```
import pandas as pd
import numpy as np

import warnings
warnings.filterwarnings('ignore')
from scipy import stats
import statsmodels.api as sm
import matplotlib.pyplot as plt

%matplotlib inline
```

```
df = pd.read_csv('/content/city_day.csv')
```

```
df.head()
```

	City	Date	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	AQI	AQI_Bucket
0	Ahmedabad	2015-01-01	NaN	NaN	0.92	18.22	17.15	NaN	0.92	27.64	133.36	0.00	0.02	0.00	NaN	NaN
1	Ahmedabad	2015-01-02	NaN	NaN	0.97	15.69	16.46	NaN	0.97	24.55	34.06	3.68	5.50	3.77	NaN	NaN
2	Ahmedabad	2015-01-03	NaN	NaN	17.40	19.30	29.70	NaN	17.40	29.07	30.70	6.80	16.40	2.25	NaN	NaN
3	Ahmedabad	2015-01-04	NaN	NaN	1.70	18.48	17.97	NaN	1.70	18.59	36.08	4.43	10.14	1.00	NaN	NaN
4	Ahmedabad	2015-01-05	NaN	NaN	22.10	21.42	37.76	NaN	22.10	39.33	39.31	7.01	18.89	2.78	NaN	NaN

```
#df['Date'] = pd.to_datetime(df['Date'])
df.tail()
```

	City	Date	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	AQI	AQI_B
26214	Thiruvananthapuram	2020-04-27	14.13	34.27	5.60	8.98	12.48	5.65	0.49	5.50	42.41	NaN	NaN	NaN	63.0	Satis
26215	Thiruvananthapuram	2020-04-28	23.84	44.32	6.27	10.01	13.80	5.73	0.44	5.62	44.55	NaN	NaN	NaN	60.0	Satis
26216	Thiruvananthapuram	2020-04-29	18.54	34.48	6.17	9.67	13.35	5.93	0.51	5.52	38.97	NaN	NaN	NaN	57.0	Satis
26217	Thiruvananthapuram	2020-04-30	20.57	48.19	6.28	9.52	13.56	5.84	0.46	5.32	39.23	NaN	NaN	NaN	57.0	Satis
26218	Thiruvananthapuram	2020-05-01	17.58	37.49	2.56	7.84	9.34	4.85	0.45	7.10	31.16	NaN	NaN	NaN	82.0	Satis

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

```
City             0
Date             0
PM2.5         4289
PM10         10766
NO            3233
NO2           3217
NOx           4043
NH3           9847
CO            1961
SO2           3544
O3            3660
Benzene       5287
Toluene       7555
Xylene       16807
AQI           4282
AQI_Bucket    4282
dtype: int64
```

```
1 df.shape
```

```
(26219, 16)
```

```
df.describe()
```

	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O
count	21930.000000	15453.000000	22986.000000	23002.000000	22176.000000	16372.000000	24258.000000	22675.000000	22559.00000
mean	71.828907	125.492380	18.232528	29.009102	33.657667	25.336162	2.446235	14.988734	34.37424
std	67.161387	94.179918	23.081236	25.158322	32.695931	27.109244	7.385259	19.087561	21.38896
min	0.040000	0.010000	0.020000	0.010000	0.000000	0.010000	0.000000	0.010000	0.01000
25%	31.450000	60.990000	5.930000	11.850000	13.110000	9.197500	0.530000	5.560000	19.01000
50%	52.220000	100.950000	10.150000	21.830000	24.300000	17.820000	0.930000	9.120000	30.89000
75%	85.655000	158.460000	20.830000	37.890000	42.102500	32.600000	1.530000	15.595000	45.37000
max	949.990000	1000.000000	351.300000	362.210000	467.630000	352.890000	175.810000	193.860000	257.73000

```
df['PM2.5'].fillna(df['PM2.5'].mean(), inplace= True)
df['NO'].fillna(df['NO'].mean(), inplace= True)
df['NO2'].fillna(df['NO2'].mean(), inplace= True)
df['NOx'].fillna(df['NOx'].mean(), inplace= True)
df['NH3'].fillna(df['NH3'].mean(), inplace= True)
df['CO'].fillna(df['CO'].mean(), inplace= True)
df['SO2'].fillna(df['SO2'].mean(), inplace= True)
df['O3'].fillna(df['O3'].mean(), inplace= True)
df['Benzene'].fillna(df['Benzene'].mean(), inplace= True)
df['Toluene'].fillna(df['Toluene'].mean(), inplace= True)
df['Xylene'].fillna(df['Xylene'].mean(), inplace= True)
df['AQI'].fillna(df['AQI'].mean(), inplace= True)
```

```
df  = df.drop(['AQI_Bucket','PM10'], axis = 1)
```

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

```
City       0
Date       0
PM2.5      0
NO         0
NO2        0
NOx        0
NH3        0
CO         0
SO2        0
O3         0
Benzene    0
Toluene    0
Xylene     0
AQI        0
dtype: int64
```

```
1 df['Date'] = pd.to_datetime(df['Date'])
```

```
1 df = df.sort_index()
2
```

```
1 df.head()
```

	City	Date	PM2.5	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	AQI
0	Ahmedabad	2015-01-01	71.828907	0.92	18.22	17.15	25.336162	0.92	27.64	133.36	0.00	0.02	0.00	175.840908
1	Ahmedabad	2015-01-02	71.828907	0.97	15.69	16.46	25.336162	0.97	24.55	34.06	3.68	5.50	3.77	175.840908

```
df['City'].value_counts()
```

```
Ahmedabad             1948
Bengaluru             1948
Delhi                 1948
Chennai               1948
Mumbai                1948
Lucknow               1948
Hyderabad             1945
Patna                 1797
Gurugram              1618
Amritsar              1160
Jorapokhar            1108
Jaipur                1053
Thiruvananthapuram    1051
Amaravati              890
Brajrajnagar           877
Talcher                864
Kolkata                753
Guwahati               441
Shillong               249
Chandigarh             243
Bhopal                 228
Kochi                  101
Ernakulam              101
Aizawl                  52
Name: City, dtype: int64
```

Forecasting for patna

before corona

```
1 df_patna= df[df['City'] == 'Patna']
2 df_patna = df_patna.drop('City' , axis = 1)
3 df_patna.set_index('Date', inplace=True)
4
```

```
1
```

```
1 df_patna.shape
```

```
(1797, 12)
```

```
1 df_patna.head()
```

Date	PM2.5	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	AQI
2015-06-01	71.828907	14.41	25.06	39.32	25.336162	1.56	1.80	8.89	0.00	0.29	0.00	175.840908
2015-06-02	71.828907	25.00	22.48	47.50	25.336162	2.35	9.69	9.90	0.08	0.83	0.09	175.840908
2015-06-03	71.828907	14.29	17.16	29.81	25.336162	1.69	20.61	12.63	0.00	0.33	0.00	175.840908
2015-06-04	71.828907	13.03	15.62	28.63	25.336162	1.20	4.35	9.77	0.01	0.28	0.00	175.840908
2015-06-05	71.828907	10.40	10.36	20.14	25.336162	1.29	7.22	11.90	0.00	0.15	0.00	175.840908

```
1 df_patna_before_covid = df_patna.loc['2016-01-01' : '2020-03-23' ]
2 df_patna_after_covid = df_patna.loc['2020-03-24' : ]
```

```
1
2
```

```
1 df_patna_before_covid.head()
```

Date	PM2.5	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	AQI
2016-01-01	553.63	161.83	39.36	201.04	25.336162	5.75	6.49	36.11	0.88	2.61	2.34	619.0
2016-01-02	454.15	122.14	35.62	167.56	25.336162	3.45	7.22	27.27	0.73	2.00	1.67	598.0
2016-01-03	406.49	96.86	36.59	133.45	25.336162	3.80	2.73	14.47	0.71	2.15	1.79	522.0
2016-01-04	361.88	131.69	34.76	166.08	25.336162	3.55	6.62	25.20	0.66	1.93	1.63	501.0
2016-01-05	373.44	152.73	39.02	141.81	25.336162	3.65	7.05	38.77	0.71	2.07	1.78	490.0

```
1 df_patna_before_covid.shape
```

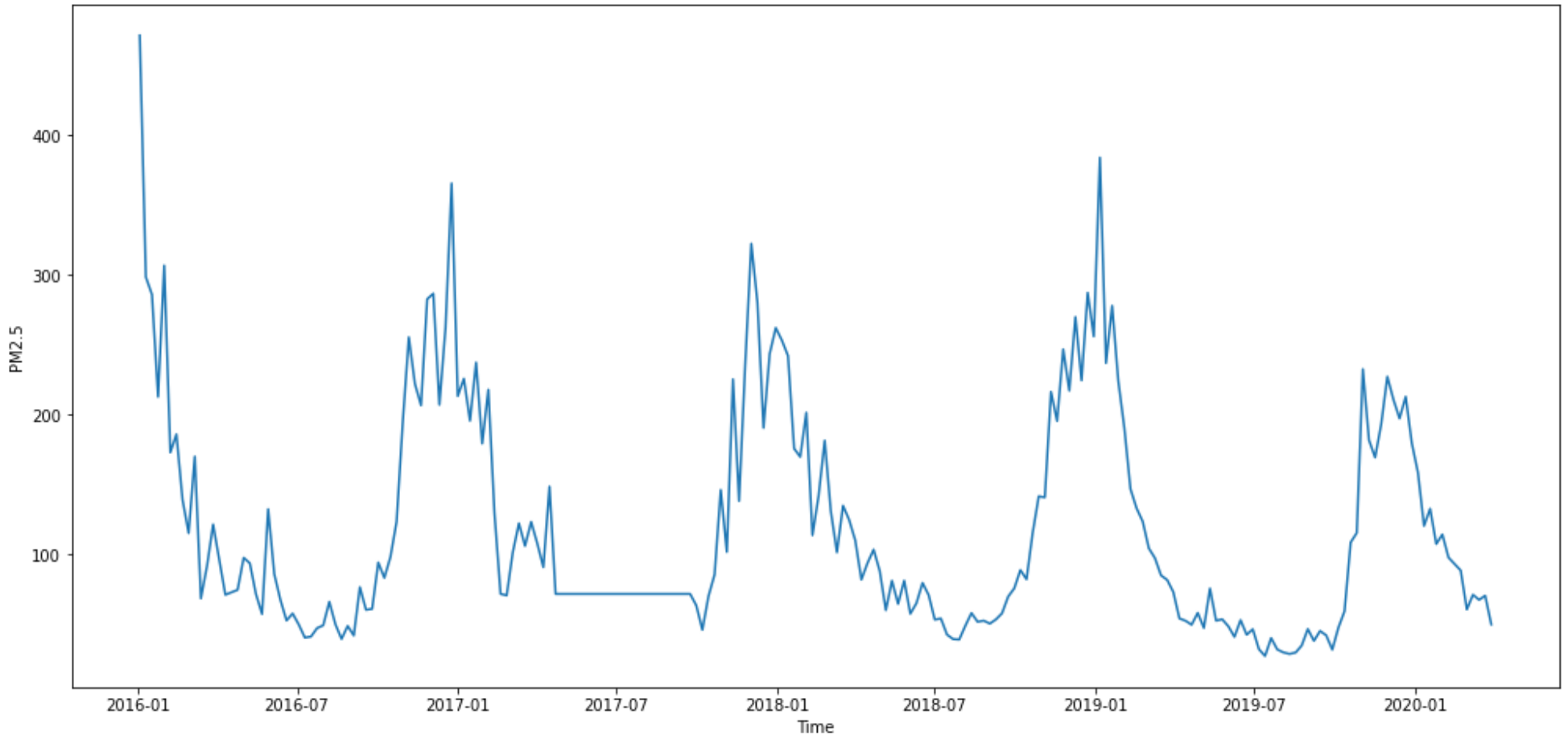
```
(1544, 12)
```

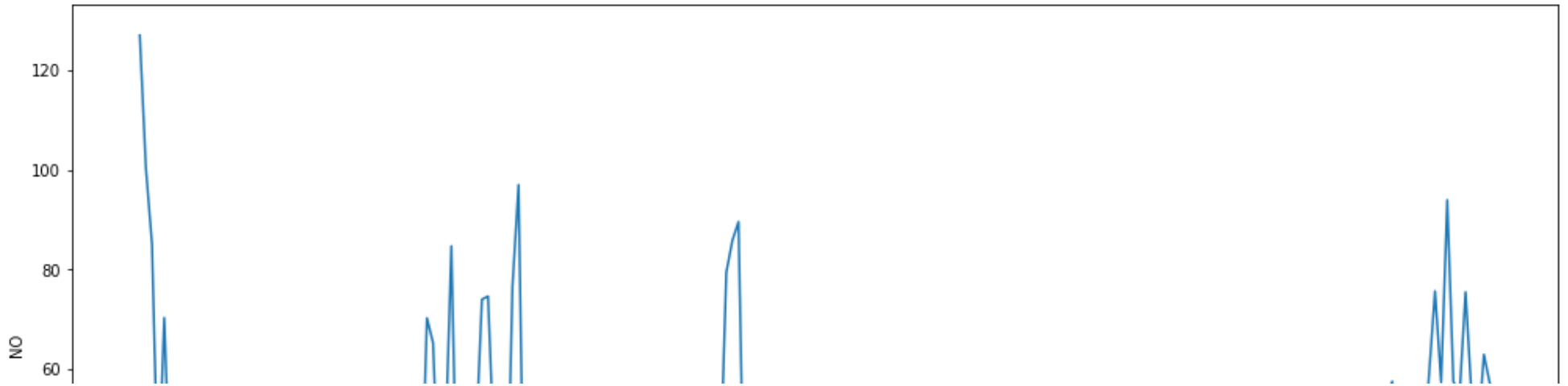
```
1 df_patna_before_covid=  df_patna_before_covid.resample('W').mean()
```

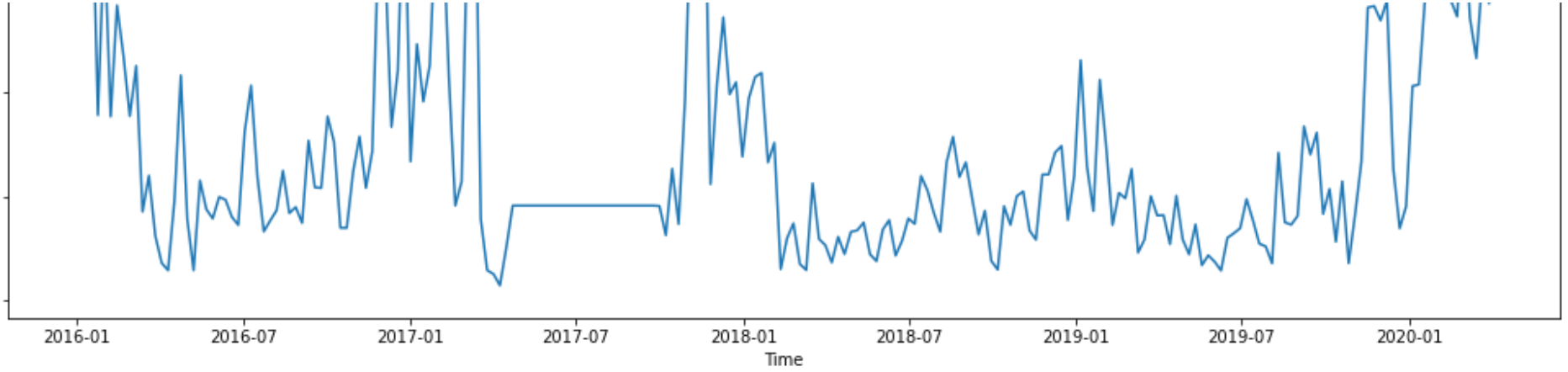
```
1 df_patna_before_covid.head()
```

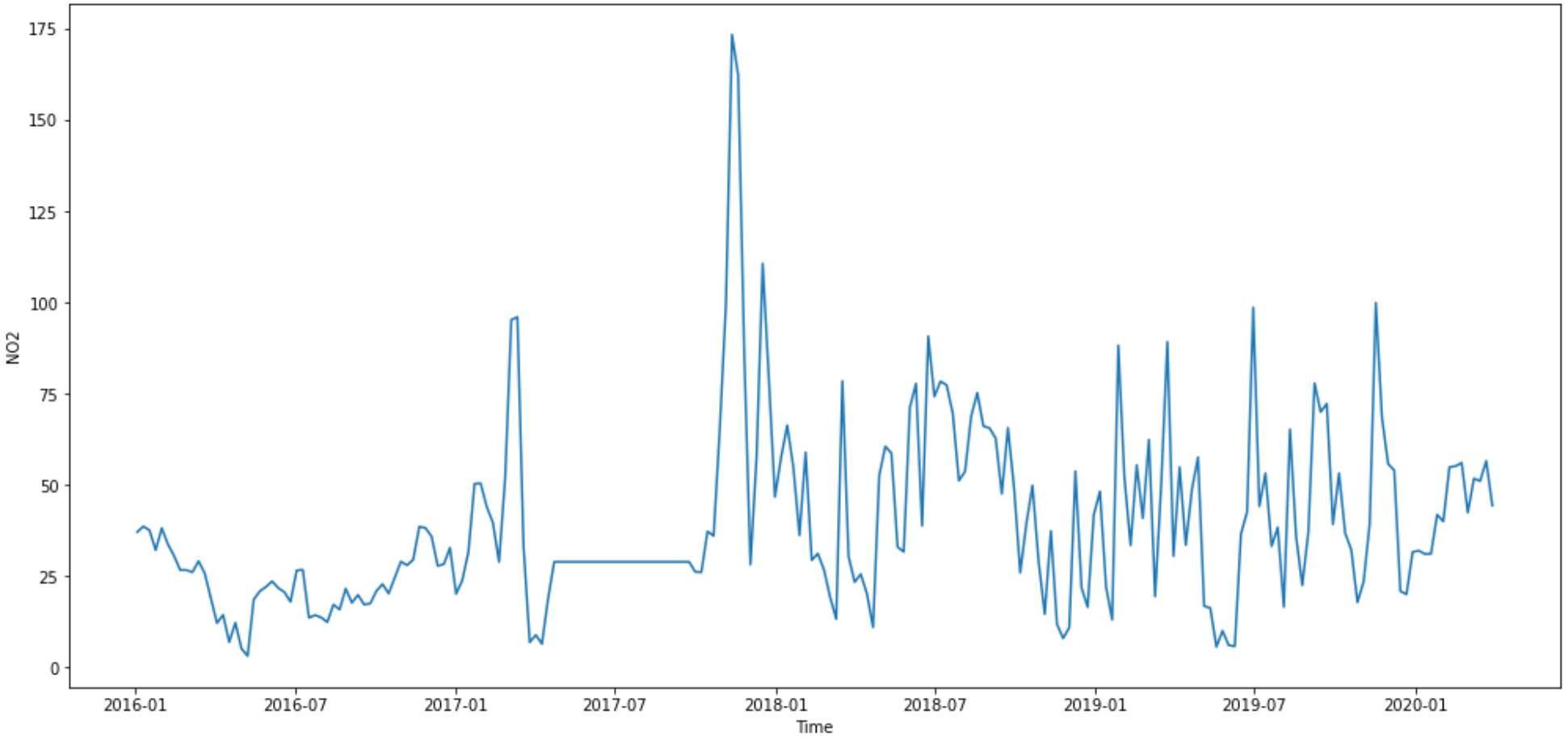
Date	PM2.5	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene

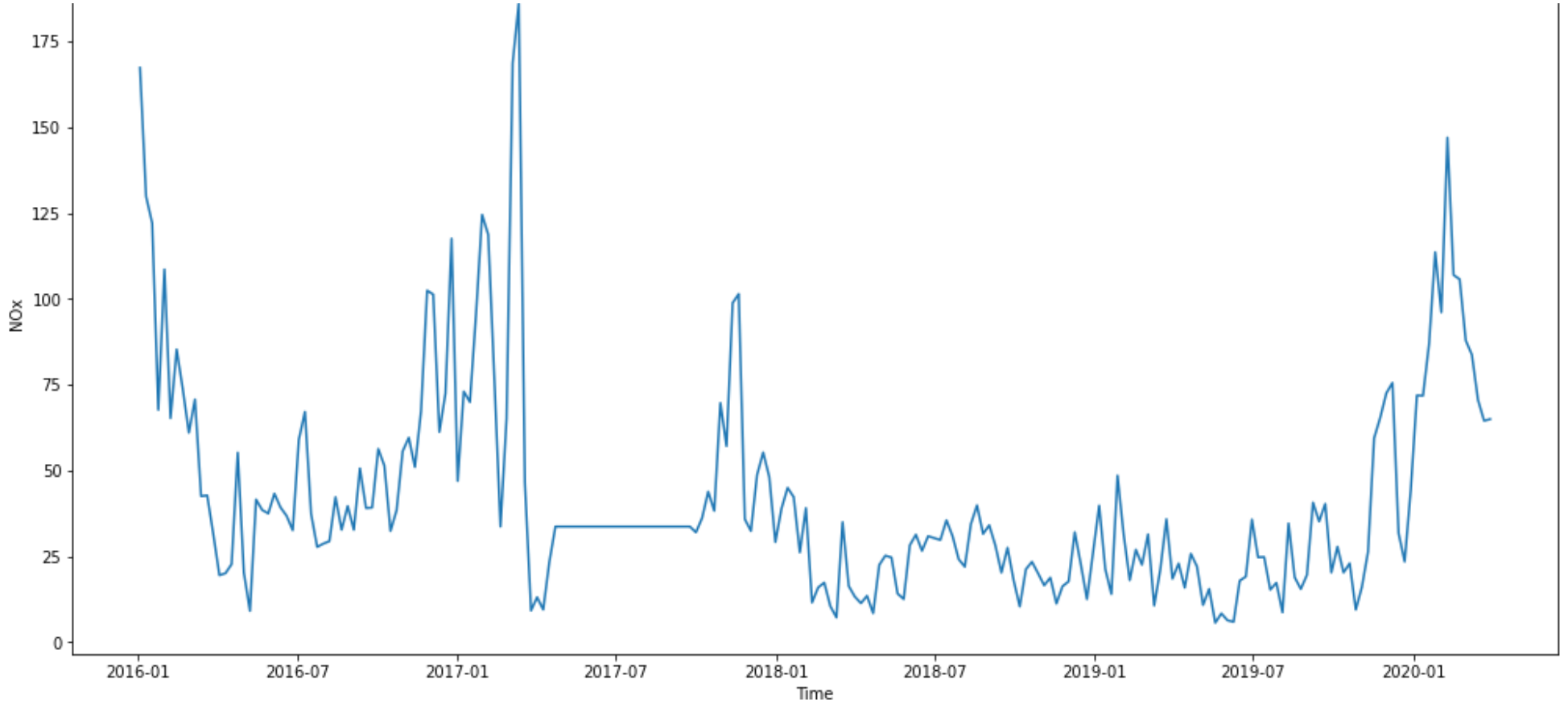
```
def plot_data(col):
    plt.figure(figsize=(17, 8))
    plt.plot(df_patna_before_covid[col])
    plt.xlabel('Time')
    plt.ylabel(col)
    plt.grid(False)
    plt.show()

for col in df_patna_before_covid.columns:
    plot_data(col)
```

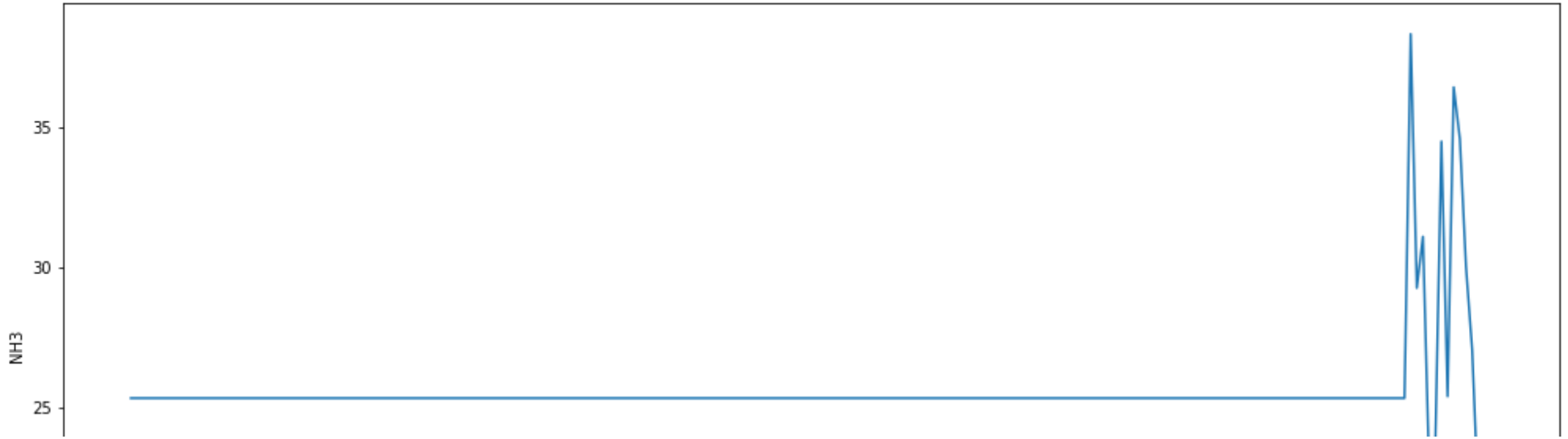

PM2.5
400
300
200
100
2016-01
2016-07
2017-01
2017-07
2018-01
2018-07
2019-01
2019-07
2020-01
Time

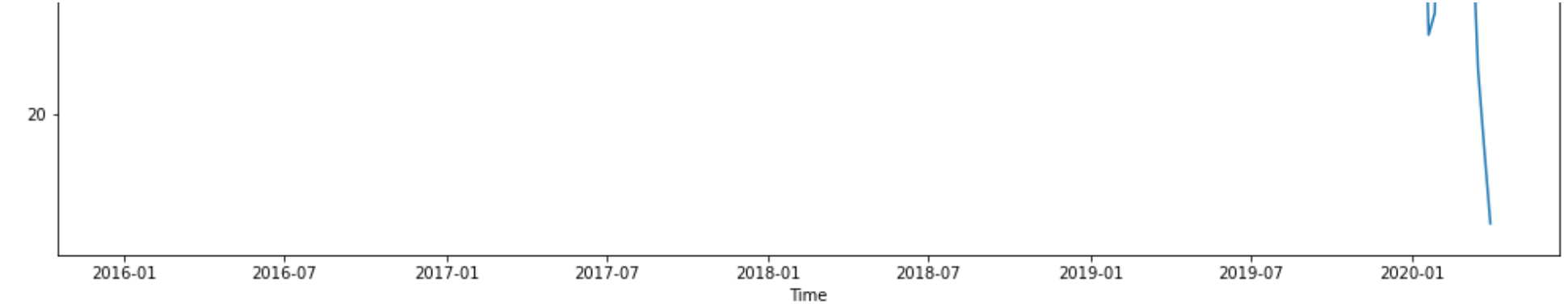

NO
120
100
80
60

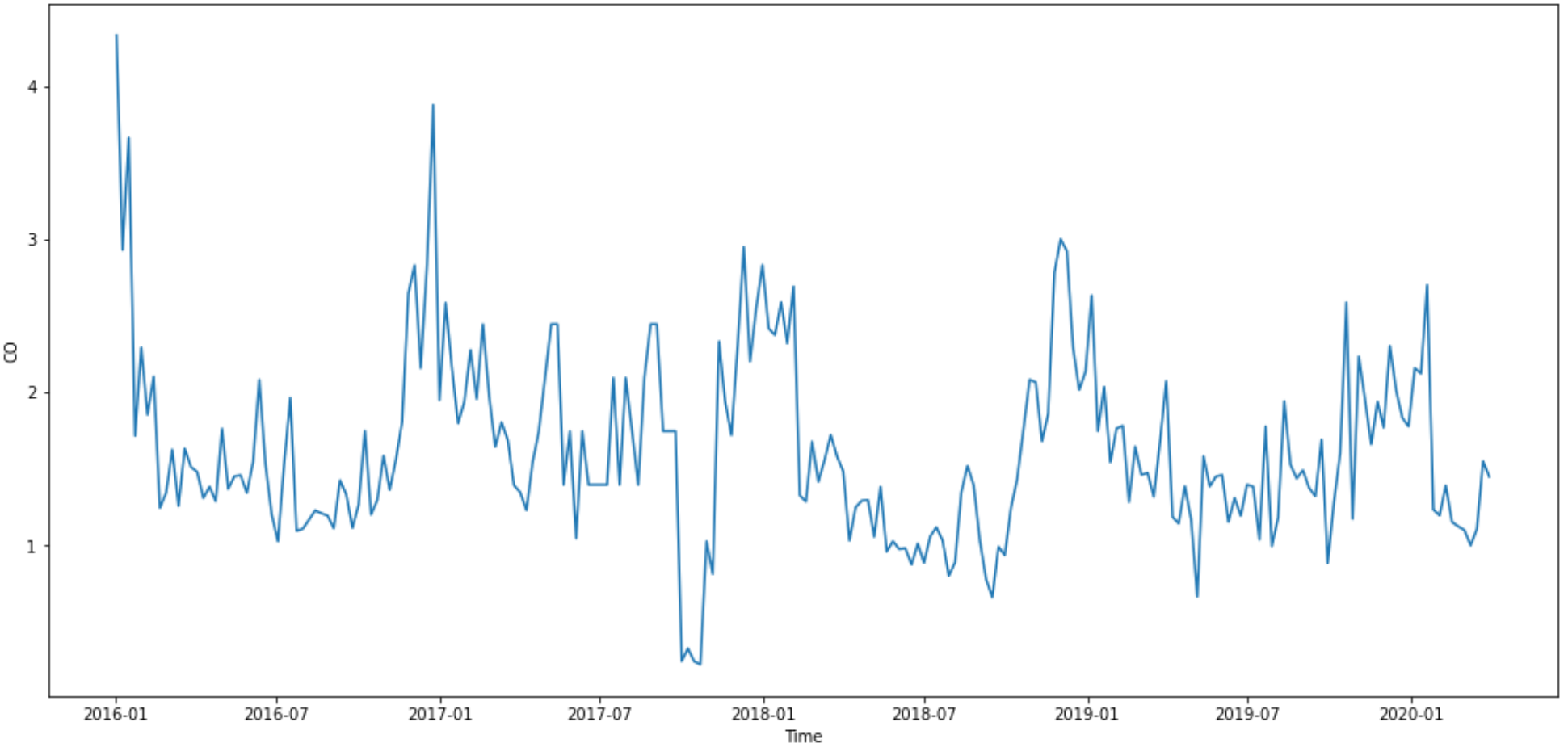

40
20
0
2016-01
2016-07
2017-01
2017-07
2018-01
2018-07
2019-01
2019-07
2020-01
Time

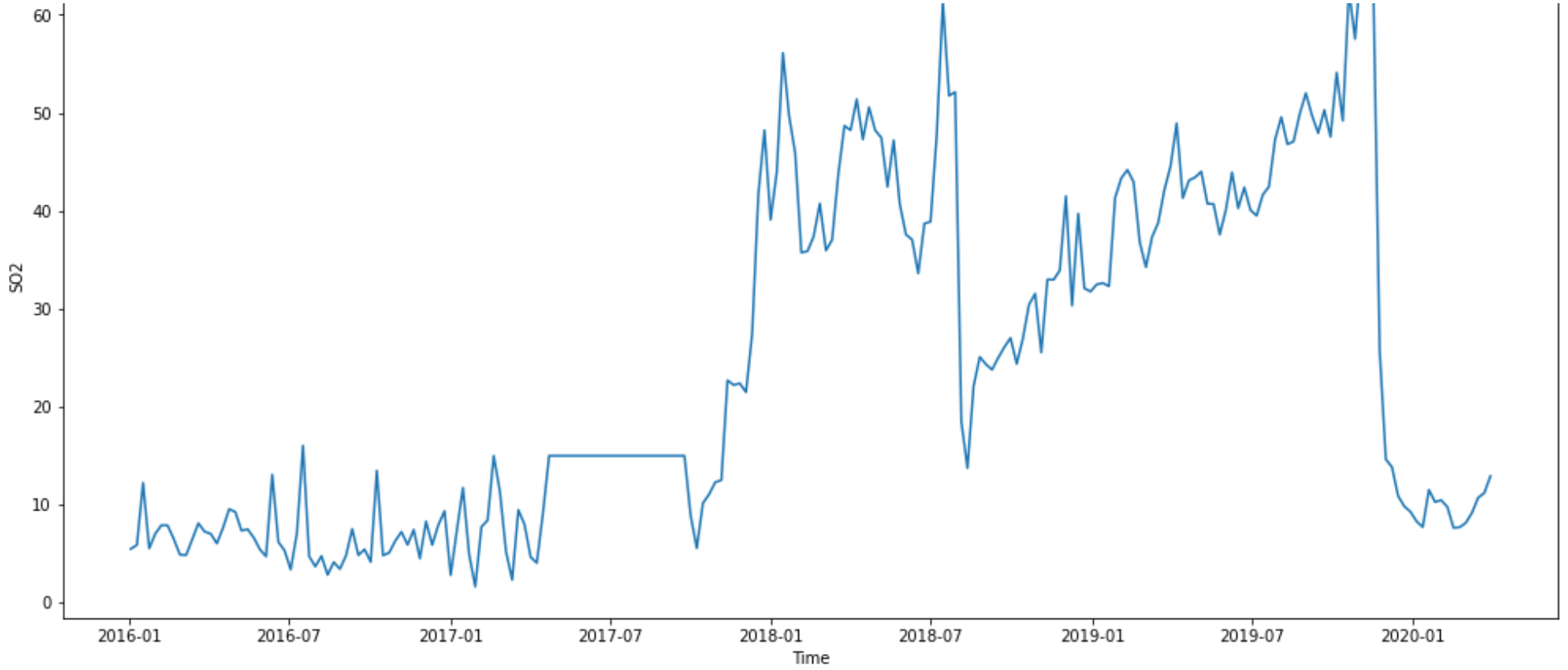

175
150
125
100
75
50
25
0
NO2
2016-01
2016-07
2017-01
2017-07
2018-01
2018-07
2019-01
2019-07
2020-01
Time

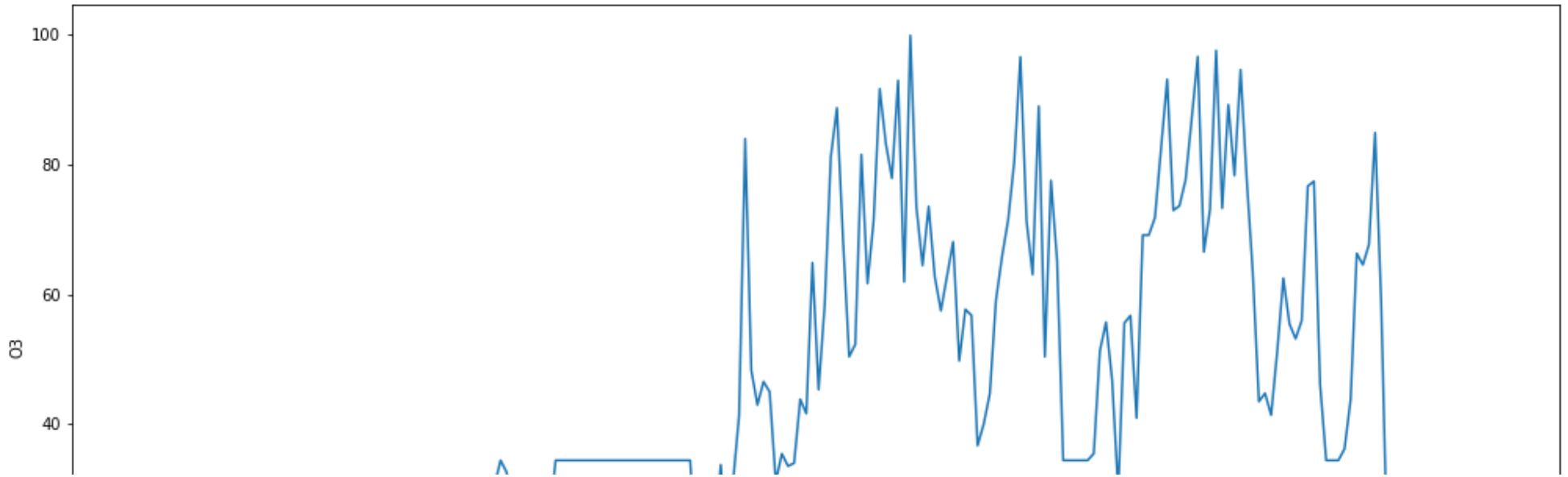

NOx
175
150
125
100
75
50
25
0
2016-01
2016-07
2017-01
2017-07
2018-01
2018-07
2019-01
2019-07
2020-01
Time

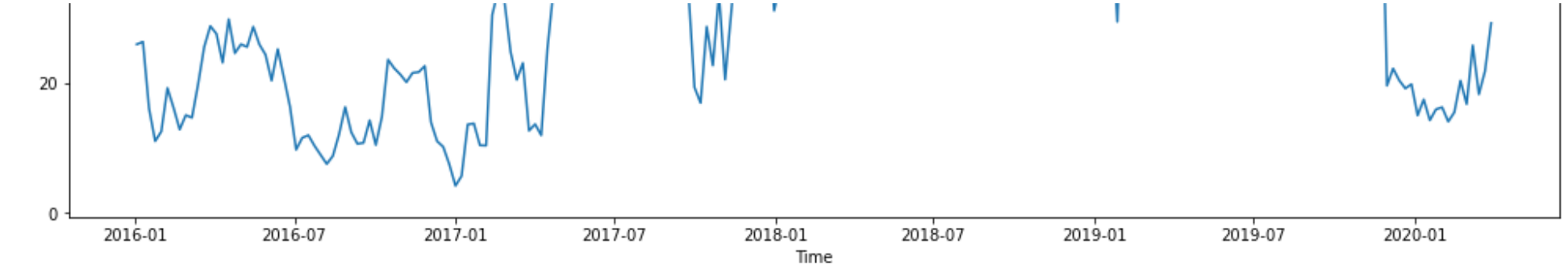

NH3
35
30
25

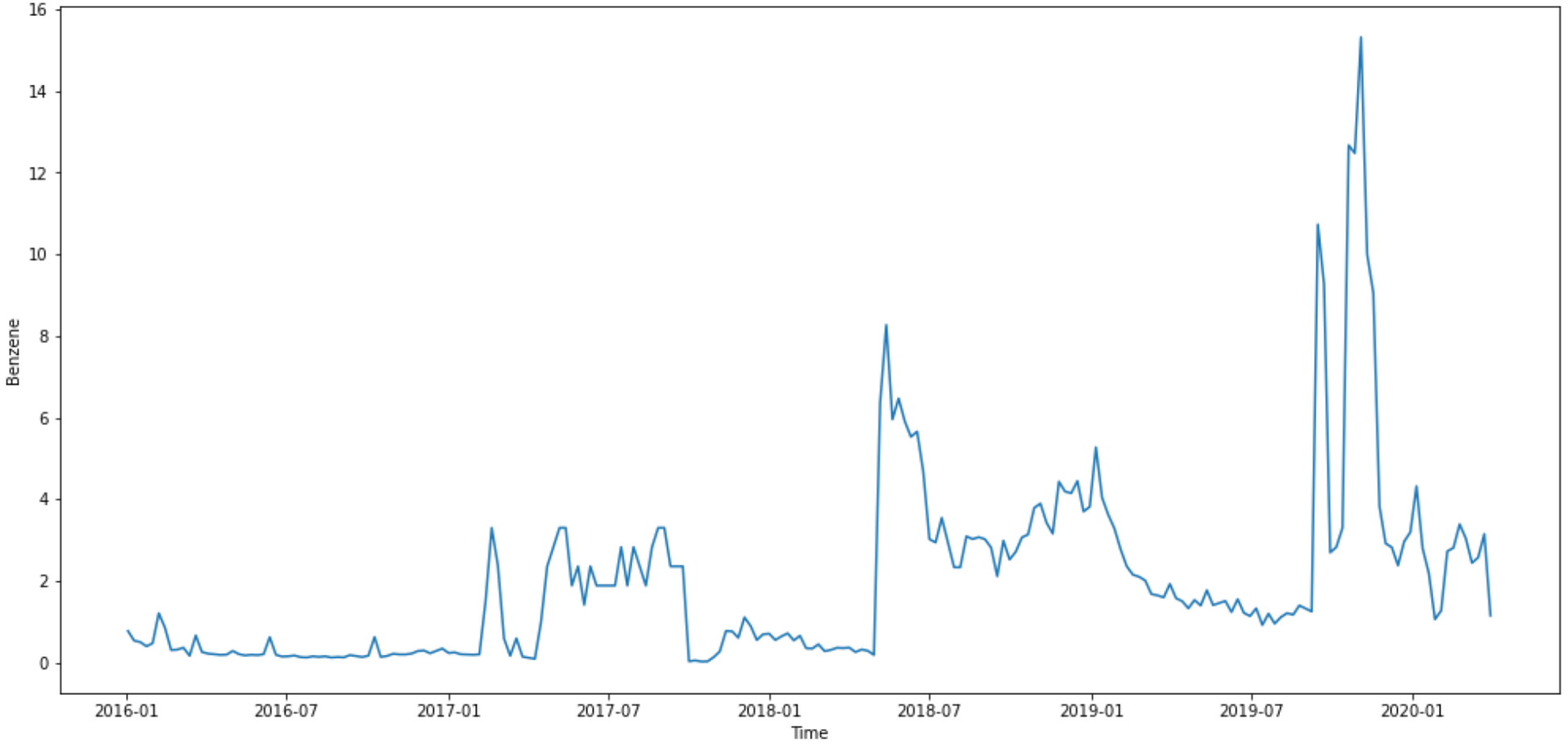

20
2016-01
2016-07
2017-01
2017-07
2018-01
2018-07
2019-01
2019-07
2020-01
Time


CO
4
3
2
1
2016-01
2016-07
2017-01
2017-07
2018-01
2018-07
2019-01
2019-07
2020-01
Time

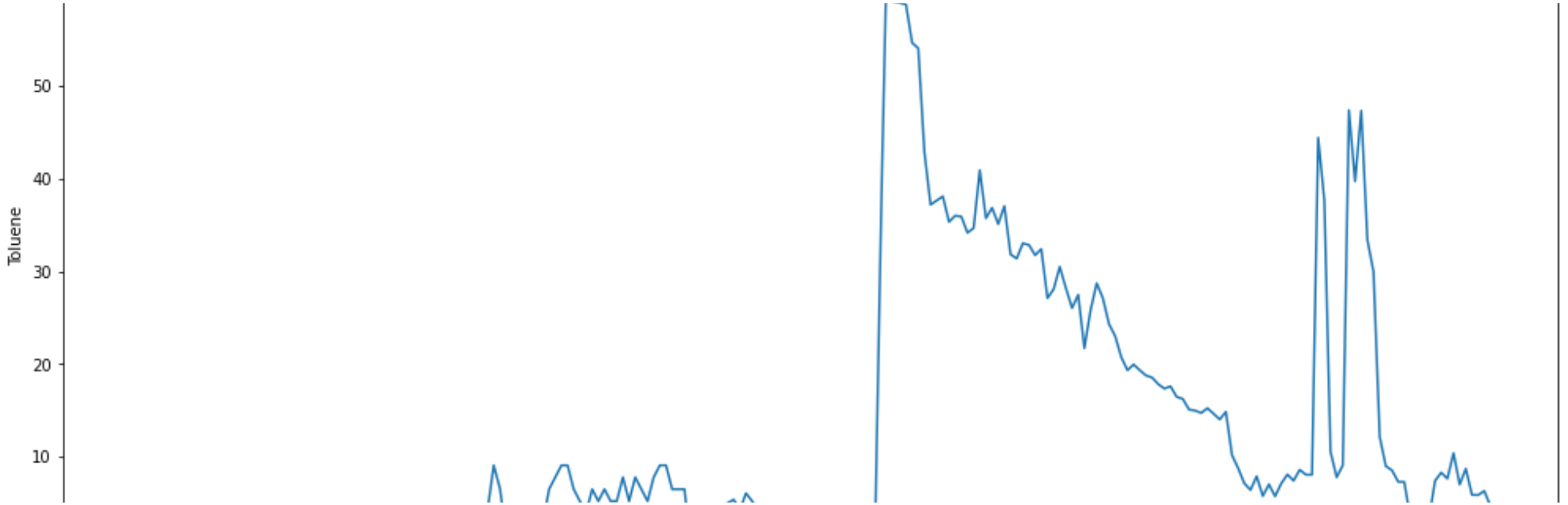

SO2
60
50
40
30
20
10
0
2016-01
2016-07
2017-01
2017-07
2018-01
2018-07
2019-01
2019-07
2020-01
Time


O3
100
80
60
40


20
0
2016-01
2016-07
2017-01
2017-07
2018-01
2018-07
2019-01
2019-07
2020-01
Time


16
14
12
10
8
6
4
2
0
Benzene
2016-01
2016-07
2017-01
2017-07
2018-01
2018-07
2019-01
2019-07
2020-01
Time


70
60




```
1 df_patna_before_covid = df_patna_before_covid.reset_index()
2 df_patna_before_covid = df_patna_before_covid.rename({'Date':'ds','AQI':'y'} , axis = 1)
3 df_patna_before_covid.head(1)
```

	ds	PM2.5	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	y
0	2016-01-03	471.423333	126.943333	37.19	167.35	25.336162	4.333333	5.48	25.95	0.773333	2.253333	1.933333	579.666667

Modeling on before dataset

```
1 train  = df_patna_before_covid
2
3 cols = train.columns.tolist()
```

```
1 cols = ['ds',
```

```
 2  'y',
 3  'PM2.5',
 4  'NO',
 5  'NO2',
 6  'NOx',
 7  'NH3',
 8  'CO',
 9  'SO2',
10  'O3',
11  'Benzene',
12  'Toluene',
13  'Xylene'
14  ]
```

```
1 train = train[cols]
2 train_y = train[['ds','y']]
3 train_y.head(3)
```

	ds	y
0	2016-01-03	579.666667
1	2016-01-10	449.714286
2	2016-01-17	426.714286

```
1 from fbprophet import Prophet
2 import logging
3
4 logging.getLogger().setLevel(logging.ERROR)
```

```
1 m = Prophet()
```

```
1 #m.add_regressor('PM2.5')
2 #m.add_regressor('NO')
```

```
#m.add_regressor('NO2')
#m.add_regressor('NOx')
#m.add_regressor('NH3')
#m.add_regressor('CO')
#m.add_regressor('SO2')
#m.add_regressor('O3')
#m.add_regressor('Benzene')
#m.add_regressor('Toluene')
#m.add_regressor('Xylene')
```

```
m.fit(train_y)
```

```
INFO:numexpr.utils:NumExpr defaulting to 2 threads.
INFO:fbprophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to override this.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
<fbprophet.forecaster.Prophet at 0x7f557fc089e8>
```

```
future = m.make_future_dataframe(periods= 365 )
forecast = m.predict(future)
forecast.tail()
```

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_terms	additive_terms_lower	additive
582	2021-03-25	138.923322	82.347472	179.062137	136.085791	141.946882	-7.362679	-7.362679	
583	2021-03-26	138.814162	80.549388	176.271504	135.969158	141.850220	-10.036569	-10.036569	
584	2021-03-27	138.705003	77.922791	168.797747	135.848338	141.756048	-12.833369	-12.833369	
585	2021-03-28	138.595843	73.825464	175.172603	135.725612	141.673429	-15.717193	-15.717193	
586	2021-03-29	138.486684	75.588540	168.888889	135.600467	141.580339	-18.649997	-18.649997	

```
1
```

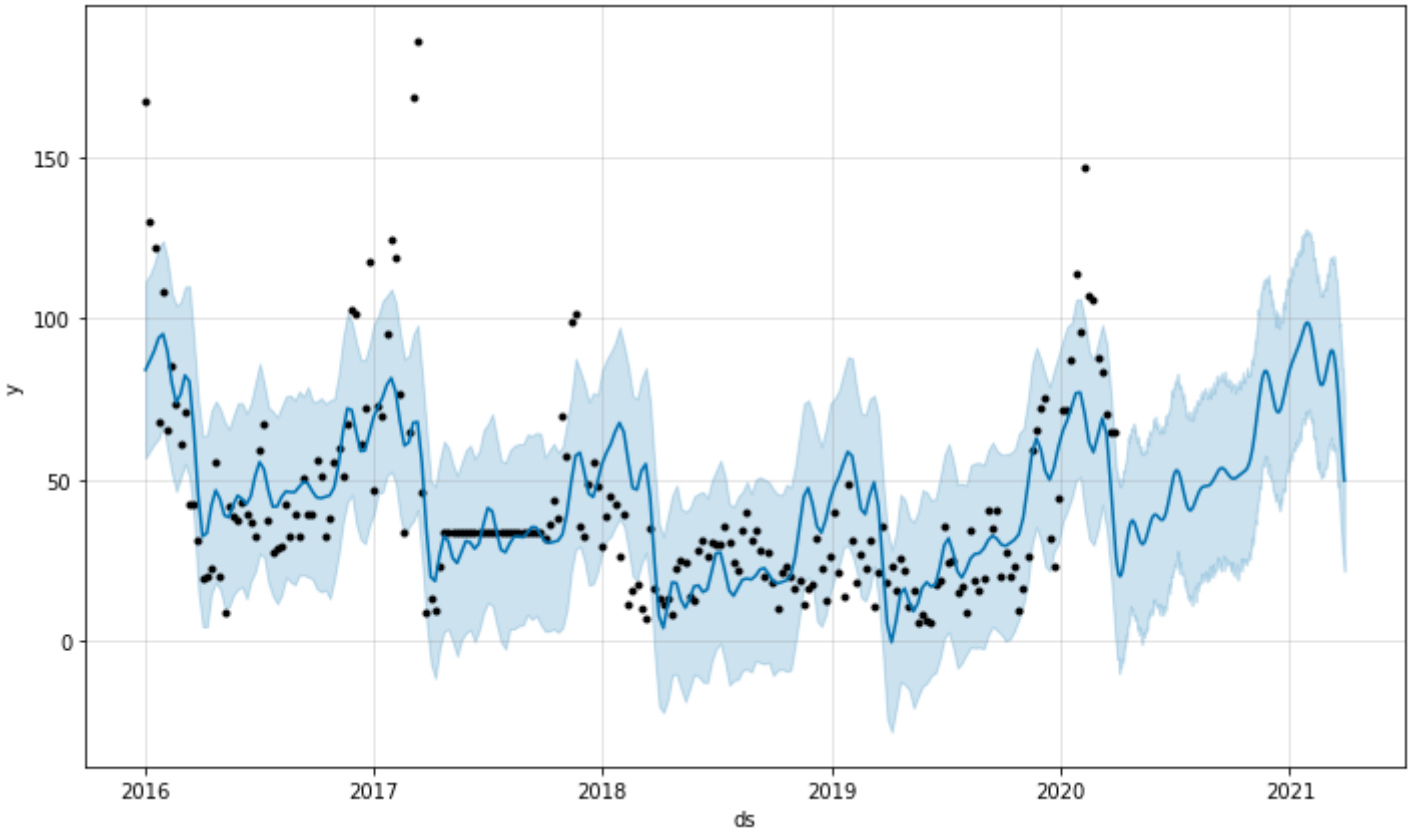
```
1 train_nox = train[['ds','NOx']]
2 train_nox = train_nox.rename({'NOx':'y'},axis=1)
3 m = Prophet()
4 m.fit(train_nox)
```

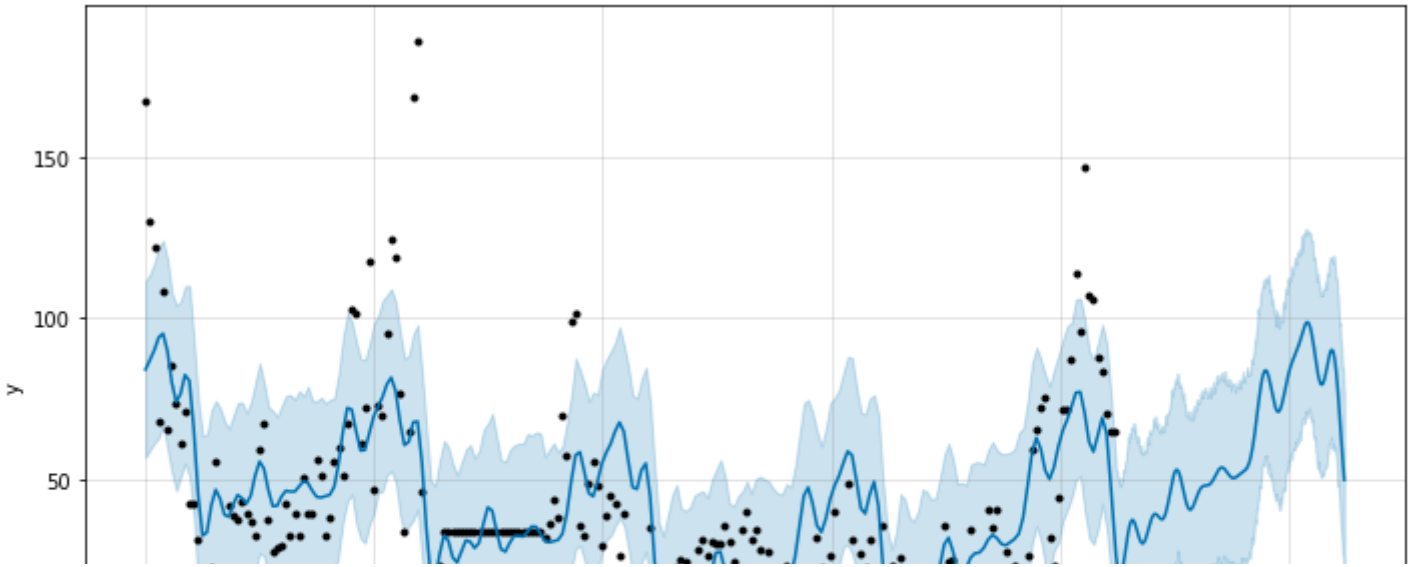
```
INFO:fbprophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to override this.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
<fbprophet.forecaster.Prophet at 0x7f557d340630>
```

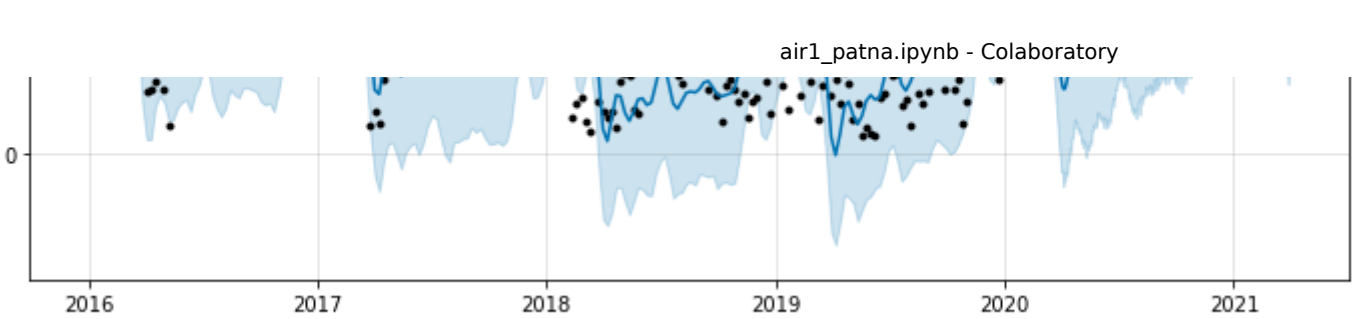
```
1 future = m.make_future_dataframe(periods= 365 )
2 forecast = m.predict(future)
3 forecast.tail()
```

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_terms	additive_terms_lower	additive_
582	2021-03-25	68.865082	30.298461	89.683481	65.850106	72.271802	-8.720750	-8.720750	
583	2021-03-26	68.922375	26.815056	84.363644	65.898336	72.334995	-11.579324	-11.579324	
584	2021-03-27	68.979668	26.192710	84.633027	65.945527	72.398027	-14.316072	-14.316072	
585	2021-03-28	69.036961	24.292331	79.675678	65.987858	72.461058	-16.891358	-16.891358	
586	2021-03-29	69.094254	21.669295	76.381633	66.029301	72.524090	-19.269002	-19.269002	

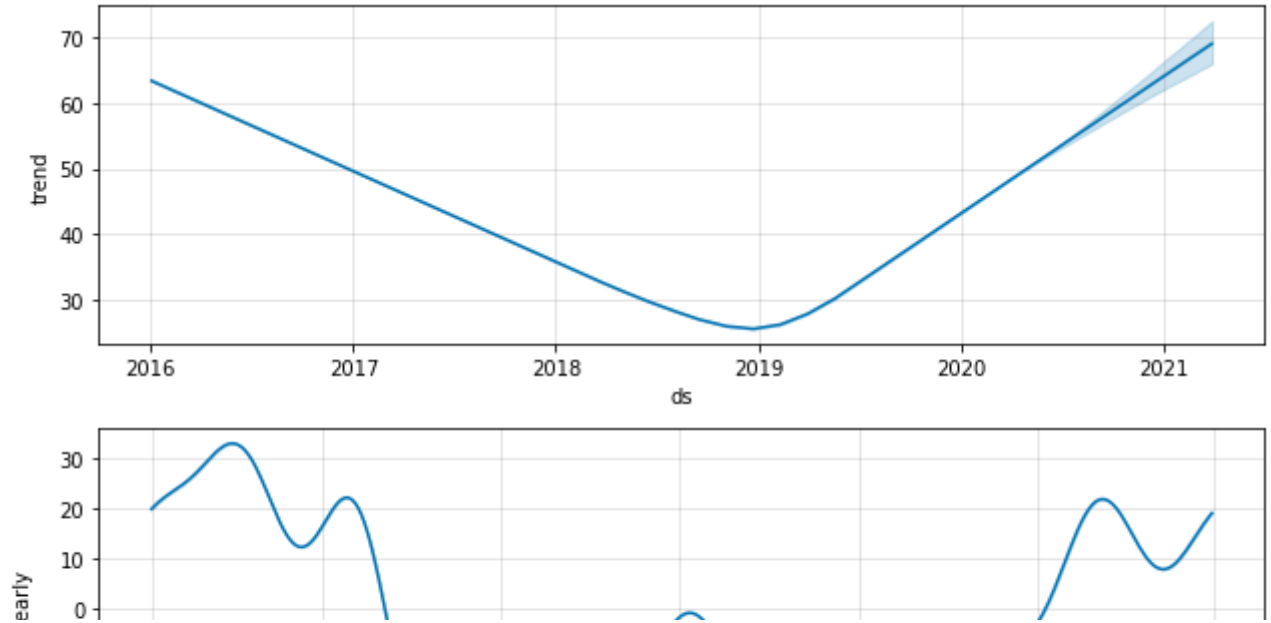
```
1 m.plot(forecast)
```


150
100
50
0
y
2016
2017
2018
2019
2020
2021
ds


150
100
50
y




```
1 m.plot_components(forecast)
```

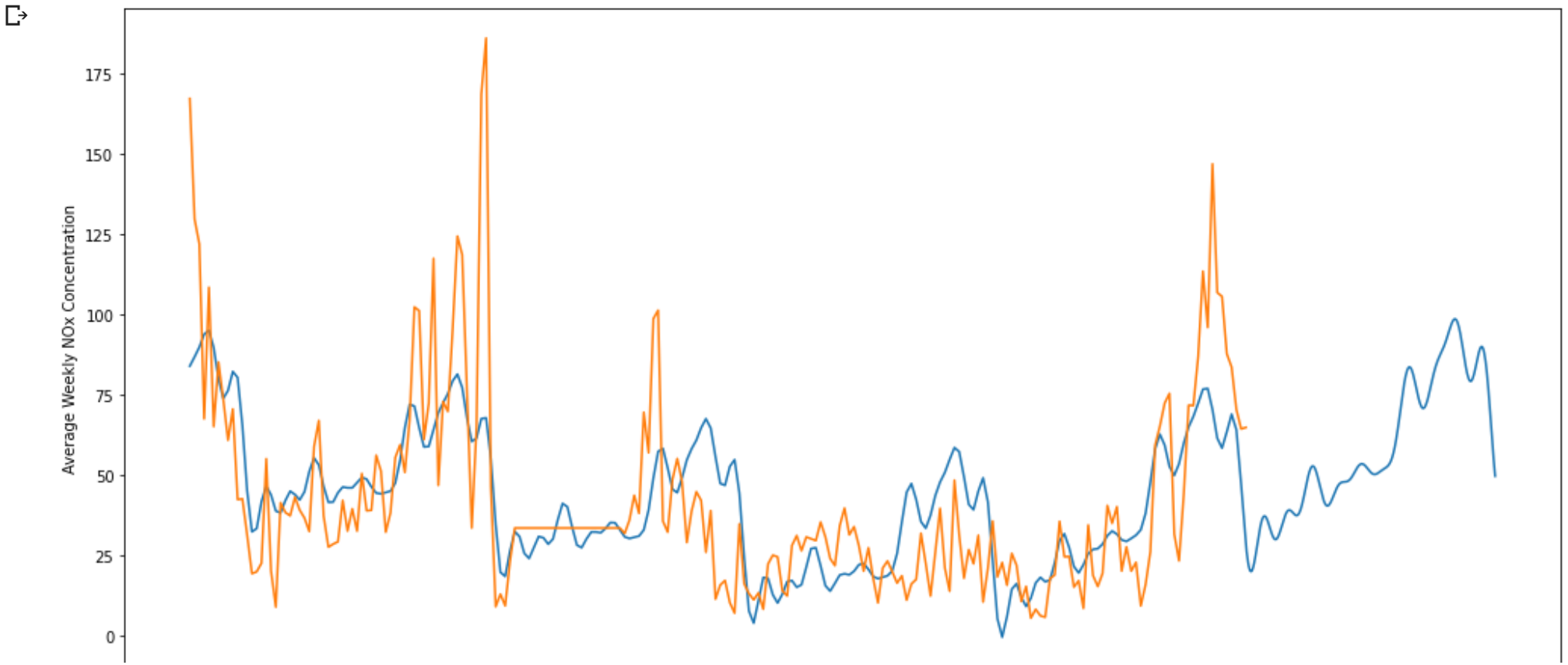



```
1 def make_comparison_dataframe(historical, forecast):
2     return forecast.set_index('ds')[['yhat', 'yhat_lower', 'yhat_upper']].join(historical.set_index
3 cmp_df = make_comparison_dataframe(train_nox, forecast)
4
5 cmp_df.head()
```

ds	yhat	yhat_lower	yhat_upper	y
2016-01-03	84.081813	56.747349	111.588069	167.350000
2016-01-10	86.945799	59.129156	113.712374	129.845714
2016-01-17	89.985114	61.239695	117.232319	122.070000
2016-01-24	93.969317	62.861407	122.464687	67.657143
2016-01-31	95.208703	67.206592	124.036524	108.597143

```
1 plt.figure(figsize=(17, 8))
```

```
plt.plot(cmp_df['yhat'])

plt.plot(cmp_df['y'])
plt.xlabel('Time')
plt.ylabel('Average Weekly NOx Concentration')
plt.grid(False)
plt.show()

```




```
1
```

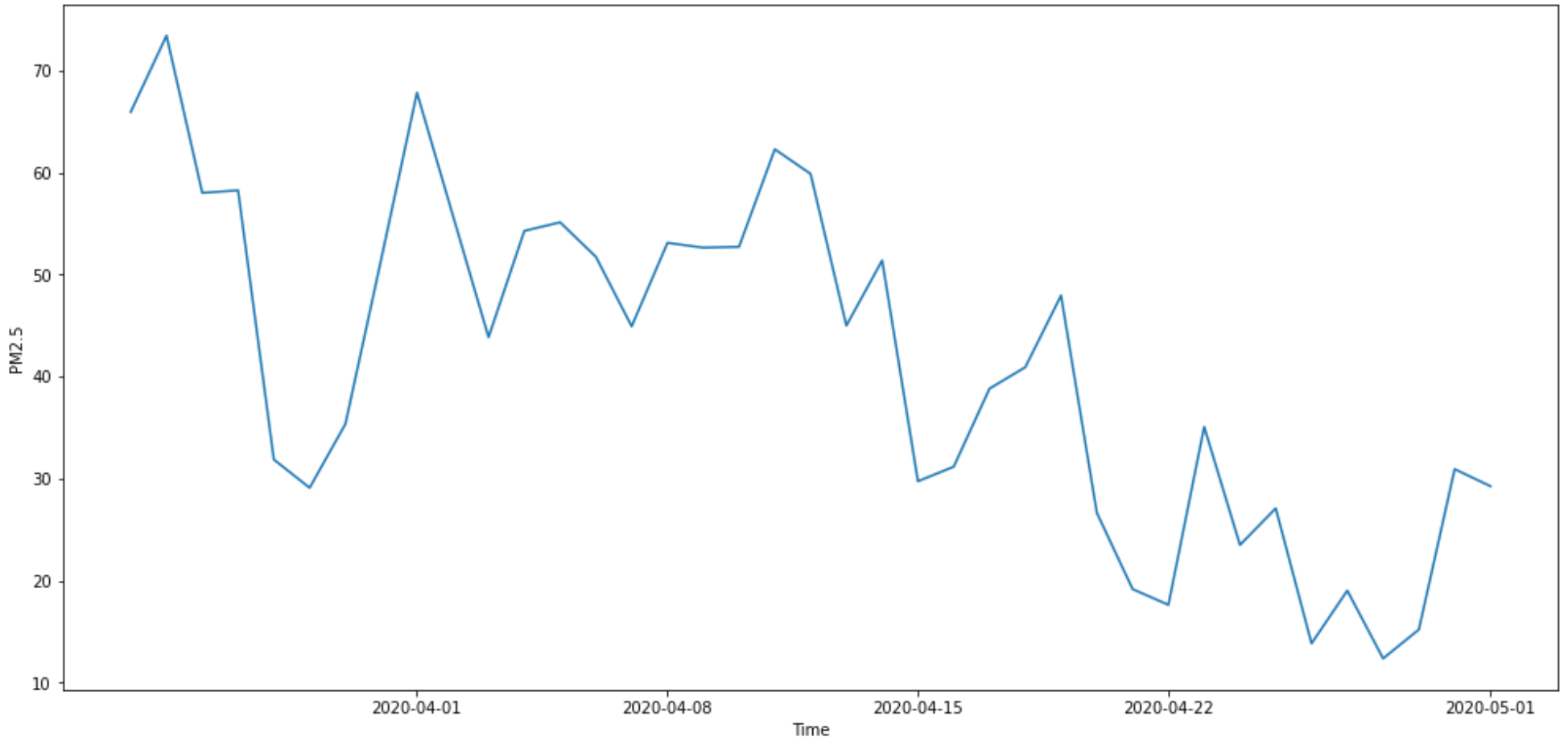
During corona

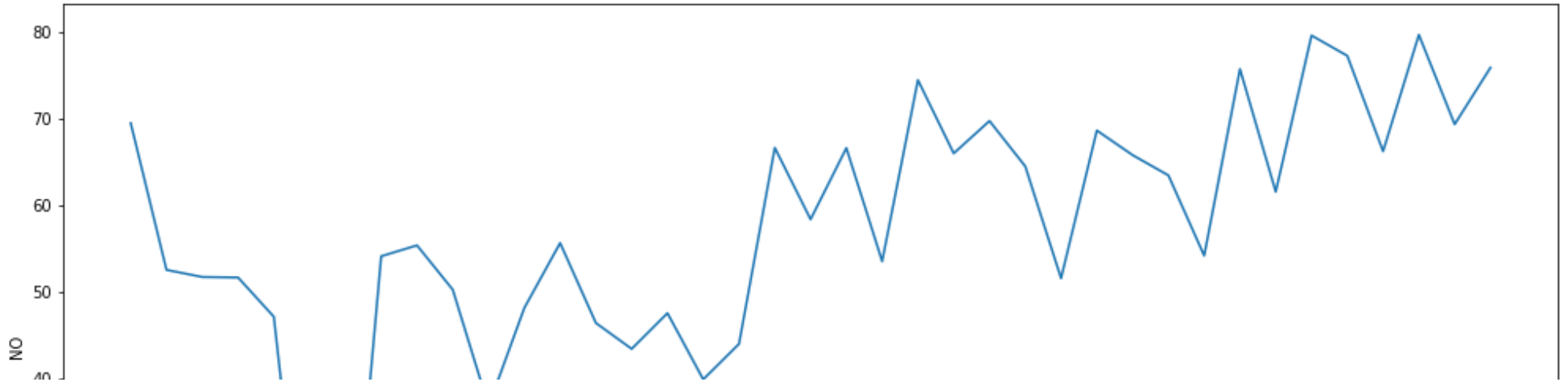
Double-click (or enter) to edit

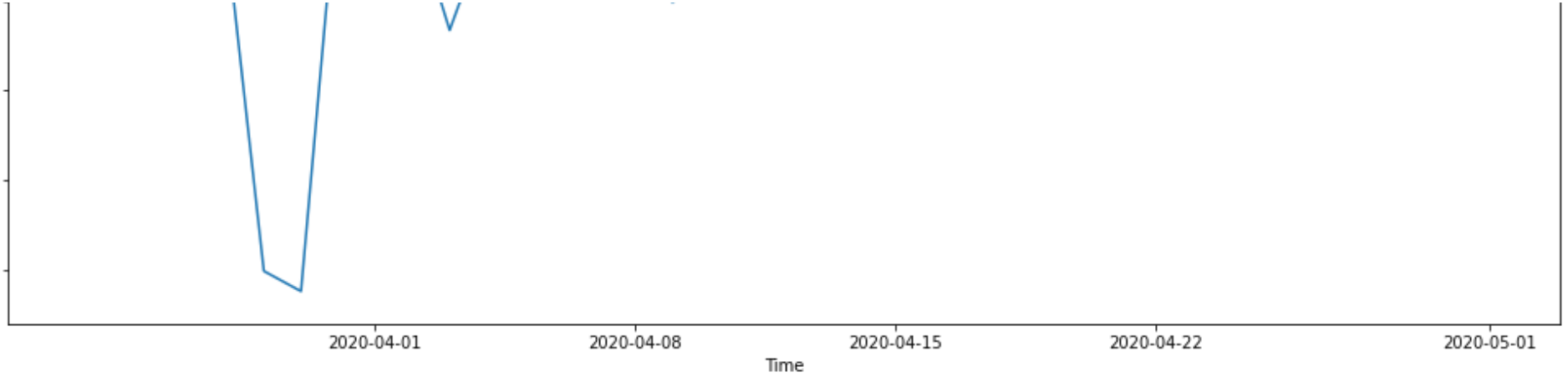
```
df_patna_after_covid.head()
```

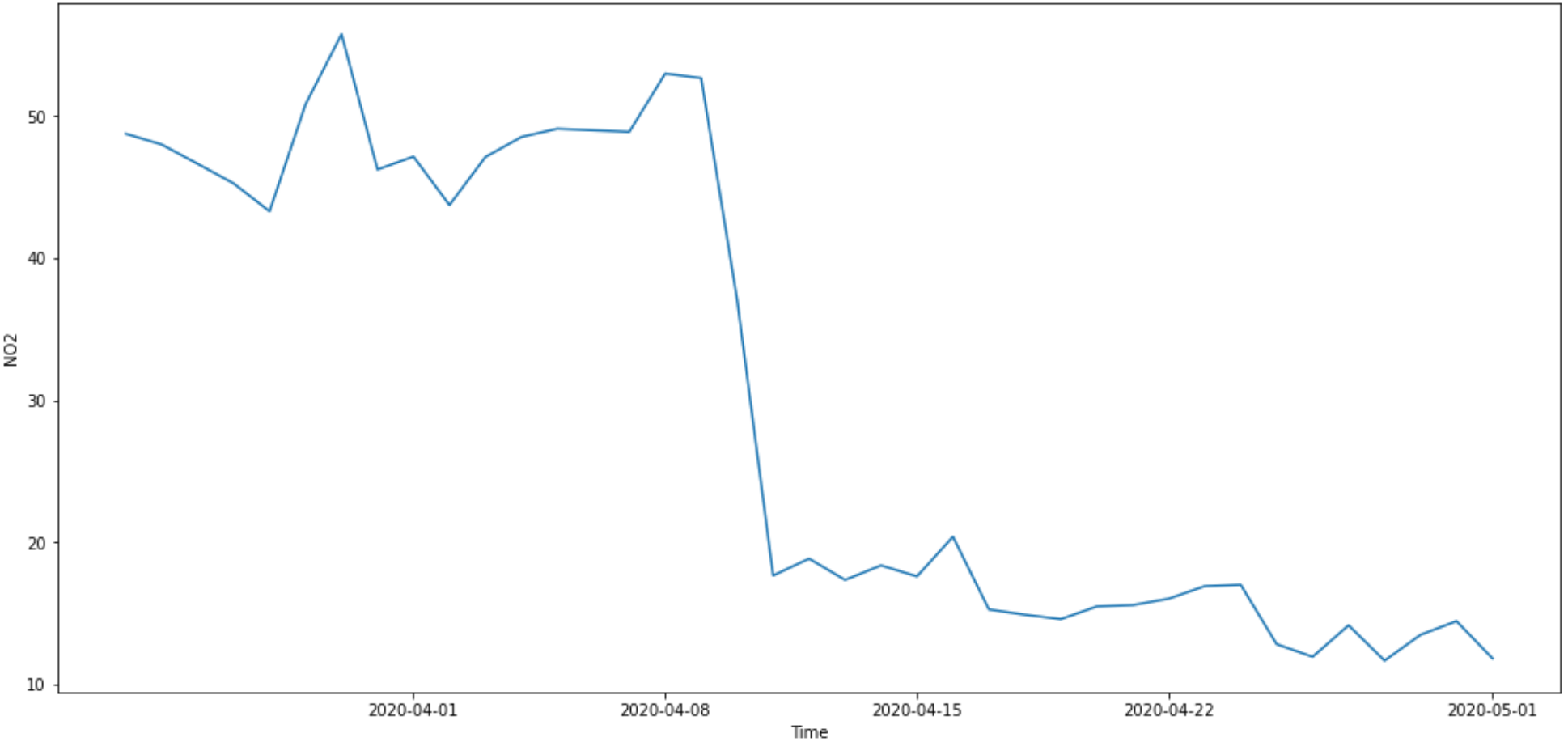
Date	PM2.5	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	AQI
2020-03-24	65.94	69.45	48.75	80.41	14.50	1.63	13.09	19.97	1.16	5.41	0.35	177.0
2020-03-25	73.42	52.49	47.99	66.72	14.13	1.74	13.57	21.77	1.20	5.50	0.28	166.0
2020-03-26	58.01	51.67	46.63	65.41	14.24	1.71	13.26	20.18	0.90	5.14	0.28	152.0
2020-03-27	58.25	51.61	45.25	64.28	13.88	1.68	12.52	20.63	0.89	4.96	0.31	146.0
2020-03-28	31.87	47.07	43.29	59.88	12.38	1.49	11.73	15.88	0.48	3.88	0.30	122.0

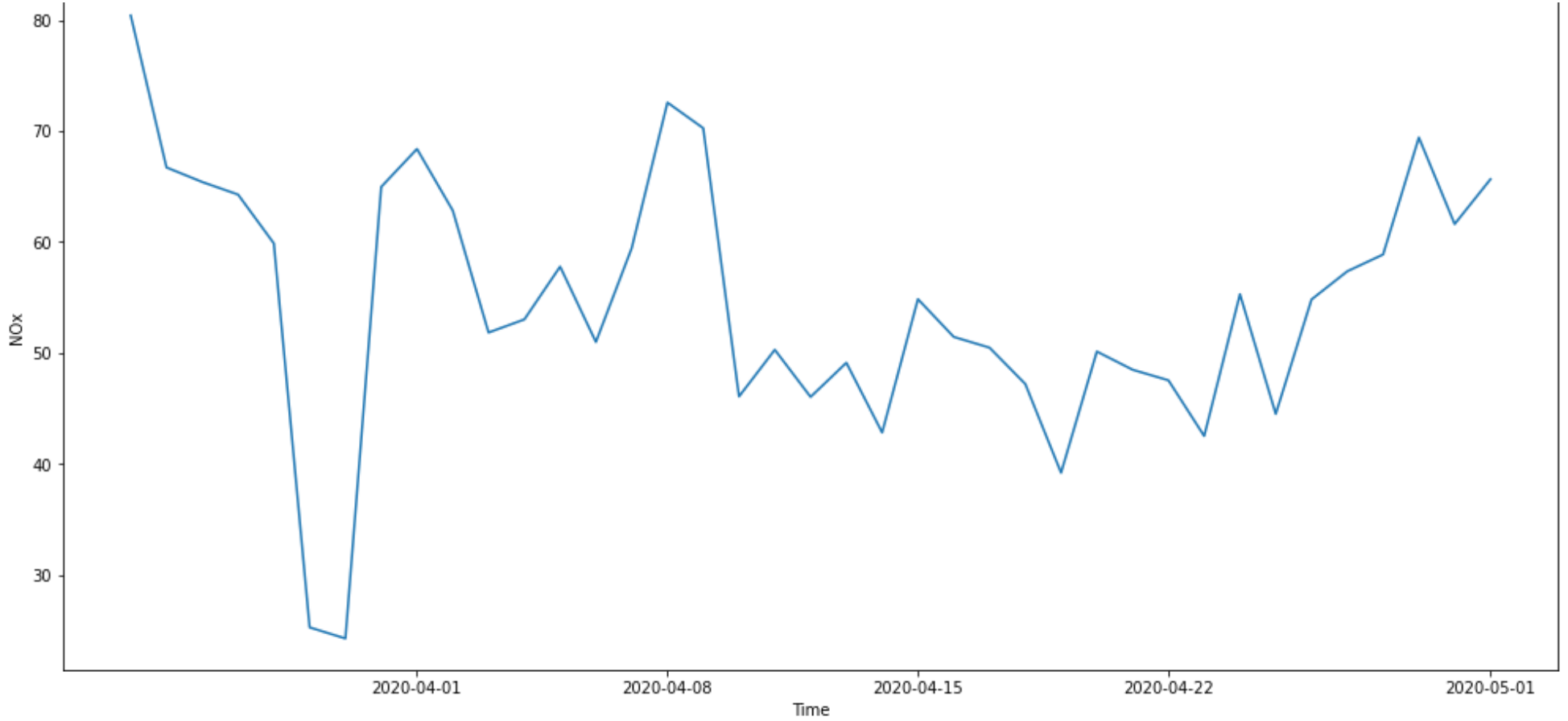
```
def plot_data(col):
    plt.figure(figsize=(17, 8))
    plt.plot(df_patna_after_covid[col])
    plt.xlabel('Time')
    plt.ylabel(col)
    plt.grid(False)
    plt.show()

for col in df_patna_after_covid.columns:
    plot_data(col)
```

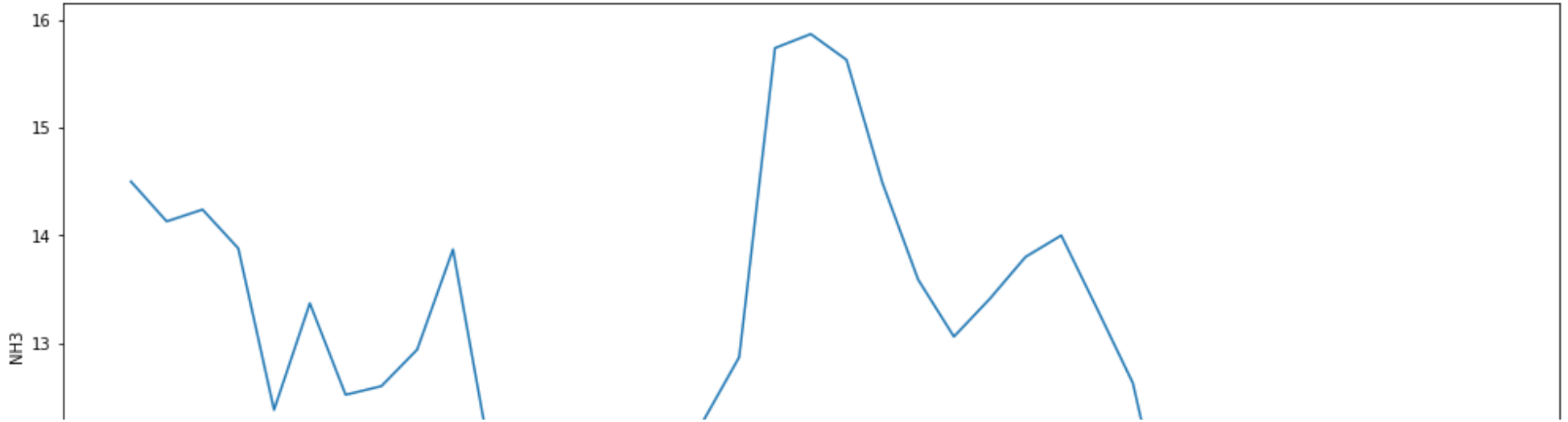

PM2.5
70
60
50
40
30
20
10
2020-04-01
2020-04-08
2020-04-15
2020-04-22
2020-05-01
Time

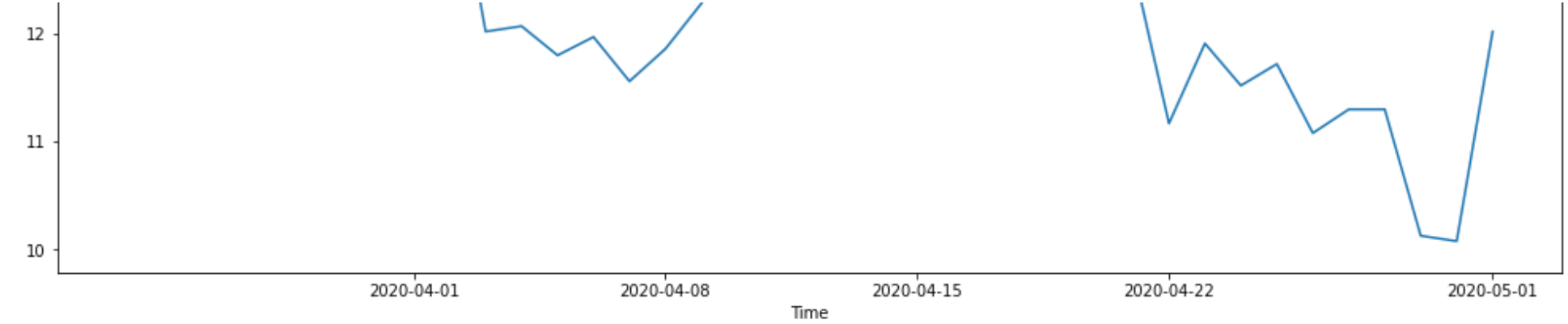

NO
80
70
60
50
40

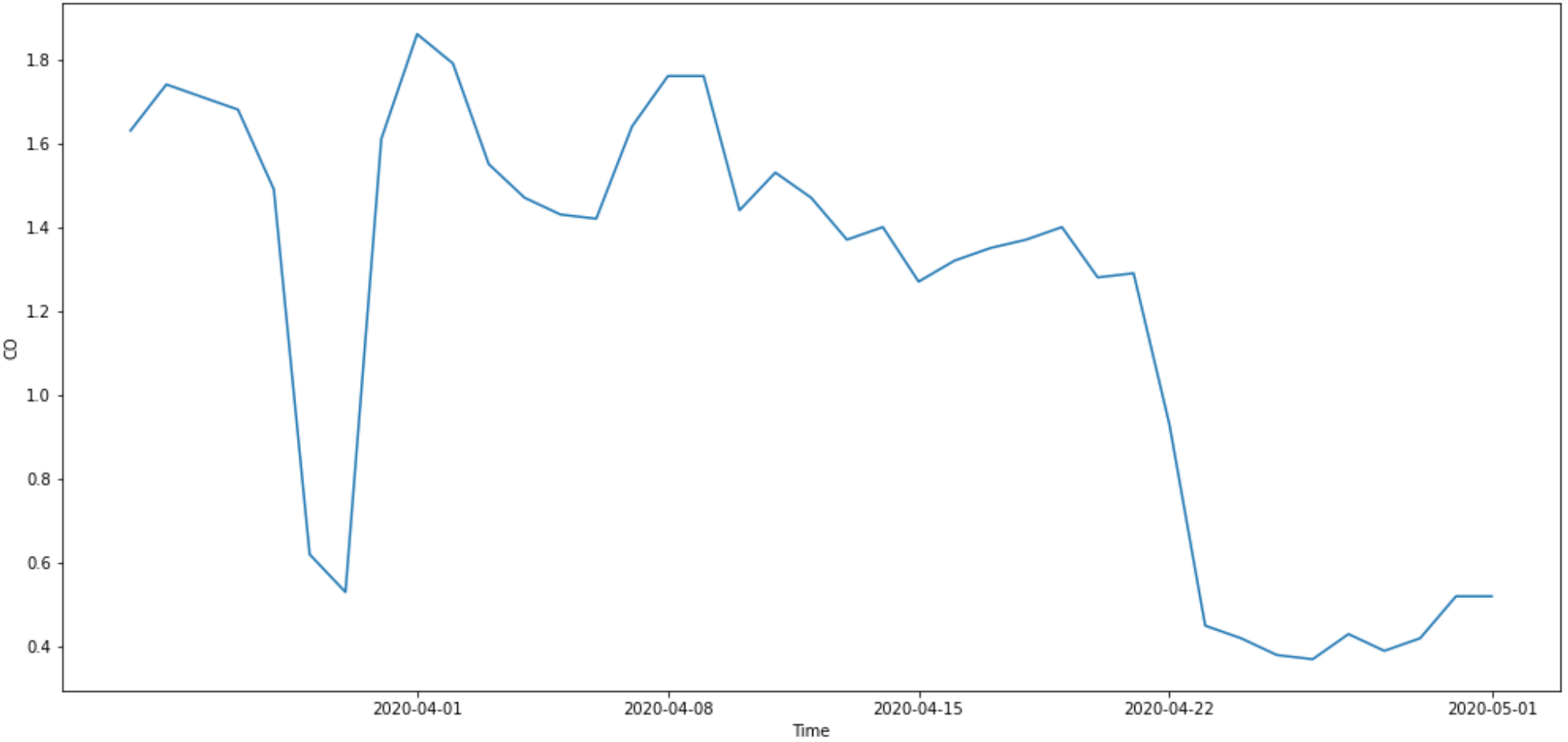

30
20
10
2020-04-01
2020-04-08
2020-04-15
2020-04-22
2020-05-01
Time

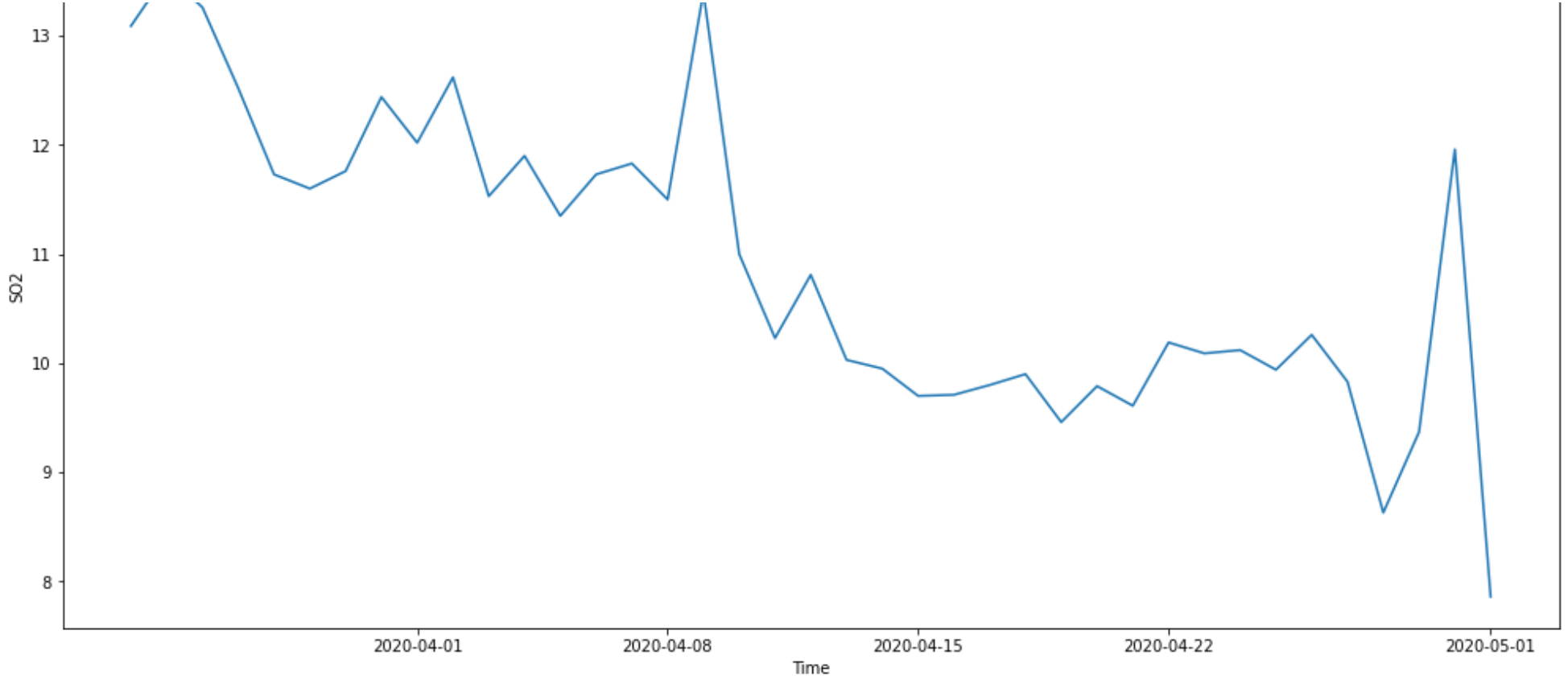

NO2
50
40
30
20
10
2020-04-01
2020-04-08
2020-04-15
2020-04-22
2020-05-01
Time

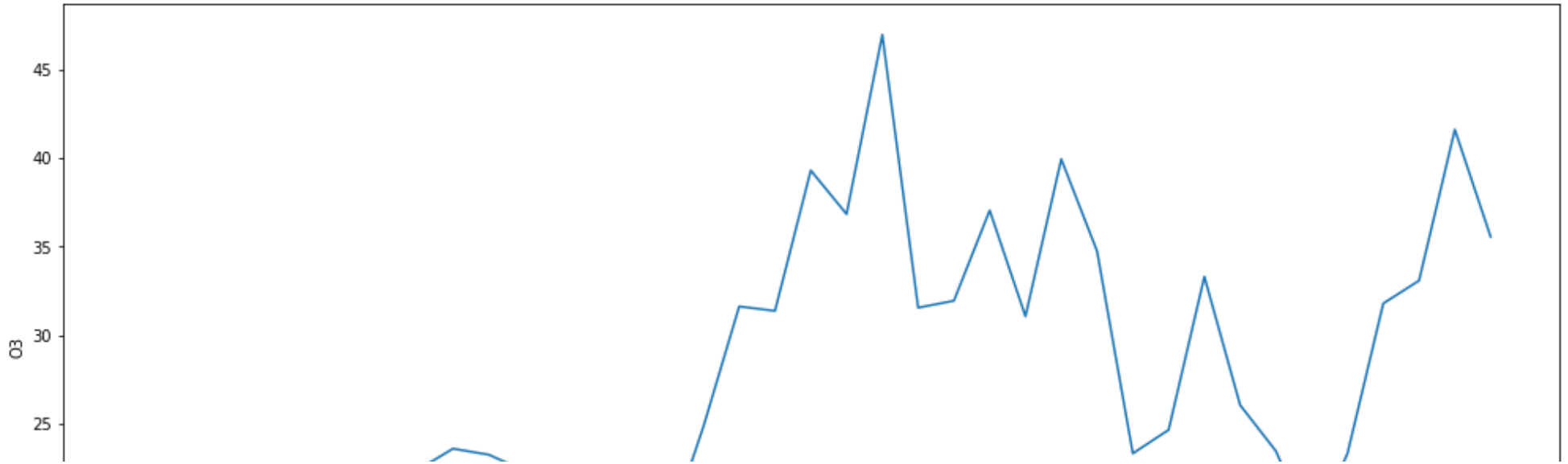

80
70
60
50
40
30
NOx
2020-04-01
2020-04-08
2020-04-15
2020-04-22
2020-05-01
Time

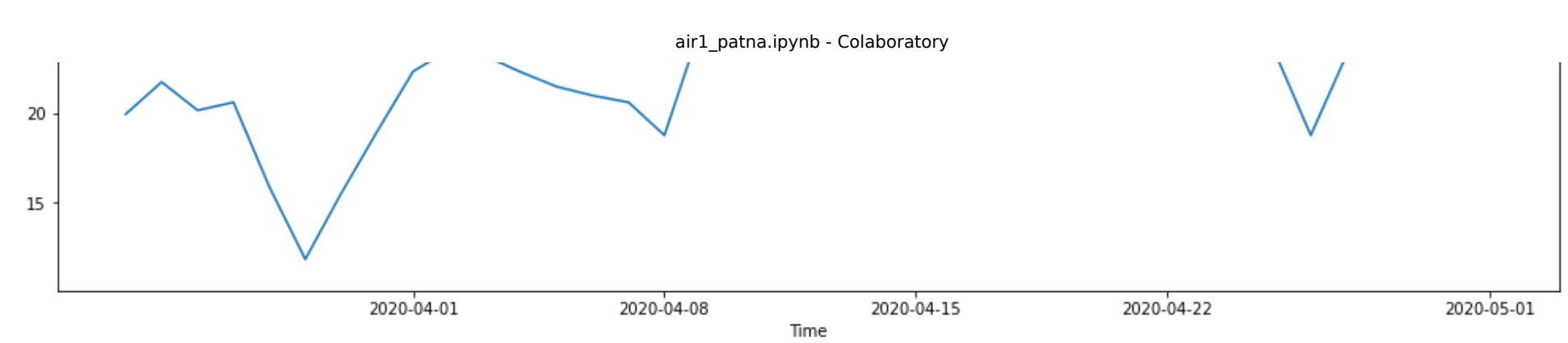

16
15
14
13
NH3

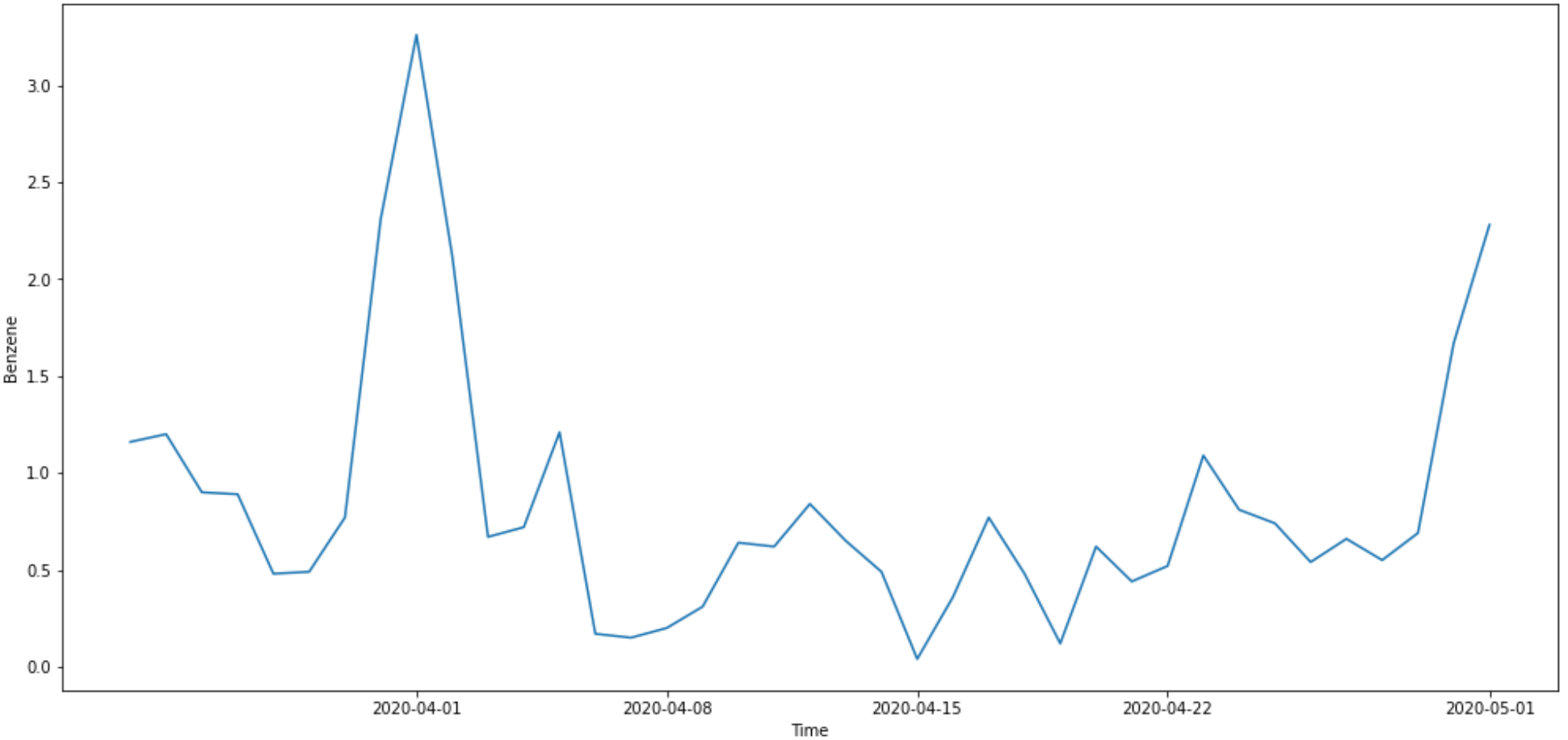

12
11
10
2020-04-01
2020-04-08
2020-04-15
2020-04-22
2020-05-01
Time

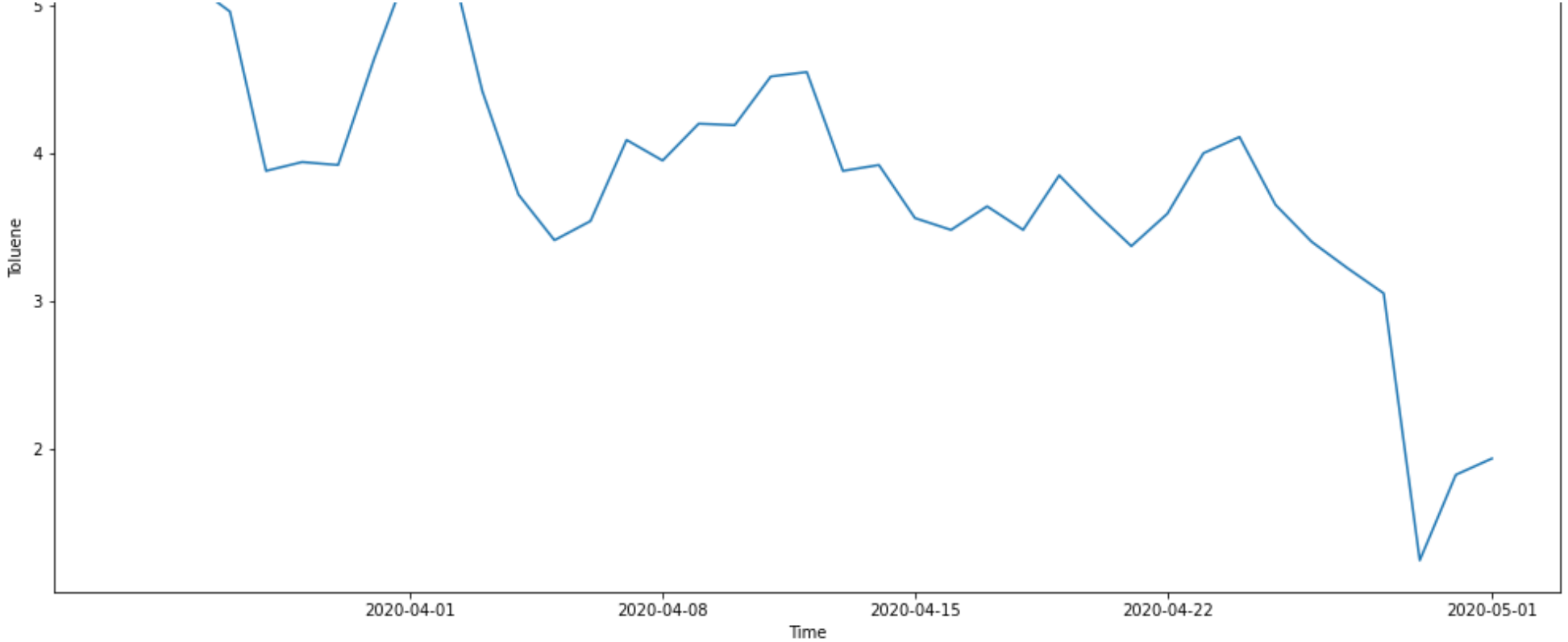

CO
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
2020-04-01
2020-04-08
2020-04-15
2020-04-22
2020-05-01
Time

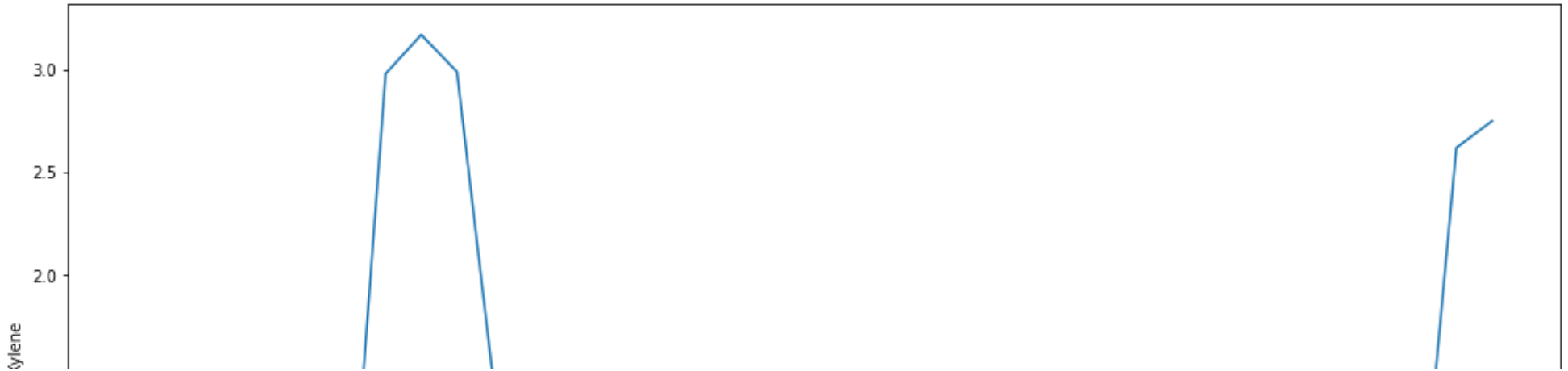

SO2
13
12
11
10
9
8
2020-04-01
2020-04-08
2020-04-15
2020-04-22
2020-05-01
Time


O3
45
40
35
30
25


20
15
2020-04-01
2020-04-08
2020-04-15
2020-04-22
2020-05-01
Time


Benzene
3.0
2.5
2.0
1.5
1.0
0.5
0.0
2020-04-01
2020-04-08
2020-04-15
2020-04-22
2020-05-01
Time






```
df_patna_after_covid = df_patna_after_covid.reset_index()
df_patna_after_covid = df_patna_after_covid.rename({'Date':'ds','AQI':'y'} , axis = 1)
df_patna_after_covid.head(1)
```

	ds	PM2.5	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	y
0	2020-03-24	65.94	69.45	48.75	80.41	14.5	1.63	13.09	19.97	1.16	5.41	0.35	177.0

```
tr  = df_patna_after_covid

cols = tr.columns.tolist()
```

```
cols = ['ds',
 'y',
 'PM2.5',
 'NO',
 'NO2',
 'NOx',
 'NH3',
 'CO',
 'SO2',
 'O3',
 'Benzene',
 'Toluene',
 'Xylene'
 ]
```

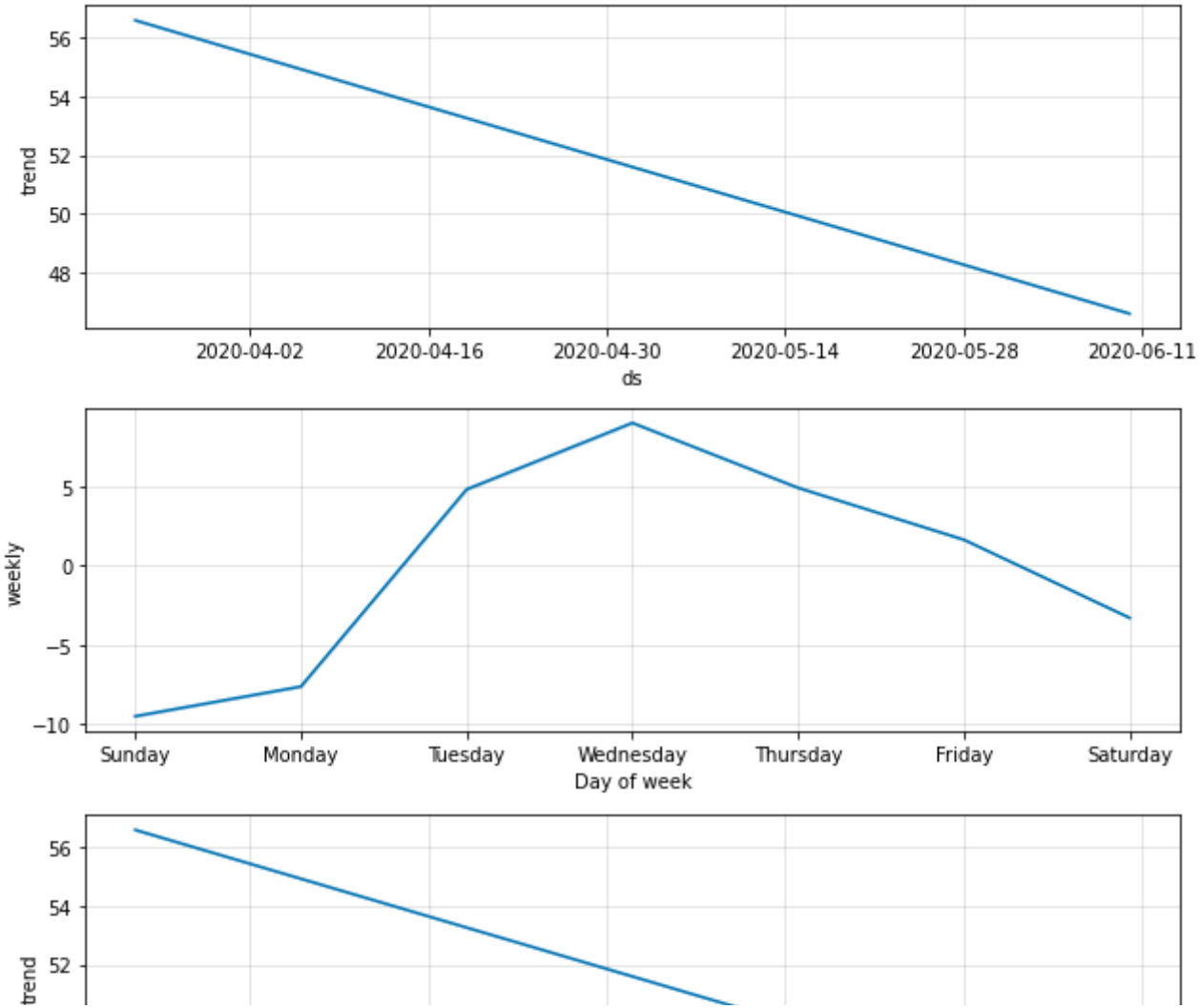
```
tr_nox = tr[['ds','NOx']]
tr_nox = tr_nox.rename({'NOx':'y'},axis=1)
f1 = Prophet()
f1.fit(tr_nox)
```

```
INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True to override this.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
<fbprophet.forecaster.Prophet at 0x7f55789e0f60>
```

```
future1 = f1.make_future_dataframe(periods= 40 )
forecast1 = f1.predict(future1)
forecast1.tail()
```

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_terms	additive_terms_lower	additive_
74	2020-06-06	47.110594	31.296973	55.627514	47.110509	47.110676	-3.290096	-3.290096	
75	2020-06-07	46.982592	25.378563	50.030132	46.982505	46.982680	-9.516346	-9.516346	
76	2020-06-08	46.854591	27.803311	50.859476	46.854500	46.854682	-7.638742	-7.638742	
77	2020-06-09	46.726590	39.284571	63.364262	46.726495	46.726684	4.830266	4.830266	
78	2020-06-10	46.598589	42.967694	67.299981	46.598489	46.598686	9.034555	9.034555	

```
1 f1.plot_components(forecast1)
```




```
1 def make_comparison_dataframe(historical, forecast):
2     return forecast.set_index('ds')[['yhat', 'yhat_lower', 'yhat_upper']].join(historical.set_index
3 cmp_df1 = make_comparison_dataframe(tr_nox, forecast1)
4
5 cmp_df1.head()
```

ds	yhat	yhat_lower	yhat_upper	y
2020-03-24	61.413260	48.259170	74.313096	80.41
2020-03-25	65.489536	53.372187	78.117013	66.72
2020-03-26	61.257499	48.304622	73.383845	65.41
2020-03-27	57.848789	45.068107	70.546612	64.28

```
plt.figure(figsize=(17, 8))
plt.plot(cmp_df1['yhat'])

plt.plot(cmp_df1['y'])
plt.xlabel('Time')
plt.ylabel('Average  NOx Concentration')
plt.grid(False)
plt.show()
```


```
1
```

Now will do analysis for SO2 for before and after covid

```
1 # before first
```



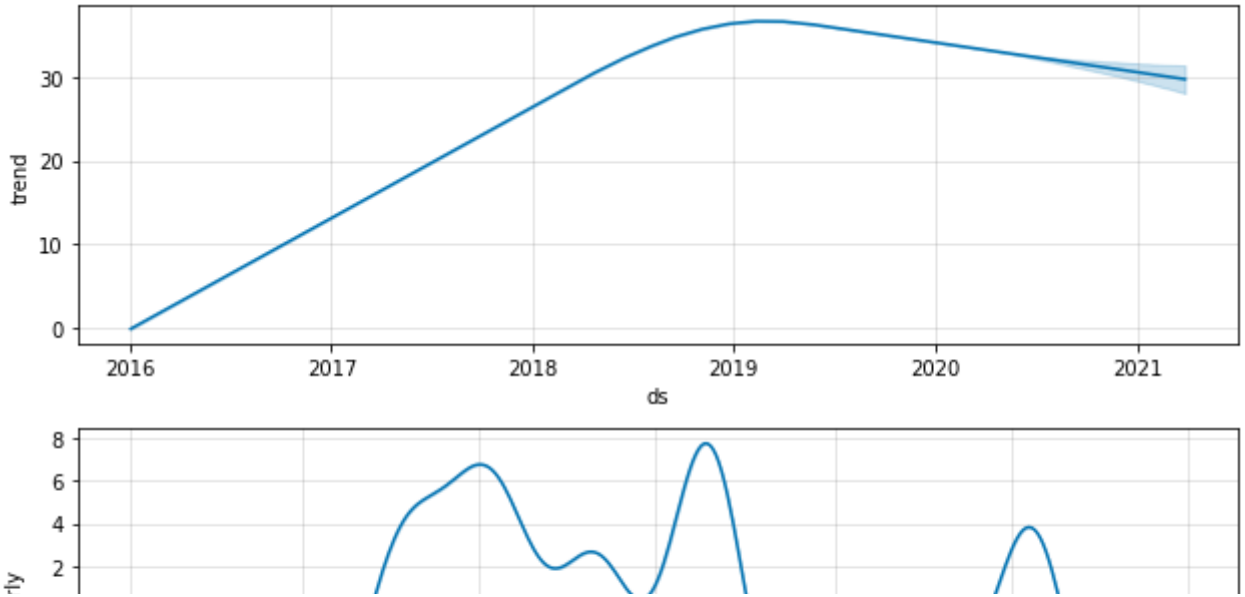
```
1 tr1_so2 = train[['ds','SO2']]
2 tr1_so2 = tr1_so2.rename({'SO2':'y'},axis=1)
3 s1 = Prophet()
4 s1.fit(tr1_so2)
```

```
INFO:fbprophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to override this.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
<fbprophet.forecaster.Prophet at 0x7f5578962c88>
```

```
1 future1_so2 = s1.make_future_dataframe(periods= 365 )
2 forecast1_so2 = s1.predict(future1_so2)
3 forecast1_so2.tail()
```

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_terms	additive_terms_lower	additive_
582	2021-03-25	29.805990	15.682218	43.640332	28.159649	31.433334	-0.015527	-0.015527	
	2021-								

```
1 s1.plot_components(forecast1_so2)
```

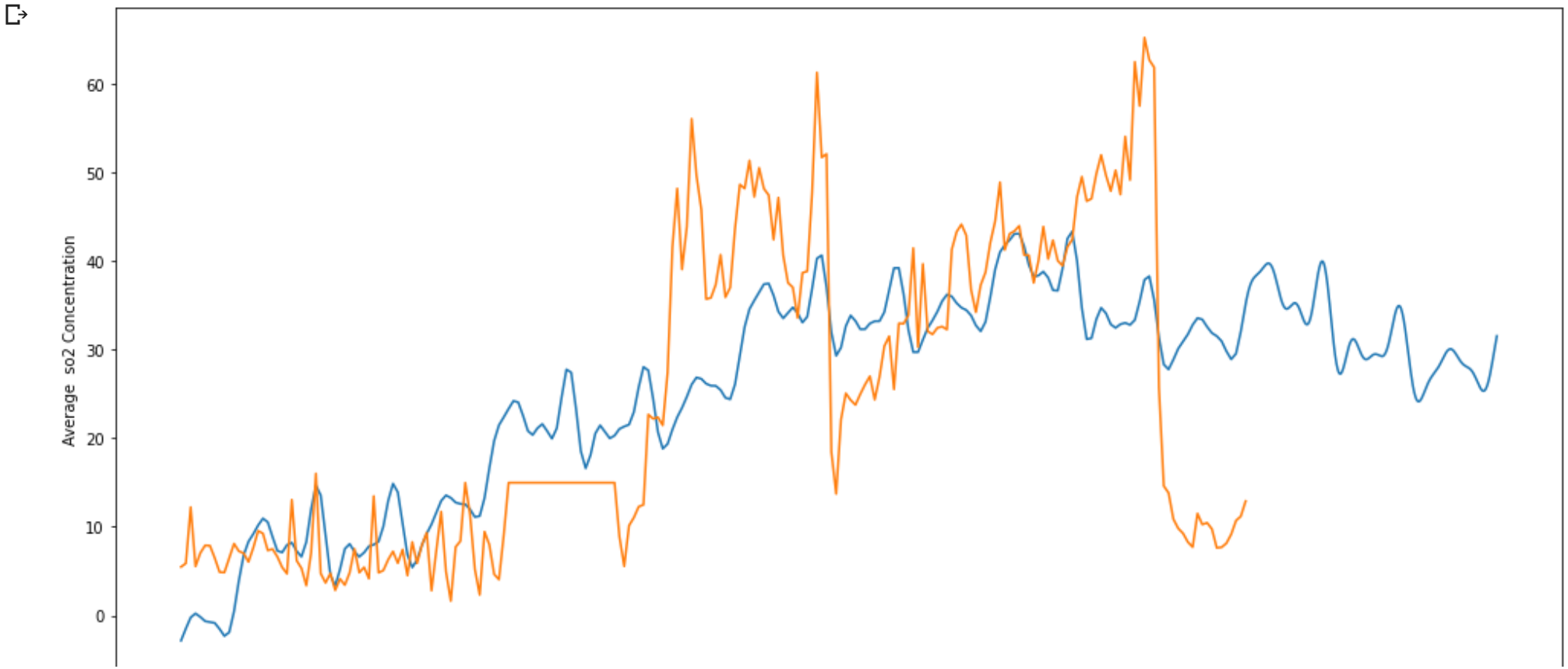



```
1 def make_comparison_dataframe(historical, forecast):
2     return forecast.set_index('ds')[['yhat', 'yhat_lower', 'yhat_upper']].join(historical.set_index
3 cmp_df1_so2 = make_comparison_dataframe(tr1_so2, forecast1_so2)
4
5 cmp_df1_so2.head()
```

ds	yhat	yhat_lower	yhat_upper	y
2016-01-03	-2.842132	-16.849085	11.301340	5.480000
2016-01-10	-1.488239	-15.942109	12.573838	5.882857
2016-01-17	-0.228333	-15.133821	13.517422	12.228571
2016-01-24	0.217880	-13.311933	14.573593	5.534286
2016-01-31	-0.173066	-14.623834	14.319388	7.044286

```
1 plt.figure(figsize=(17, 8))
2 plt.plot(cmp_df1_so2['yhat'])
```

```
3
4 plt.plot(cmp_df1_so2['y'])
5 plt.xlabel('Time')
6 plt.ylabel('Average  so2 Concentration')
7 plt.grid(False)
8 plt.show()
```




```
1 # after covid
```

```
1 tr2_so2 = tr[['ds','SO2']]
2 tr2_so2 = tr_nox.rename({'SO2':'y'},axis=1)
3 s2 = Prophet()
4 s2.fit(tr2_so2)
```

```
INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True to override this.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
<fbprophet.forecaster.Prophet at 0x7f5578a33a58>
```

```
future2_so2 = f1.make_future_dataframe(periods= 40 )
forecast2_so2 = f1.predict(future2_so2)
forecast2_so2.tail()
```

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_terms	additive_terms_lower	additive_
74	2020-06-06	47.110594	31.487536	55.780998	47.110509	47.110676	-3.290096	-3.290096	
75	2020-06-07	46.982592	25.469050	49.860974	46.982505	46.982677	-9.516346	-9.516346	
76	2020-06-08	46.854591	27.332315	50.913562	46.854500	46.854679	-7.638742	-7.638742	
77	2020-06-09	46.726590	39.189479	63.379992	46.726496	46.726681	4.830266	4.830266	
78	2020-06-10	46.598589	43.241578	67.979149	46.598491	46.598683	9.034555	9.034555	

```
s2.plot_components(forecast2_so2)
```