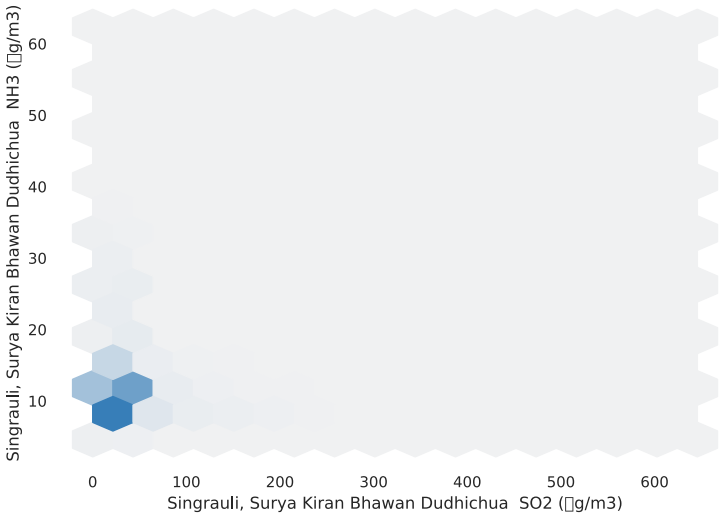

Singrauli, Surya Kiran Bhawan Dudhichua NO (µg/m3)
160
140
120
100
80
60
40
20
0
0
100
200
300
400
500
600
Singrauli, Surya Kiran Bhawan Dudhichua SO2 (µg/m3)

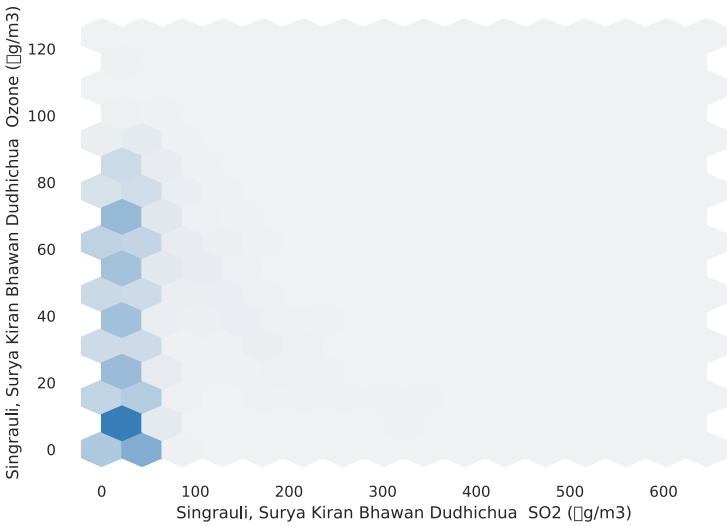

Singrauli, Surya Kiran Bhawan Dudhichua NO2 (µg/m3)
100
80
60
40
20
0
0
100
200
300
400
500
600
Singrauli, Surya Kiran Bhawan Dudhichua SO2 (µg/m3)


Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb)
160
140
120
100
80
60
40
20
0
0
100
200
300
400
500
600
Singrauli, Surya Kiran Bhawan Dudhichua SO2 (µg/m3)

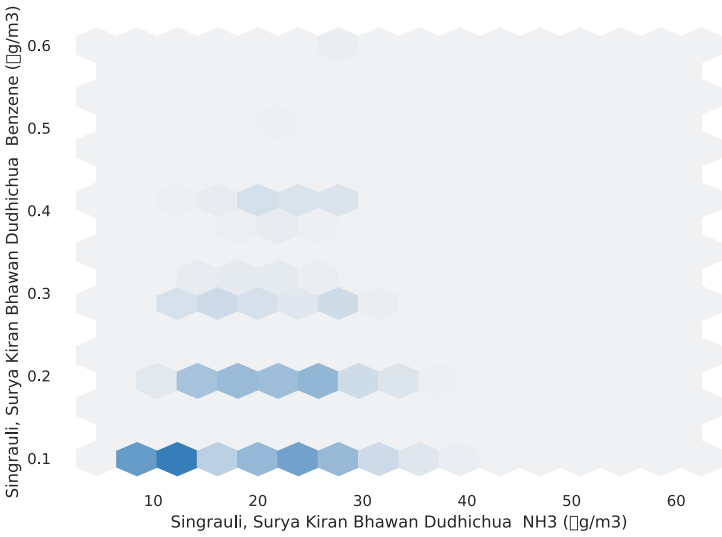

Singrauli, Surya Kiran Bhawan Dudhichua CO (mg/m3)
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
100
200
300
400
500
600
Singrauli, Surya Kiran Bhawan Dudhichua SO2 (g/m3)

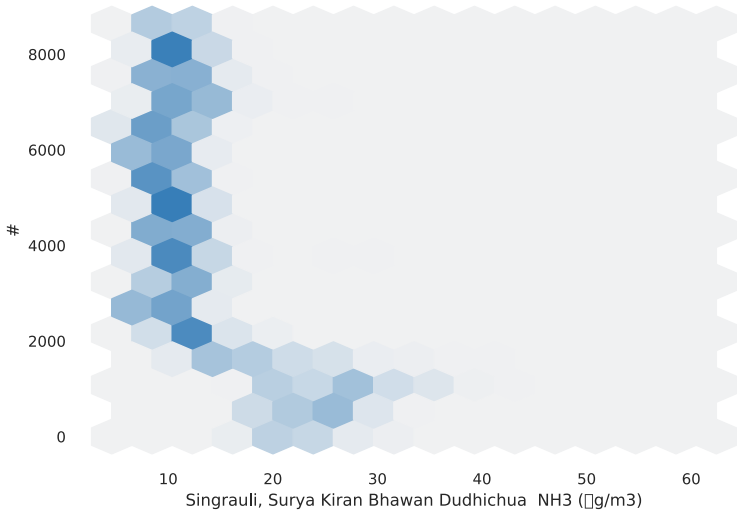

Singrauli, Surya Kiran Bhawan Dudhichua SO2 (g/m3)
600
500
400
300
200
100
0
0
100
200
300
400
500
600
Singrauli, Surya Kiran Bhawan Dudhichua SO2 (g/m3)

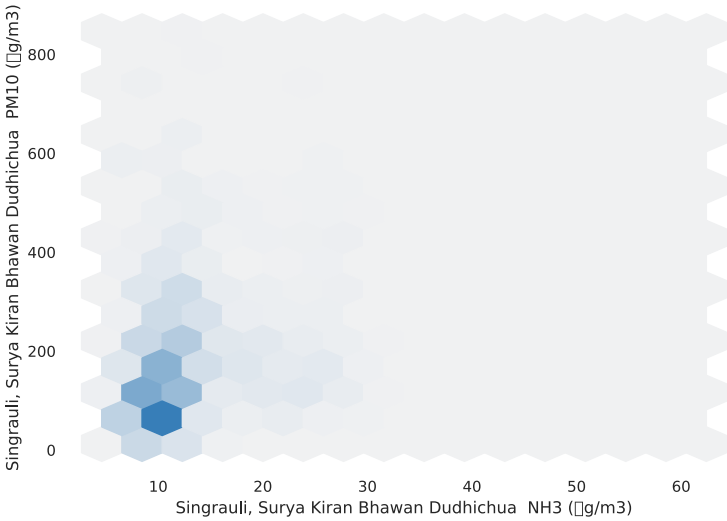

Singrauli, Surya Kiran Bhawan Dudhichua NH3 (g/m3)
60
50
40
30
20
10
0
100
200
300
400
500
600
Singrauli, Surya Kiran Bhawan Dudhichua SO2 (g/m3)

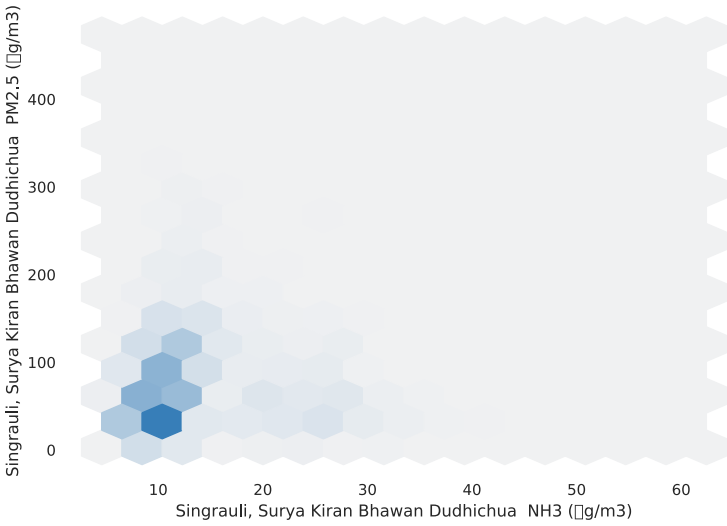


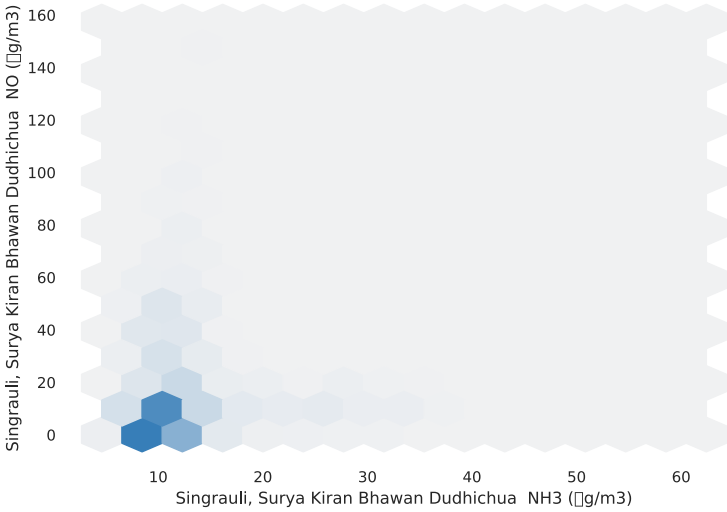
Singrauli, Surya Kiran Bha
#
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha

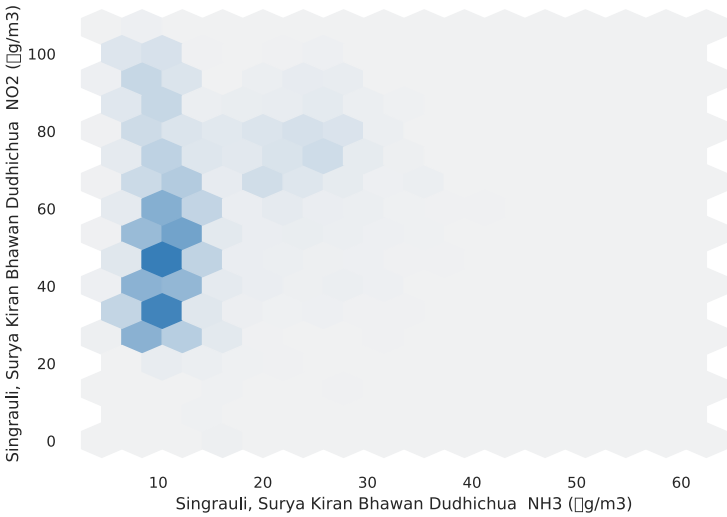


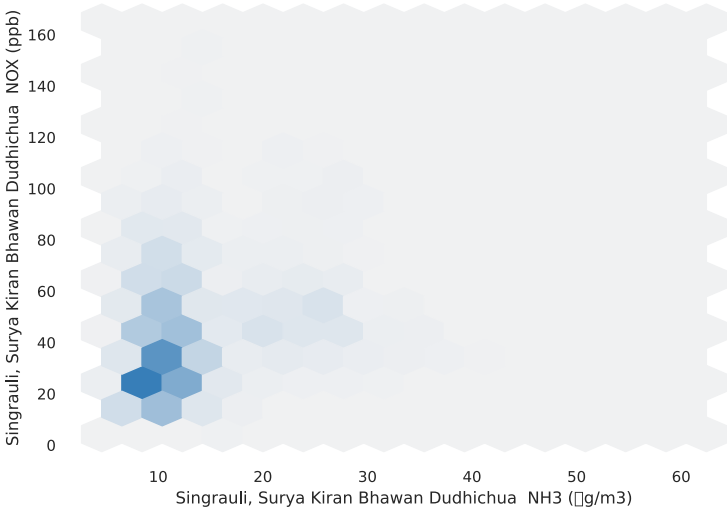


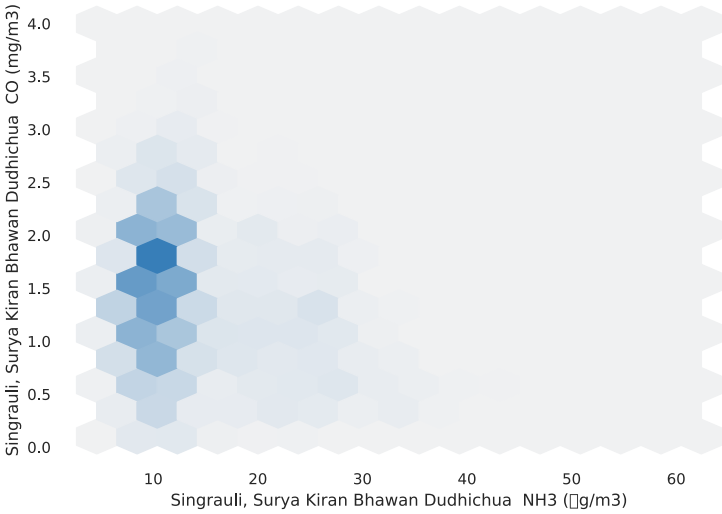

Singrauli, Surya Kiran Bhawan Dudhichua PM10 (g/m3)
800
600
400
200
0
10
20
30
40
50
60
Singrauli, Surya Kiran Bhawan Dudhichua NH3 (g/m3)

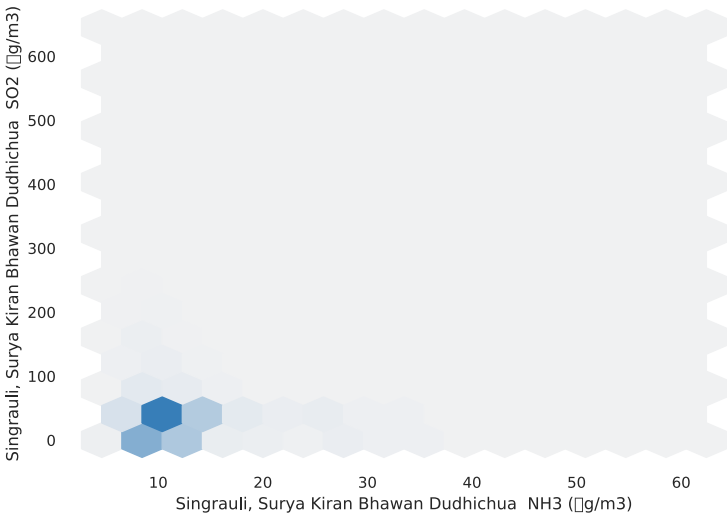

Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (g/m3)
400
300
200
100
0
10
20
30
40
50
60
Singrauli, Surya Kiran Bhawan Dudhichua NH3 (g/m3)

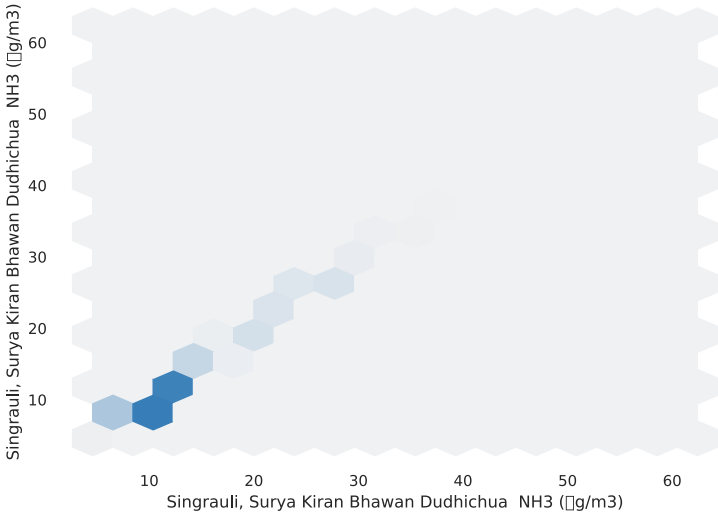

Singrauli, Surya Kiran Bhawan Dudhichua NO (g/m3)
160
140
120
100
80
60
40
20
0
10
20
30
40
50
60
Singrauli, Surya Kiran Bhawan Dudhichua NH3 (g/m3)

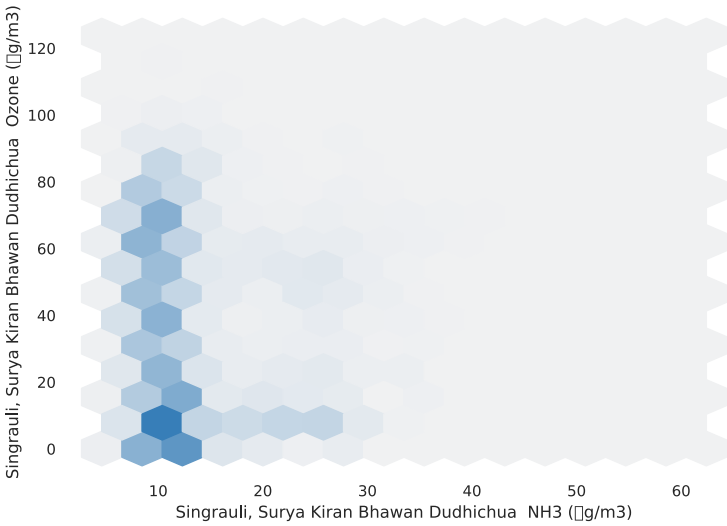

Singrauli, Surya Kiran Bhawan Dudhichua NO2 (g/m3)
0
20
40
60
80
100
10
20
30
40
50
60
Singrauli, Surya Kiran Bhawan Dudhichua NH3 (g/m3)


Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb)
0
20
40
60
80
100
120
140
160
10
20
30
40
50
60
Singrauli, Surya Kiran Bhawan Dudhichua NH3 (g/m3)


Singrauli, Surya Kiran Bhawan Dudhichua CO (mg/m3)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
10
20
30
40
50
60
Singrauli, Surya Kiran Bhawan Dudhichua NH3 (g/m3)

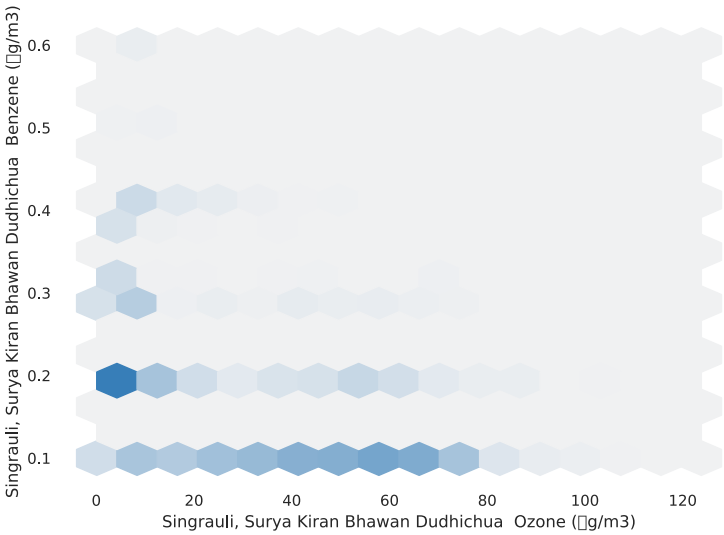


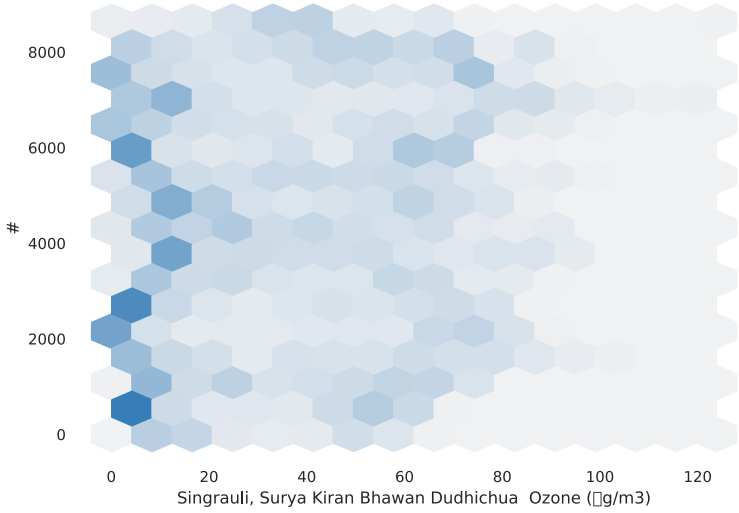


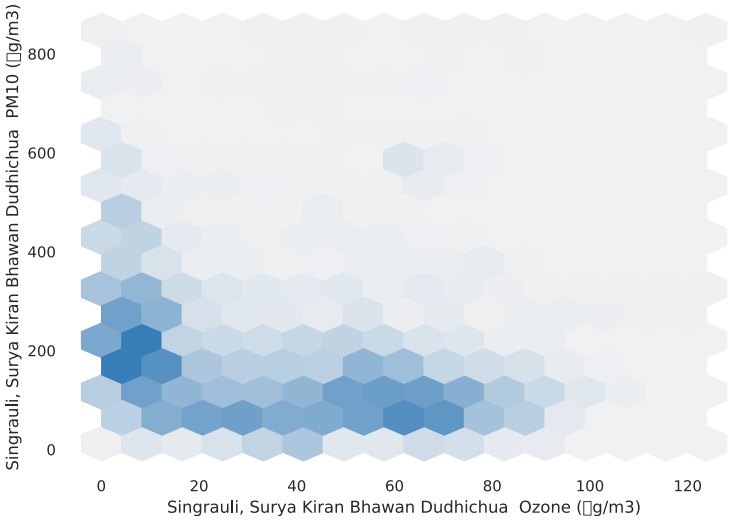


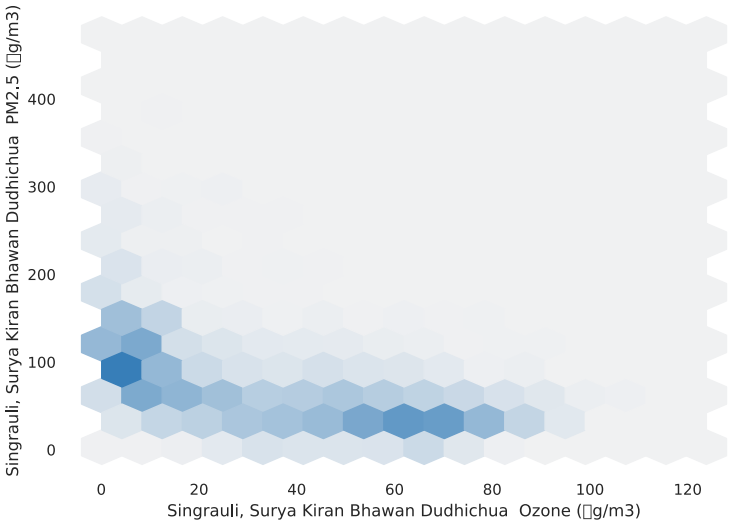
Singrauli, Surya Kiran Bha
#
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha

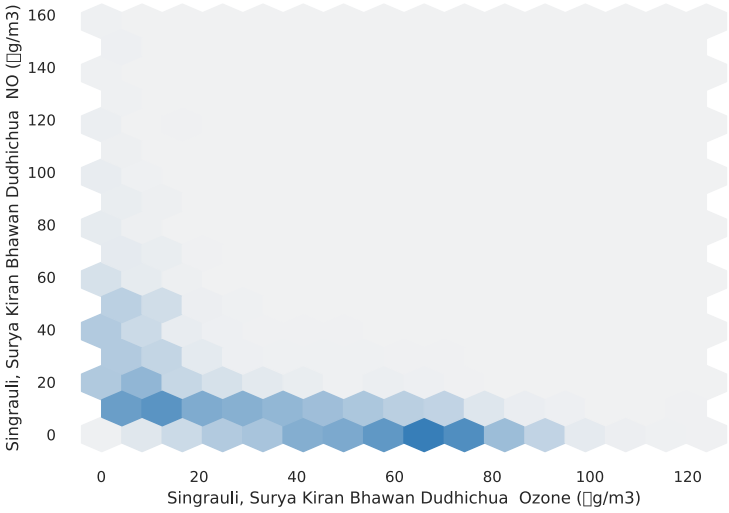
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha

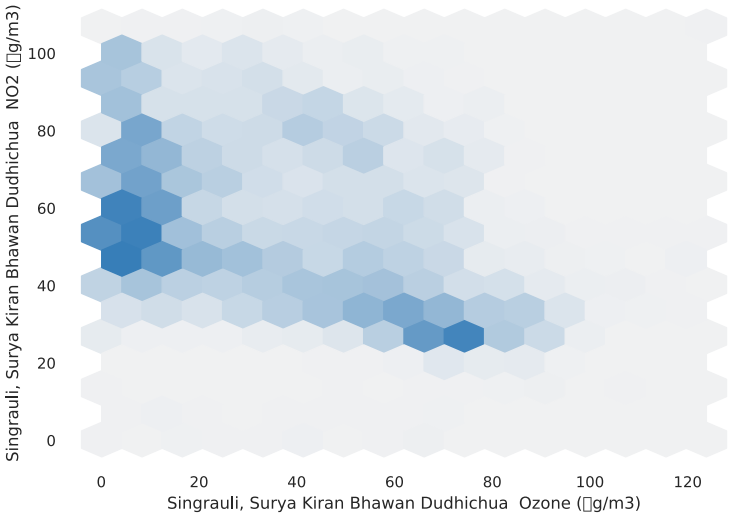


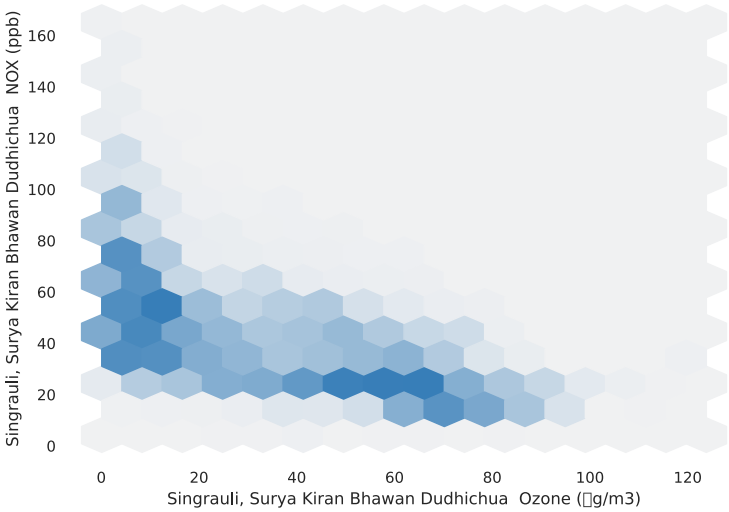


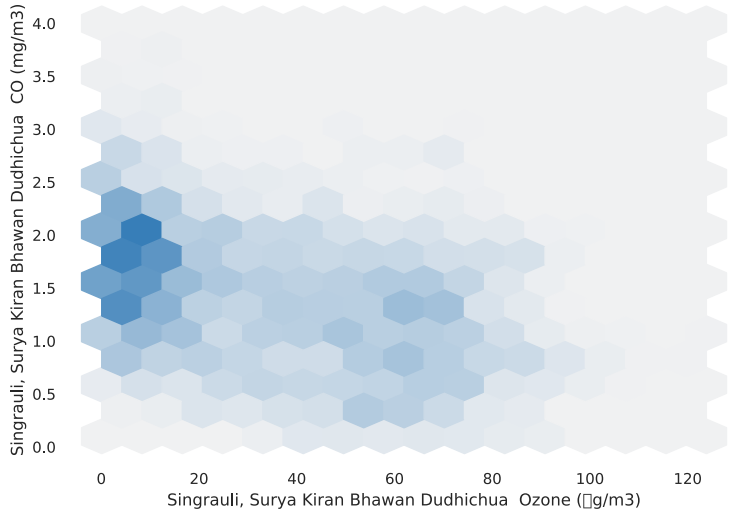


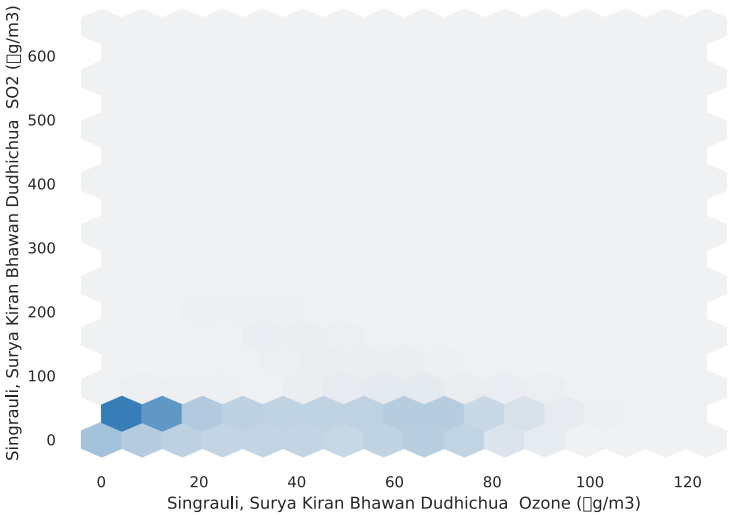

Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (g/m3)
400
300
200
100
0
0
20
40
60
80
100
120
Singrauli, Surya Kiran Bhawan Dudhichua Ozone (g/m3)

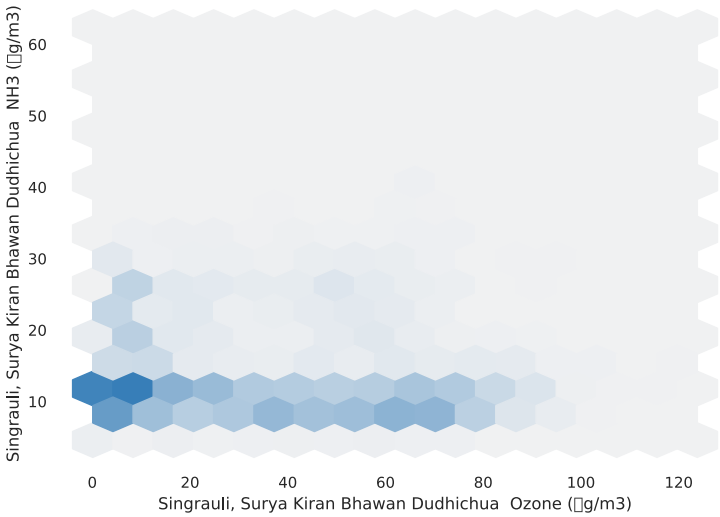

Singrauli, Surya Kiran Bhawan Dudhichua NO (g/m3)
160
140
120
100
80
60
40
20
0
0
20
40
60
80
100
120
Singrauli, Surya Kiran Bhawan Dudhichua Ozone (g/m3)

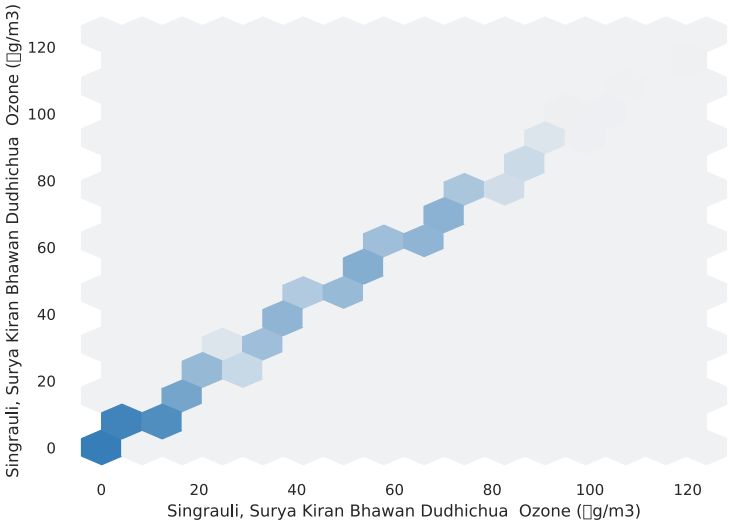

Singrauli, Surya Kiran Bhawan Dudhichua NO2 (g/m3)
100
80
60
40
20
0
0
20
40
60
80
100
120
Singrauli, Surya Kiran Bhawan Dudhichua Ozone (g/m3)


Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb)
0
20
40
60
80
100
120
140
160
0
20
40
60
80
100
120
Singrauli, Surya Kiran Bhawan Dudhichua Ozone (□g/m3)

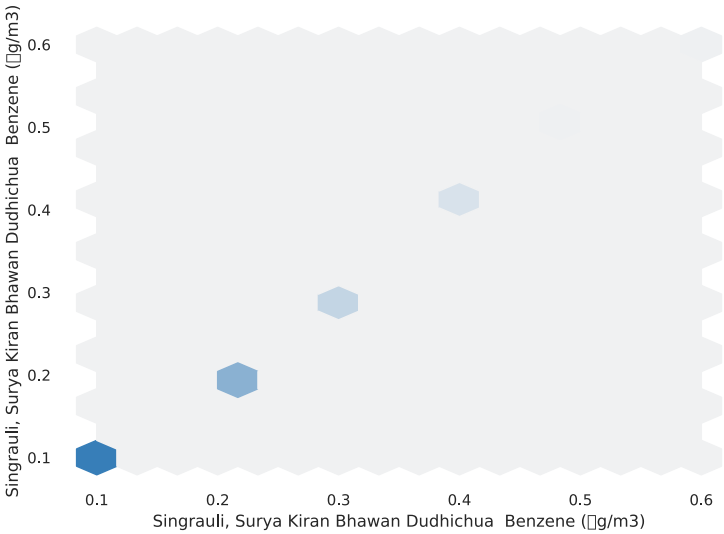

Singrauli, Surya Kiran Bhawan Dudhichua CO (mg/m3)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
20
40
60
80
100
120
Singrauli, Surya Kiran Bhawan Dudhichua Ozone (□g/m3)

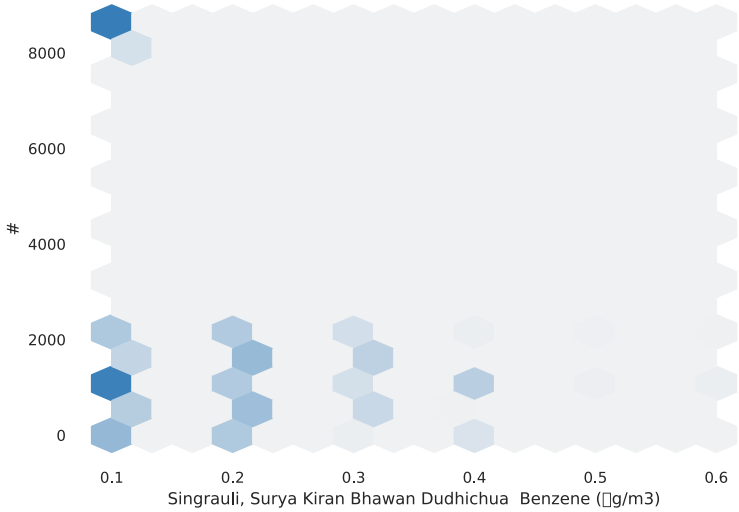

Singrauli, Surya Kiran Bhawan Dudhichua SO2 (□g/m3)
0
100
200
300
400
500
600
0
20
40
60
80
100
120
Singrauli, Surya Kiran Bhawan Dudhichua Ozone (□g/m3)

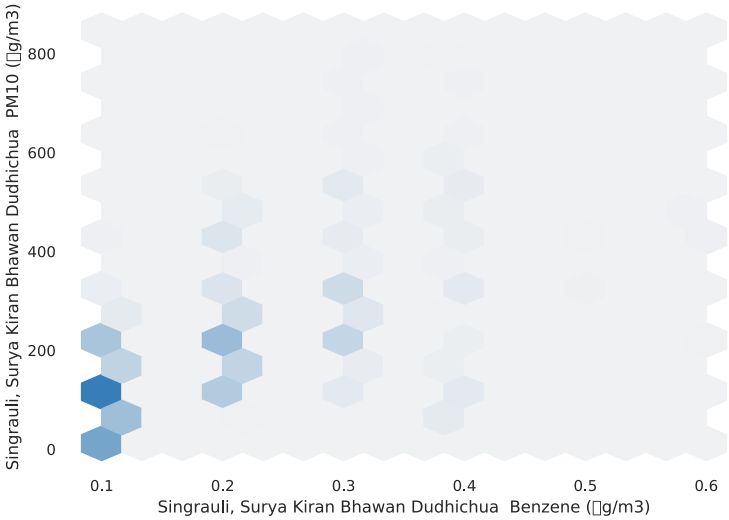


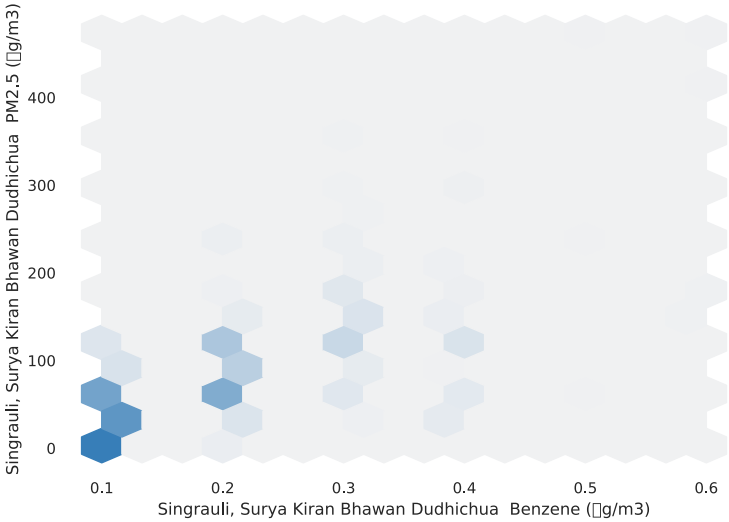


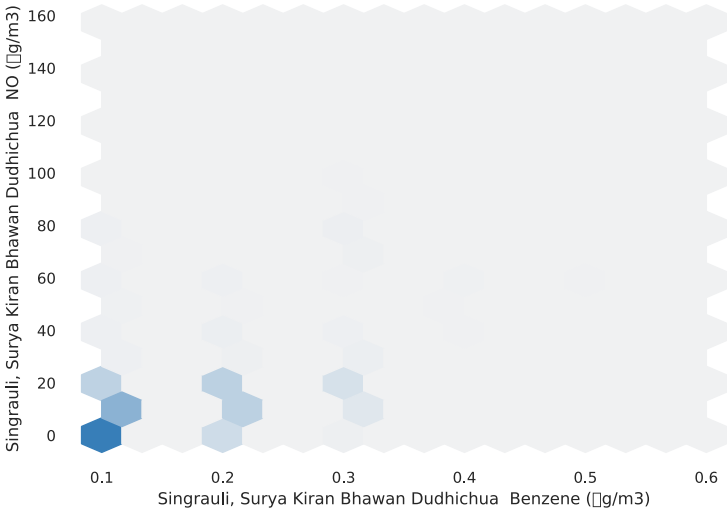
Singrauli, Surya Kiran Bha
#
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha
Singrauli, Surya Kiran Bha

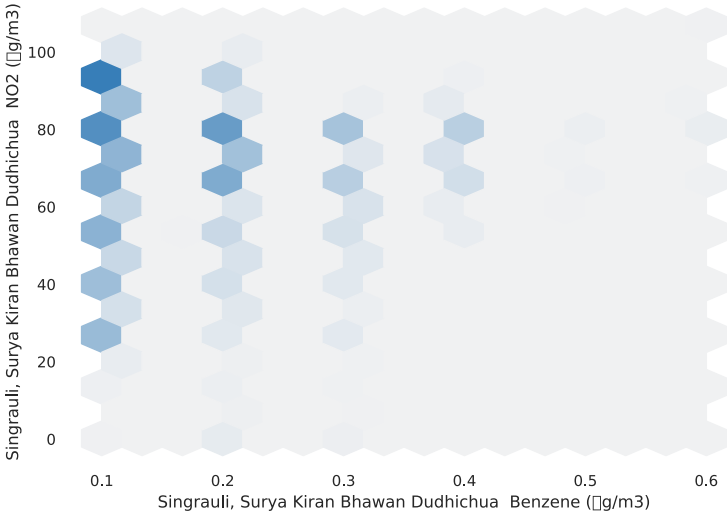


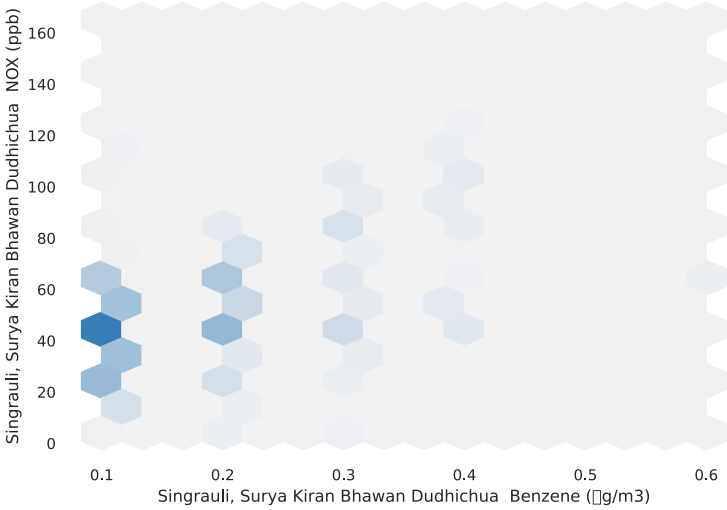

#
8000
6000
4000
2000
0
0.1
0.2
0.3
0.4
0.5
0.6
Singrauli, Surya Kiran Bhawan Dudhichua Benzene (□g/m3)

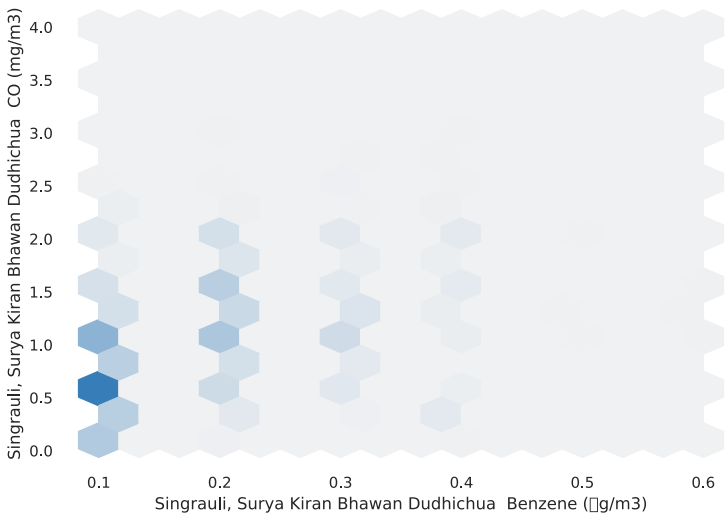

Singrauli, Surya Kiran Bhawan Dudhichua PM10 (□g/m3)
800
600
400
200
0
0.1
0.2
0.3
0.4
0.5
0.6
Singrauli, Surya Kiran Bhawan Dudhichua Benzene (□g/m3)

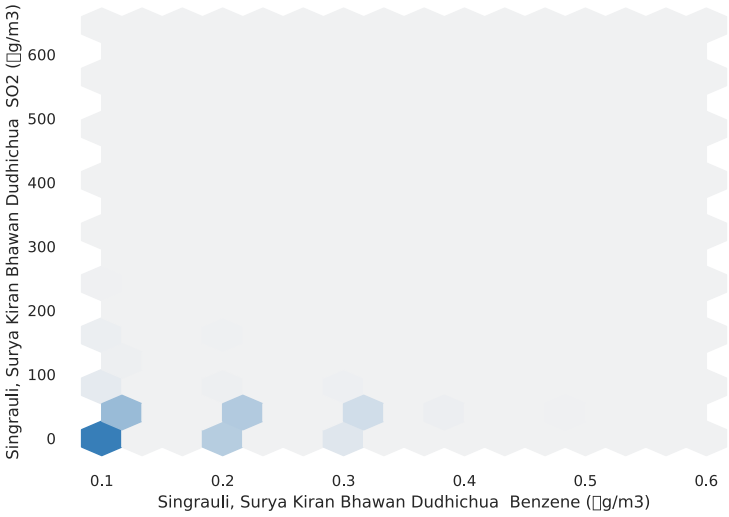

Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (□g/m3)
400
300
200
100
0
0.1
0.2
0.3
0.4
0.5
0.6
Singrauli, Surya Kiran Bhawan Dudhichua Benzene (□g/m3)

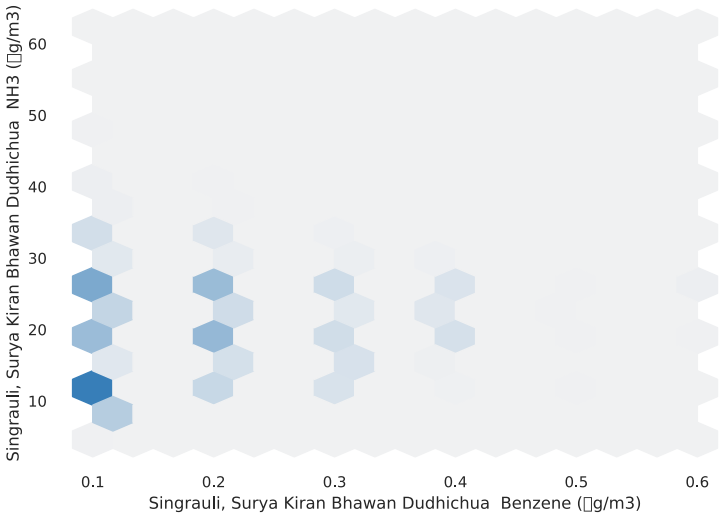

Singrauli, Surya Kiran Bhawan Dudhichua NO (□g/m3)
160
140
120
100
80
60
40
20
0
0.1
0.2
0.3
0.4
0.5
0.6
Singrauli, Surya Kiran Bhawan Dudhichua Benzene (□g/m3)

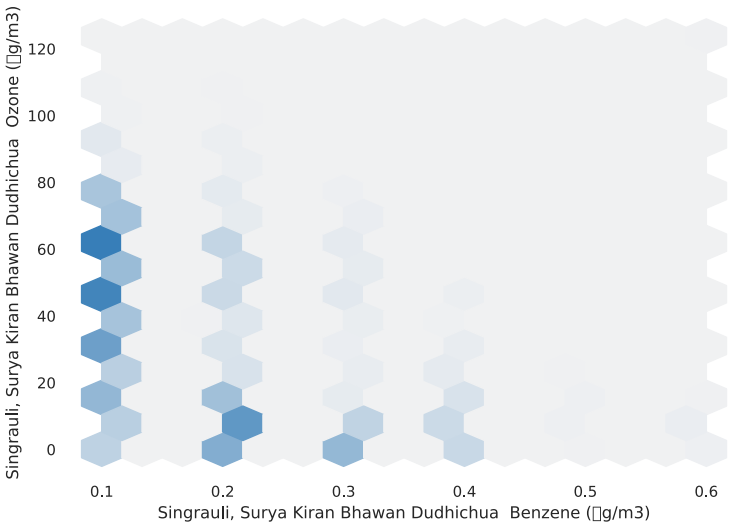

Singrauli, Surya Kiran Bhawan Dudhichua NO2 (□g/m3)
100
80
60
40
20
0
0.1
0.2
0.3
0.4
0.5
0.6
Singrauli, Surya Kiran Bhawan Dudhichua Benzene (□g/m3)


Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb)
160
140
120
100
80
60
40
20
0
0.1
0.2
0.3
0.4
0.5
0.6
Singrauli, Surya Kiran Bhawan Dudhichua Benzene (□g/m3)

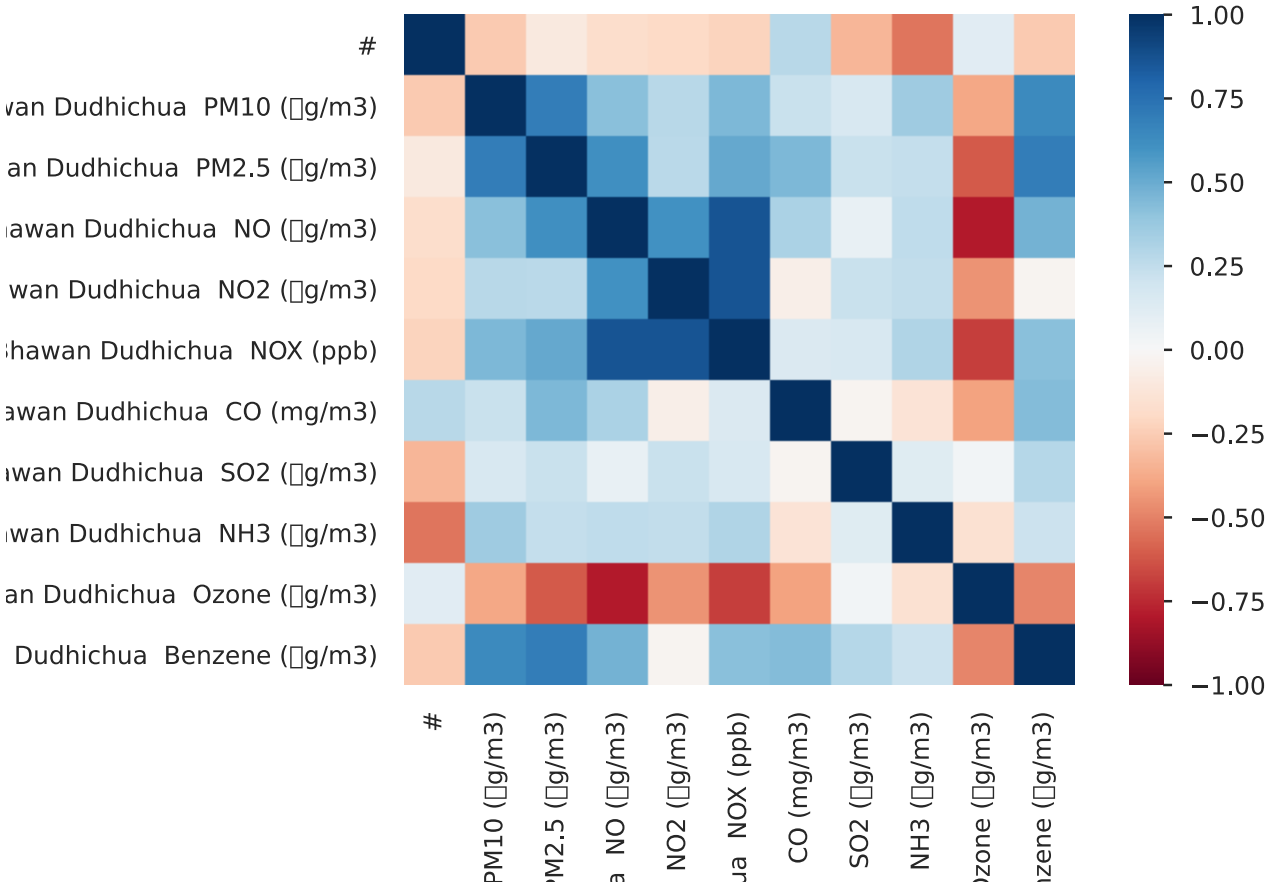

Singrauli, Surya Kiran Bhawan Dudhichua CO (mg/m3)
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Singrauli, Surya Kiran Bhawan Dudhichua Benzene (□g/m3)


Singrauli, Surya Kiran Bhawan Dudhichua SO2 (□g/m3)
600
500
400
300
200
100
0
0.1
0.2
0.3
0.4
0.5
0.6
Singrauli, Surya Kiran Bhawan Dudhichua Benzene (□g/m3)


Singrauli, Surya Kiran Bhawan Dudhichua NH3 (□g/m3)
60
50
40
30
20
10
0.1
0.2
0.3
0.4
0.5
0.6
Singrauli, Surya Kiran Bhawan Dudhichua Benzene (□g/m3)

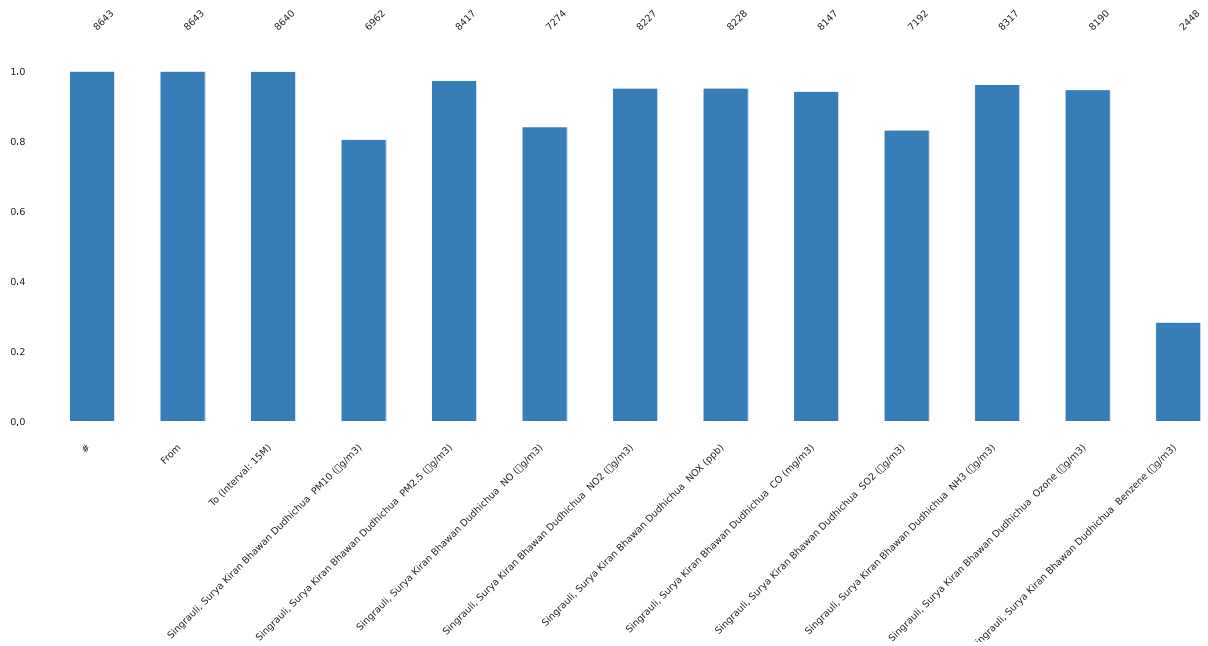


Correlations

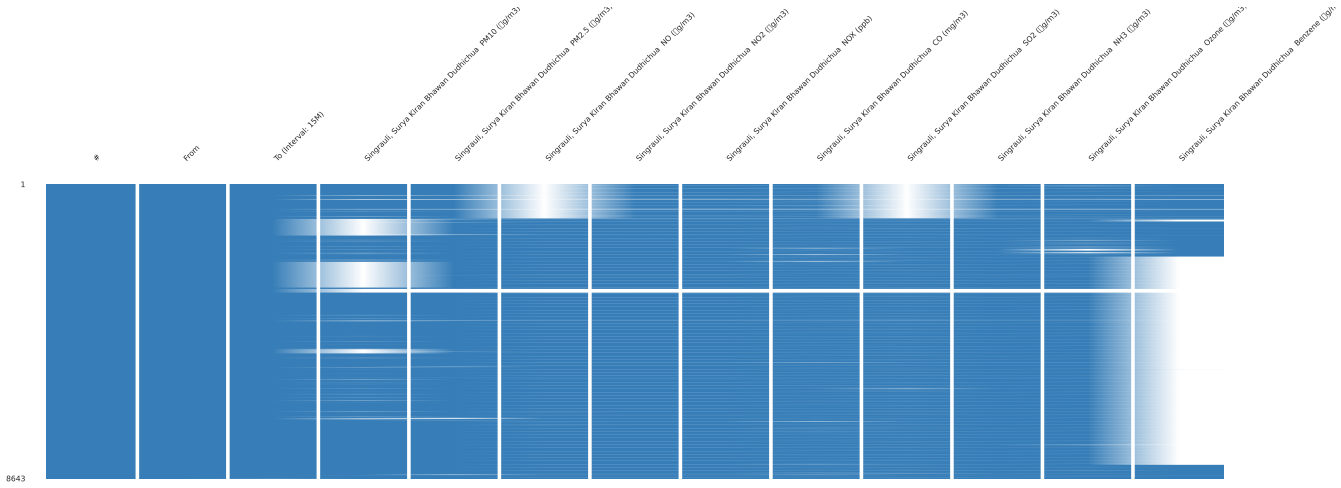


	#	Singrauli, Surya Kiran Bhawan Dudhichua PM10 (碌g/m3)	Singrauli, Surya Kiran Bha
#	1.000	-0.261	-0.100
Singrauli, Surya Kiran Bhawan Dudhichua PM10 (碌g/m3)	-0.261	1.000	0.695
Singrauli, Surya Kiran Bhawan Dudhichua PM2.5 (碌g/m3)	-0.100	0.695	1.000
Singrauli, Surya Kiran Bhawan Dudhichua NO (碌g/m3)	-0.177	0.421	0.615
Singrauli, Surya Kiran Bhawan Dudhichua NO2 (碌g/m3)	-0.202	0.279	0.271
Singrauli, Surya Kiran Bhawan Dudhichua NOX (ppb)	-0.221	0.451	0.514
Singrauli, Surya Kiran Bhawan Dudhichua CO (mg/m3)	0.278	0.226	0.447
Singrauli, Surya Kiran Bhawan Dudhichua SO2 (碌g/m3)	-0.330	0.157	0.223
Singrauli, Surya Kiran Bhawan Dudhichua NH3 (碌g/m3)	-0.532	0.352	0.237
Singrauli, Surya Kiran Bhawan Dudhichua Ozone (碌g/m3)	0.111	-0.388	-0.613

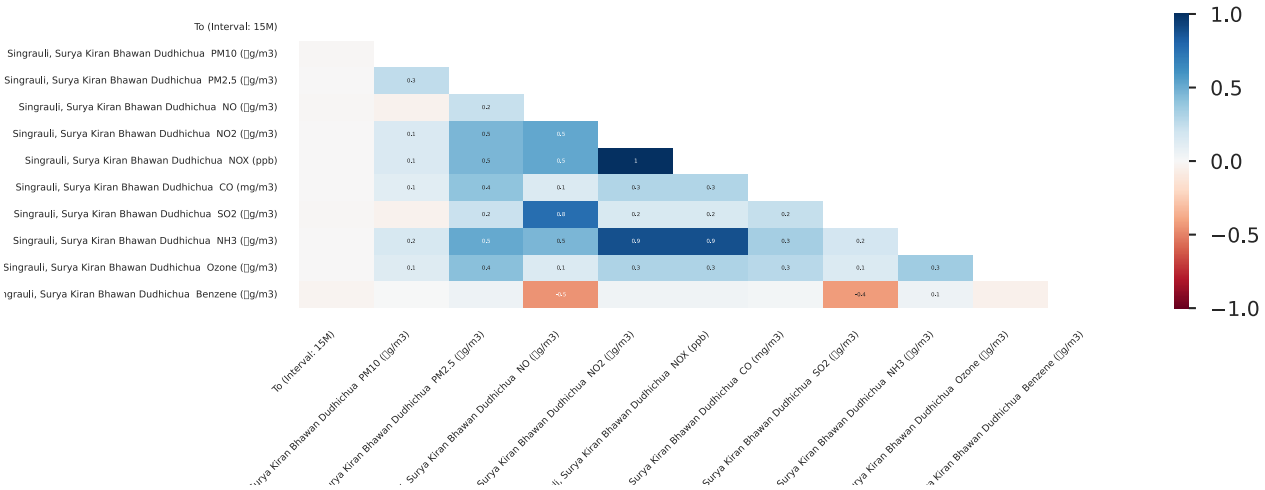
Missing values



A simple visualization of nullity by column.



Nullity matrix is a data-dense display which lets you quickly visually pick out patterns in data completion.



The correlation heatmap measures nullity correlation: how strongly the presence or absence of one variable affects the presence of another.

Sample

	#	From	To (Interval: 15M)	Singrauli, Surya Kiran Bhawan Dudhichua PM10 (碌g/m3)	Singrauli, Surya Kiran Bhawan Dudhichua PM2.5
0	1	2023-02-01 0:00:00	2023-02-01 0:15:00	95.0	35.0
1	2	2023-02-01 0:15:00	2023-02-01 0:30:00	95.0	35.0
2	3	2023-02-01 0:30:00	2023-02-01 0:45:00	95.0	35.0
3	4	2023-02-01 0:45:00	2023-02-01 1:00:00	122.0	34.0
4	5	2023-02-01 1:00:00	2023-02-01 1:15:00	122.0	34.0
5	6	2023-02-01 1:15:00	2023-02-01 1:30:00	122.0	34.0
6	7	2023-02-01 1:30:00	2023-02-01 1:45:00	122.0	34.0
7	8	2023-02-01 1:45:00	2023-02-01 2:00:00	90.0	35.0
8	9	2023-02-01 2:00:00	2023-02-01 2:15:00	90.0	35.0

9	10	2023-02-01 2:15:00	2023-02-01 2:30:00	90.0	35.0

	#	From	To (Interval: 15M)	Singrauli, Surya Kiran Bhawan Dudhichua PM10 (碌g/m3)	Singrauli, Surya Kiran Bhawan Dudhichu
8633	8634	2023-05-01 22:15:00	2023-05-01 22:30:00	28.00	7.00
8634	8635	2023-05-01 22:30:00	2023-05-01 22:45:00	28.00	7.00
8635	8636	2023-05-01 22:45:00	2023-05-01 23:00:00	19.00	11.00
8636	8637	2023-05-01 23:00:00	2023-05-01 23:15:00	19.00	11.00
8637	8638	2023-05-01 23:15:00	2023-05-01 23:30:00	19.00	11.00
8638	8639	2023-05-01 23:30:00	2023-05-01 23:45:00	19.00	11.00
8639	8640	2023-05-01 23:45:00	2023-05-02 0:00:00	32.00	6.00
8640	8641	Min	NaN	12.00	3.00
8641	8642	Max	NaN	847.00	474.00
8642	8643	Avg.	NaN	181.41	75.69

Report generated by YData (https://ydata.ai/?utm_source=opensource&utm_medium=pandasprofiling&utm_campaign=report).

After performing curve fitting, exploratory analysis, and forecasting on the NO2 data, we can gain insights into its role in acid rain. Here's a summary of the process and findings:

1. Curve Fitting: By fitting a curve to the NO2 data, we can analyze its historical trend and patterns. This helps us understand the overall behavior of NO2 levels over time. The curve fitting process may involve using regression models or other mathematical functions to find the best-fit curve that represents the data.

2. Exploratory Analysis: Exploratory analysis involves examining the relationships between NO2 levels and other relevant variables. We can explore correlations between NO2 and factors such as precipitation, temperature, wind patterns, or emissions from other pollutants. This analysis helps us identify potential contributing factors to NO2 levels and their impact on acid rain formation.

3. Forecasting: Forecasting NO2 levels allows us to predict its future behavior based on historical patterns and trends. Using time series forecasting techniques, such as ARIMA or exponential smoothing, we can estimate NO2 levels for specific time periods. These forecasts provide insights into potential future levels of NO2 and help in assessing its potential impact on acid rain formation.

4. Study on NO2 and Acid Rain: With the information obtained from curve fitting, exploratory analysis, and forecasting, we can study the role of NO2 in acid rain. Acid rain is caused by the release of pollutants into the atmosphere, including nitrogen oxides (NOx). NO2 is a major component of NOx emissions, and its presence contributes to the formation of acid rain when combined with other pollutants and atmospheric conditions.

By analyzing the relationship between NO2 levels and acid rain occurrences, we can assess the impact of NO2 on acid rain formation. This study may involve statistical analysis, modeling, and further investigation into the complex interactions between NO2, other pollutants, meteorological factors, and the chemistry of acid rain formation.

The findings from these analyses provide valuable insights into the role of NO2 in acid rain and help in understanding the mechanisms and factors influencing acid rain occurrence and severity. This knowledge can contribute to the development of effective strategies and policies to mitigate acid rain and reduce NO2 emissions for environmental conservation.