CO2,CH4

June 29, 2021

[101]:
```
import pandas
import numpy as np
import matplotlib.pyplot as plt
```

1 Texas

1.1 * The datasets are obtained from GOSAT; dates from 2017 to 2020(span of 3 years) *

[102]:
```
import os
os.listdir()    # Show me what is in the directory
```

[102]:
```
['FIS-data-access.ipynb',
 '.config',
 '.s3',
 '.cache',
 'Untitled1.ipynb',
 'data',
 '.ipynb_checkpoints',
 'README.ipynb',
 '.jupyter',
 '.ipython',
 'processing-API_ESA_JAXA_NASA_data-access.ipynb',
 'Untitled.ipynb',
 'OSM_API_access.ipynb',
 '.local',
 '.shared',
 'geodb_and_RACE.ipynb',
 'Excel Data',
 '.env',
 '.contribute-staging',
 'CO2,CH4.ipynb']
```

[103]:
```
data_blue = pandas.read_excel("Excel Data/Texas_CH4.xls")
```

[104]:
```
data_blue.head()
```

[104]:

	Dates	CH4_apr	CH4_tot	CH4_low	CH4_upp
0	2017/05/01	1.8249	1.8589	1.9766	1.8659
1	2017/05/07	1.8246	1.8463	1.8937	1.9007
2	2017/05/13	1.8232	1.8491	1.9461	1.8643
3	2017/05/25	1.8198	1.8623	1.9814	1.8645
4	2017/06/12	1.8154	1.7897	1.7776	1.8635

[105]:

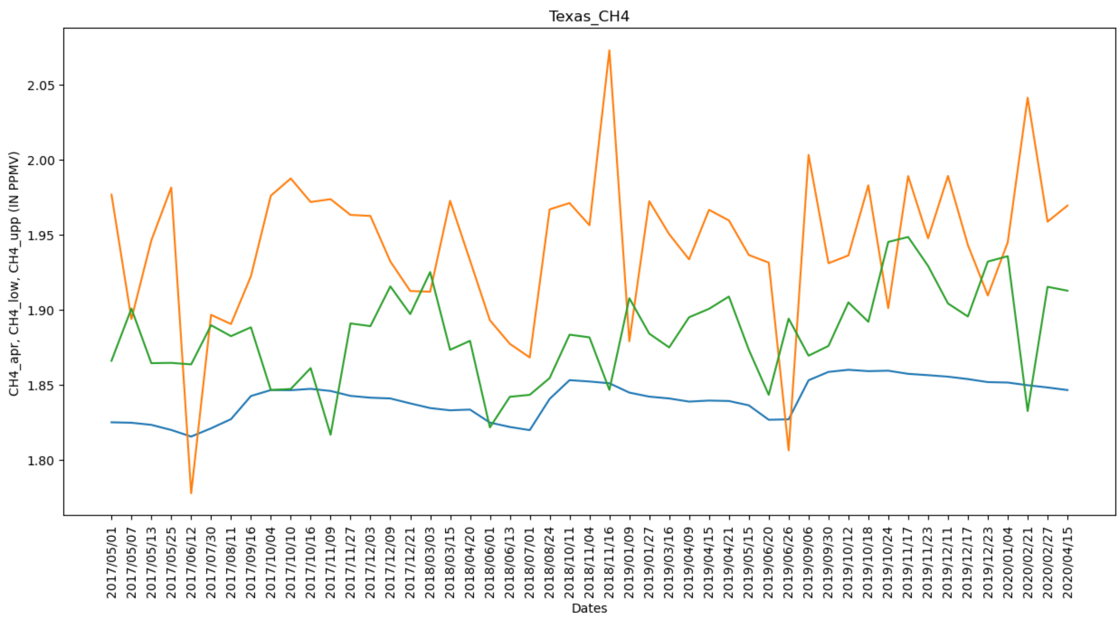
```
print(data_blue['Dates'])
```

```
0     2017/05/01
1     2017/05/07
2     2017/05/13
3     2017/05/25
4     2017/06/12
5     2017/07/30
6     2017/08/11
7     2017/09/16
8     2017/10/04
9     2017/10/10
10    2017/10/16
11    2017/11/09
12    2017/11/27
13    2017/12/03
14    2017/12/09
15    2017/12/21
16    2018/03/03
17    2018/03/15
18    2018/04/20
19    2018/06/01
20    2018/06/13
21    2018/07/01
22    2018/08/24
23    2018/10/11
24    2018/11/04
25    2018/11/16
26    2019/01/09
27    2019/01/27
28    2019/03/16
29    2019/04/09
30    2019/04/15
31    2019/04/21
32    2019/05/15
33    2019/06/20
34    2019/06/26
35    2019/09/06
36    2019/09/30
37    2019/10/12
```

```
38     2019/10/18
39     2019/10/24
40     2019/11/17
41     2019/11/23
42     2019/12/11
43     2019/12/17
44     2019/12/23
45     2020/01/04
46     2020/02/21
47     2020/02/27
48     2020/04/15
Name: Dates, dtype: object
```

[106]:

```python
plt.figure(figsize=(15,7))    # Set the size of your plot. It will determine the␣
↪relative size of all the labels.
plt.xticks(rotation=90)
plt.plot(data_blue['Dates'],data_blue['CH4_apr'],label="Blue Led")
plt.plot(data_blue['Dates'],data_blue['CH4_low'],label="Red")
plt.plot(data_blue['Dates'],data_blue['CH4_upp'],label="Green")# Plot a curve.
plt.xlabel("Dates")
plt.ylabel("CH4_apr, CH4_low, CH4_upp (IN PPMV)")
plt.title("Texas_CH4")
plt.show()
```



1.1.1 The blue line represent the priori of the column averaged dry air mole fraction of CH4 over the period of 3 years

1.1.2 Red line shows the CH4 levels in lower troposphere

1.1.3 Green line shows the CH4 levels in upper troposphere

[107]:
```
os.listdir()
```

[107]:
```
['FIS-data-access.ipynb',
 '.config',
 '.s3',
 '.cache',
 'Untitled1.ipynb',
 'data',
 '.ipynb_checkpoints',
 'README.ipynb',
 '.jupyter',
 '.ipython',
 'processing-API_ESA_JAXA_NASA_data-access.ipynb',
 'Untitled.ipynb',
 'OSM_API_access.ipynb',
 '.local',
 '.shared',
 'geodb_and_RACE.ipynb',
 'Excel Data',
 '.env',
 '.contribute-staging',
 'CO2,CH4.ipynb']
```

[108]:
```
data_blue = pandas.read_excel("Excel Data/Texas_CO2.xls")
```

[109]:
```
data_blue.head()
```

[109]:

	Dates	CO2_apr	CO2_tot	CO2_low	CO2_upp
0	2017/05/01	407.3179	405.7720	407.4197	406.9988
1	2017/05/07	407.4105	403.8550	404.4660	404.8063
2	2017/05/13	407.3584	405.9935	407.1953	407.3828
3	2017/05/25	407.1538	406.0407	408.1818	406.0533
4	2017/06/12	406.8214	392.2171	381.1371	397.9275

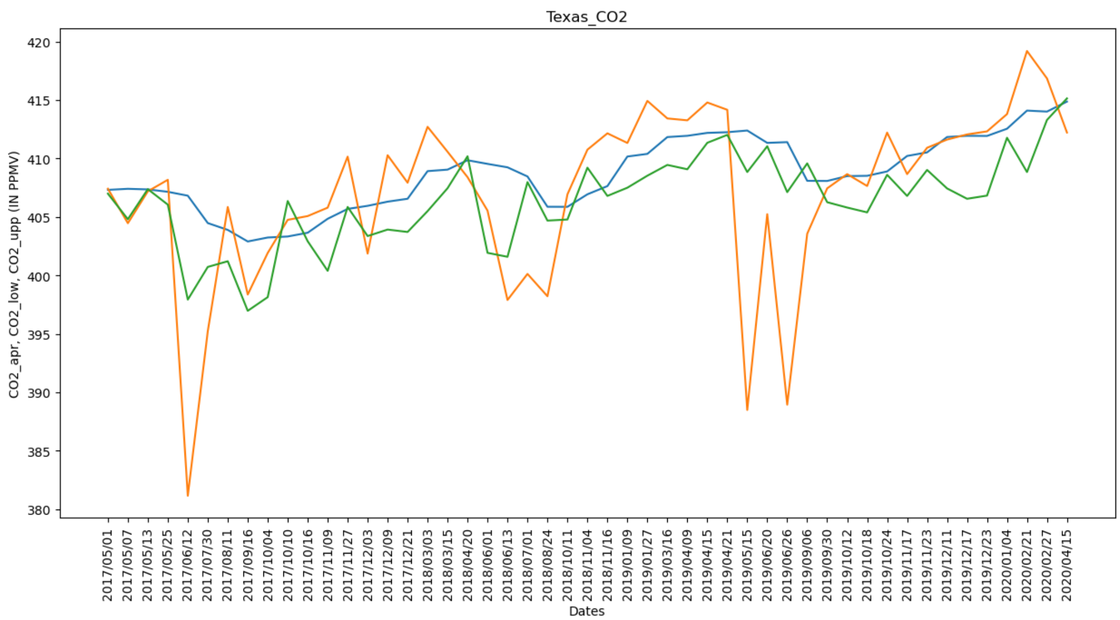
[110]:
```
print(data_blue['Dates'])
```

```
0      2017/05/01
1      2017/05/07
2      2017/05/13
3      2017/05/25
4      2017/06/12
5      2017/07/30
```

```
6      2017/08/11
7      2017/09/16
8      2017/10/04
9      2017/10/10
10     2017/10/16
11     2017/11/09
12     2017/11/27
13     2017/12/03
14     2017/12/09
15     2017/12/21
16     2018/03/03
17     2018/03/15
18     2018/04/20
19     2018/06/01
20     2018/06/13
21     2018/07/01
22     2018/08/24
23     2018/10/11
24     2018/11/04
25     2018/11/16
26     2019/01/09
27     2019/01/27
28     2019/03/16
29     2019/04/09
30     2019/04/15
31     2019/04/21
32     2019/05/15
33     2019/06/20
34     2019/06/26
35     2019/09/06
36     2019/09/30
37     2019/10/12
38     2019/10/18
39     2019/10/24
40     2019/11/17
41     2019/11/23
42     2019/12/11
43     2019/12/17
44     2019/12/23
45     2020/01/04
46     2020/02/21
47     2020/02/27
48     2020/04/15
Name: Dates, dtype: object
```

[111]:
```
plt.figure(figsize=(15,7))    # Set the size of your plot. It will determine the␣
 ↪relative size of all the labels.
```

```python
plt.xticks(rotation=90)
plt.plot(data_blue['Dates'],data_blue['CO2_apr'],label="Blue Led")
plt.plot(data_blue['Dates'],data_blue['CO2_low'],label="Red")
plt.plot(data_blue['Dates'],data_blue['CO2_upp'],label="Green")# Plot a curve.
plt.xlabel("Dates")
plt.ylabel("CO2_apr, CO2_low, CO2_upp (IN PPMV)")
plt.title("Texas_CO2")
plt.show()
```



1.1.4 We obtained the datasets from GOSAT, for different regions. By monitoring the levels of CO2 and CH4 through these datasets we would probably be able to find out the regions under threat(the ones contributing to the global warming the highest) in the near future(5 years or more from now)from our mathematical analysis and would be marking those areas as red zones. Moreover we would also be able to monitor climate change, temperature fluctuations, affects on crop through datasets obtained from Sentinel-2 and GOSAT.

[ ]:

[112]:
```python
data_blue = pandas.read_excel("Excel Data/Texas_comb.xls")
```

[113]:
```python
data_blue.head()
```

[113]:
```
        Dates   CO2_apr   CO2_tot   CO2_low   CO2_upp  CH4_apr  CH4_tot  \
0  2017/05/01  407.3179  405.7720  407.4197  406.9988   1.8249   1.8589
1  2017/05/07  407.4105  403.8550  404.4660  404.8063   1.8246   1.8463
```

```
2  2017/05/13  407.3584  405.9935  407.1953  407.3828   1.8232   1.8491
3  2017/05/25  407.1538  406.0407  408.1818  406.0533   1.8198   1.8623
4  2017/06/12  406.8214  392.2171  381.1371  397.9275   1.8154   1.7897

   CH4_low  CH4_upp
0   1.9766   1.8659
1   1.8937   1.9007
2   1.9461   1.8643
3   1.9814   1.8645
4   1.7776   1.8635
```

2 Delhi

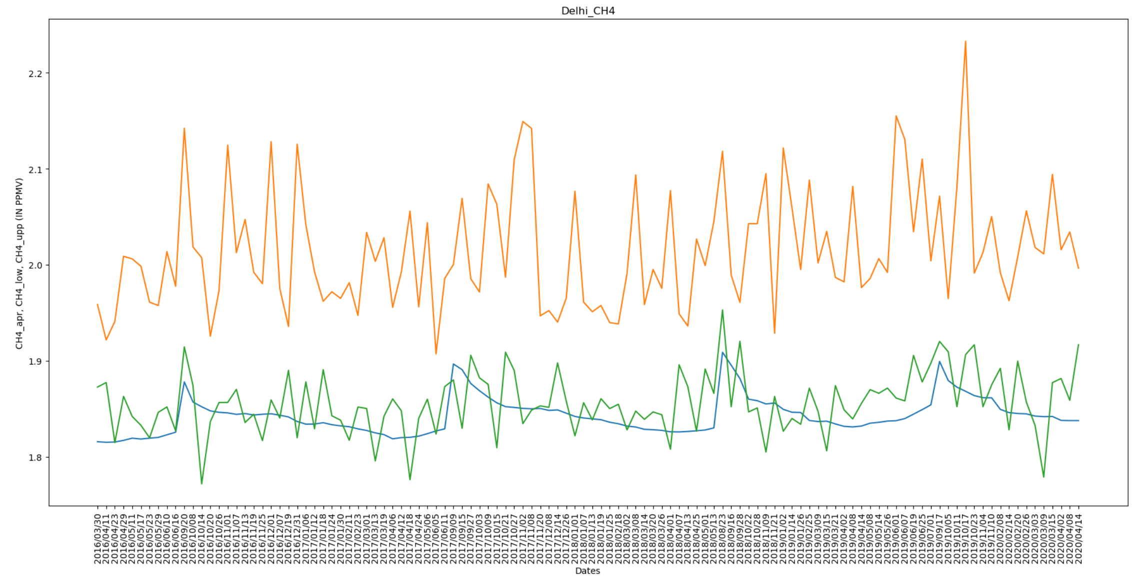
2.1 Methane Statistics

[114]:
```python
data_blue = pandas.read_excel("Excel Data/Delhi_CH4.xls")
```

[115]:
```python
data_blue.head()
```

[115]:
```
        Dates  CH4_apr  CH4_tot  CH4_low  CH4_upp
0  2016/03/30   1.8159   1.8483   1.9587   1.8728
1  2016/04/11   1.8153   1.8486   1.9218   1.8775
2  2016/04/23   1.8156   1.8366   1.9411   1.8150
3  2016/04/29   1.8173   1.8758   2.0089   1.8631
4  2016/05/11   1.8196   1.8658   2.0062   1.8422
```

[116]:
```python
plt.figure(figsize=(22,10))   # Set the size of your plot. It will determine␣
 ↪the relative size of all the labels.
plt.xticks(rotation=90)
plt.plot(data_blue['Dates'],data_blue['CH4_apr'],label="Blue Led")
plt.plot(data_blue['Dates'],data_blue['CH4_low'],label="Red")
plt.plot(data_blue['Dates'],data_blue['CH4_upp'],label="Green")# Plot a curve.
plt.xlabel("Dates")
plt.ylabel("CH4_apr, CH4_low, CH4_upp (IN PPMV)")
plt.title("Delhi_CH4")
plt.show()
```



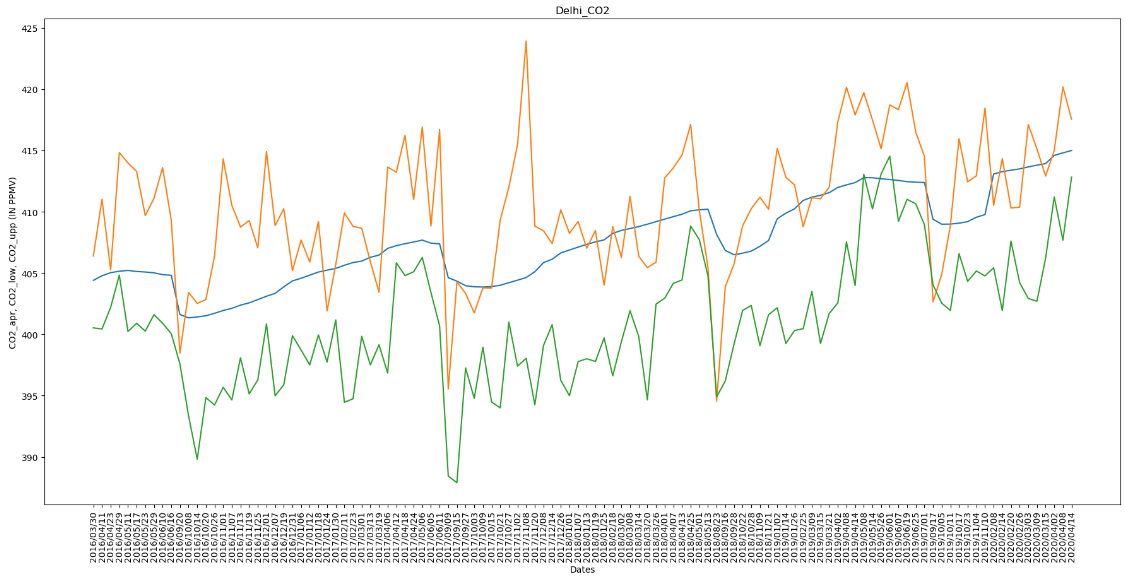
2.2 Carbon Dioxide Statistics

[117]:
```
data_blue = pandas.read_excel("Excel Data/Delhi_CO2.xls")
data_blue.head()
```

[117]:

	Dates	CO2_apr	CO2_tot	CO2_low	CO2_upp
0	2016/03/30	404.4056	402.0120	406.3860	400.5298
1	2016/04/11	404.7828	403.4066	411.0226	400.4420
2	2016/04/23	405.0342	402.2790	405.2618	402.2020
3	2016/04/29	405.1517	406.8357	414.8321	404.8370
4	2016/05/11	405.2183	404.6457	413.9846	400.2349

[118]:
```
plt.figure(figsize=(22,10))    # Set the size of your plot. It will determine␣
↪the relative size of all the labels.
plt.xticks(rotation=90)
plt.plot(data_blue['Dates'],data_blue['CO2_apr'],label="Blue Led")
plt.plot(data_blue['Dates'],data_blue['CO2_low'],label="Red")
plt.plot(data_blue['Dates'],data_blue['CO2_upp'],label="Green")# Plot a curve.
plt.xlabel("Dates")
plt.ylabel("CO2_apr, CO2_low, CO2_upp (IN PPMV)")
plt.title("Delhi_CO2")
plt.show()
```



2.2.1 From sources Delhi, the capital of India has got one of the highest carbon footprint in the country. From the datsets of NASA, ESA and JAXA we would possibly able to locate the regions in Delhi which contributes to the highest carbon level by monitoring the greenhouse gases, CO2, temperature through time series and hence produce the future vision of how possibly the city and its nearby places will be affected if the same mean rate continues.

3 Schipol Airport

3.1 Methane Statistics

[119]:
```
data_blue = pandas.read_excel("Excel Data/Schipol.xls")
```

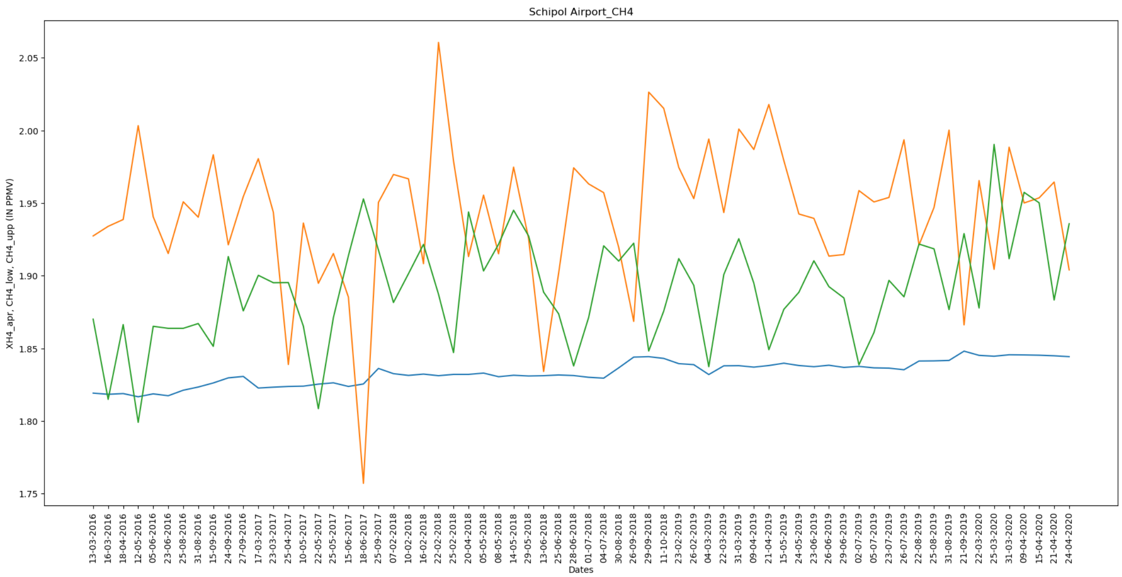
[120]:
```
data_blue.head()
```

[120]:

	Dates	XCO2_apr	XCO2_low	XCO2_upp	XCH4_apr	XCH4_low	XCH4_upp
0	13-03-2016	405.0702	409.0610	401.7155	1.8192	1.9274	1.8701
1	16-03-2016	405.0452	408.4106	397.0137	1.8184	1.9340	1.8149
2	18-04-2016	405.1181	405.0510	402.7833	1.8189	1.9387	1.8664
3	12-05-2016	405.0669	400.0309	392.3549	1.8167	2.0033	1.7991
4	05-06-2016	404.3826	401.8332	396.5163	1.8187	1.9406	1.8652

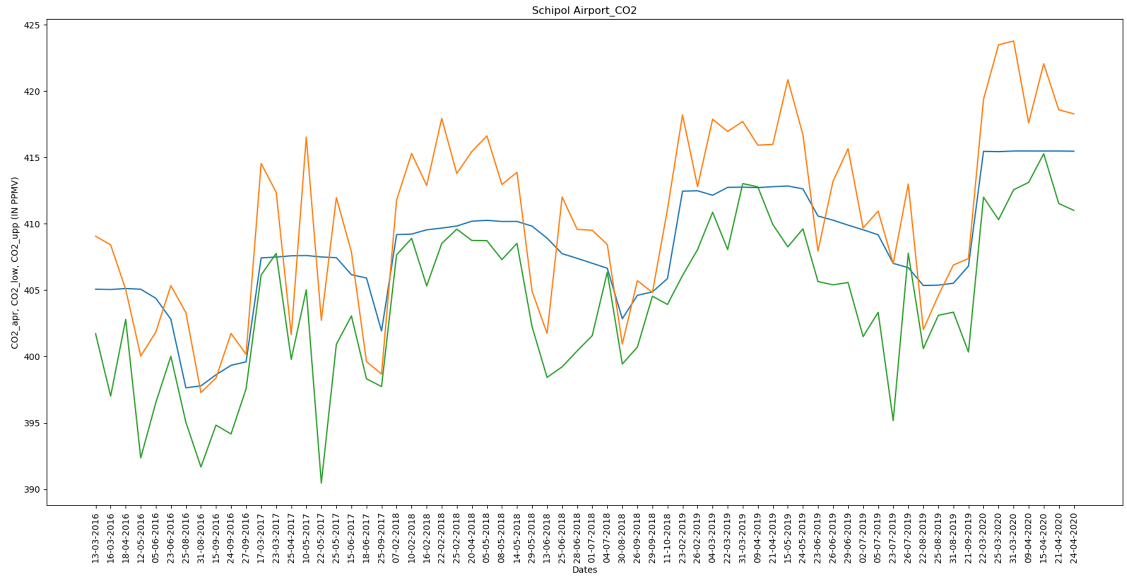
[121]:
```
plt.figure(figsize=(22,10))    # Set the size of your plot. It will determine␣
 ↪the relative size of all the labels.
plt.xticks(rotation=90)
plt.plot(data_blue['Dates'],data_blue['XCH4_apr'],label="Blue Led")
plt.plot(data_blue['Dates'],data_blue['XCH4_low'],label="Red")
plt.plot(data_blue['Dates'],data_blue['XCH4_upp'],label="Green")# Plot a curve.
```

```
plt.xlabel("Dates")
plt.ylabel("XH4_apr, CH4_low, CH4_upp (IN PPMV)")
plt.title("Schipol Airport_CH4")
plt.show()
```



3.2 CO2 Statistics

[122]:
```
plt.figure(figsize=(22,10))    # Set the size of your plot. It will determine␣
↪the relative size of all the labels.
plt.xticks(rotation=90)
plt.plot(data_blue['Dates'],data_blue['XCO2_apr'],label="Blue Led")
plt.plot(data_blue['Dates'],data_blue['XCO2_low'],label="Red")
plt.plot(data_blue['Dates'],data_blue['XCO2_upp'],label="Green")# Plot a curve.
plt.xlabel("Dates")
plt.ylabel("CO2_apr, CO2_low, CO2_upp (IN PPMV)")
plt.title("Schipol Airport_CO2")
plt.show()
```



[ ]:

4 Lahore

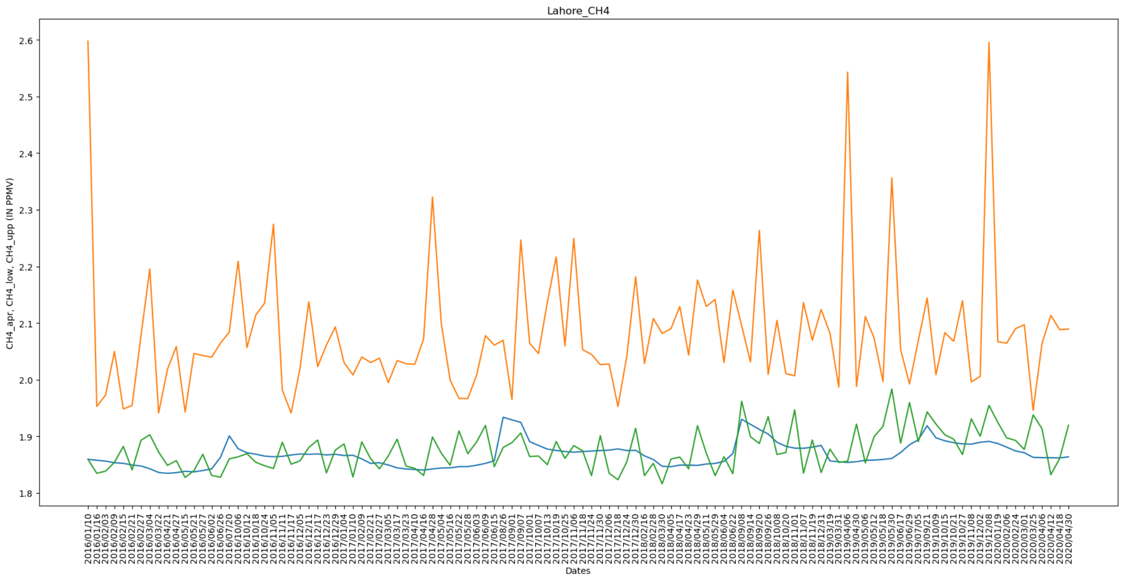
4.1 Methane Statistics

[123]:
```
data_blue = pandas.read_excel("Excel Data/Lahore.xls")
data_blue.head()
```

[123]:

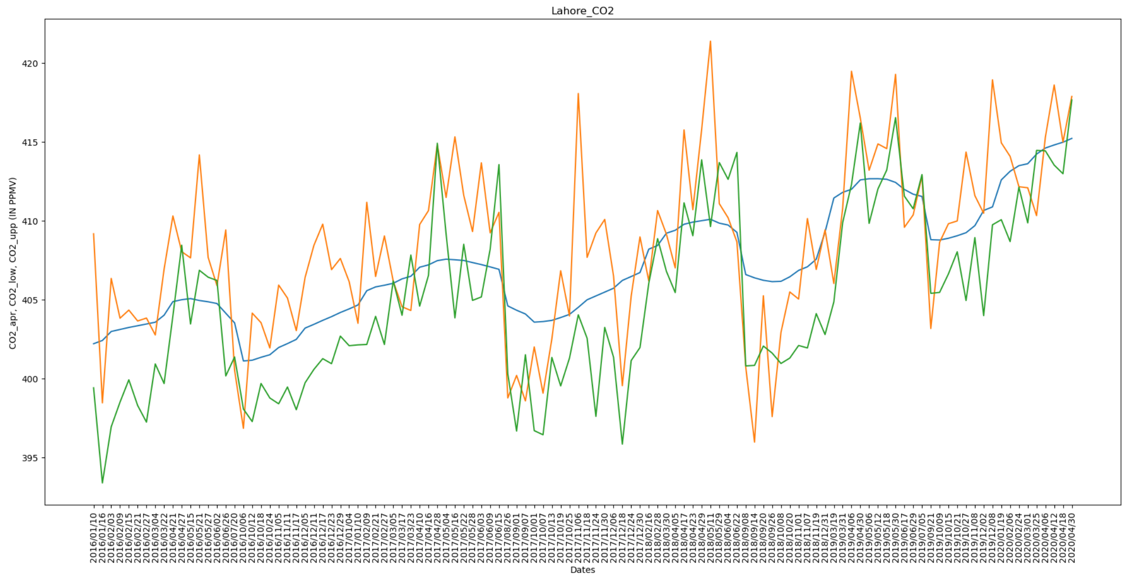
	Dates	XCO2_apr	XCO2_low	XCO2_upp	XCH4_apr	XCH4_low	XCH4_upp
0	2016/01/10	402.1956	409.1649	399.4023	1.8595	2.5974	1.8595
1	2016/01/16	402.4032	398.4462	393.3685	1.8578	1.9528	1.8347
2	2016/02/03	402.9724	406.3348	396.9299	1.8563	1.9731	1.8386
3	2016/02/09	403.0995	403.8143	398.4944	1.8534	2.0497	1.8540
4	2016/02/15	403.2271	404.3289	399.9094	1.8523	1.9483	1.8822

[124]:
```
plt.figure(figsize=(22,10))    # Set the size of your plot. It will determine␣
↪the relative size of all the labels.
plt.xticks(rotation=90)
plt.plot(data_blue['Dates'],data_blue['XCH4_apr'],label="Blue Led")
plt.plot(data_blue['Dates'],data_blue['XCH4_low'],label="Red")
plt.plot(data_blue['Dates'],data_blue['XCH4_upp'],label="Green")# Plot a curve.
plt.xlabel("Dates")
plt.ylabel("CH4_apr, CH4_low, CH4_upp (IN PPMV)")
plt.title("Lahore_CH4")
plt.show()
```



4.2 CO2 Statistics

[125]:
```
plt.figure(figsize=(22,10))    # Set the size of your plot. It will determine
 ↪the relative size of all the labels.
plt.xticks(rotation=90)
plt.plot(data_blue['Dates'],data_blue['XCO2_apr'],label="Blue Led")
plt.plot(data_blue['Dates'],data_blue['XCO2_low'],label="Red")
plt.plot(data_blue['Dates'],data_blue['XCO2_upp'],label="Green")# Plot a curve.
plt.xlabel("Dates")
plt.ylabel("CO2_apr, CO2_low, CO2_upp (IN PPMV)")
plt.title("Lahore_CO2")
plt.show()
```



[ ]:

```

```

5 San Francisco

[126]:

```
data_blue = pandas.read_excel("Excel Data/SanF.xls")
data_blue.head()
```

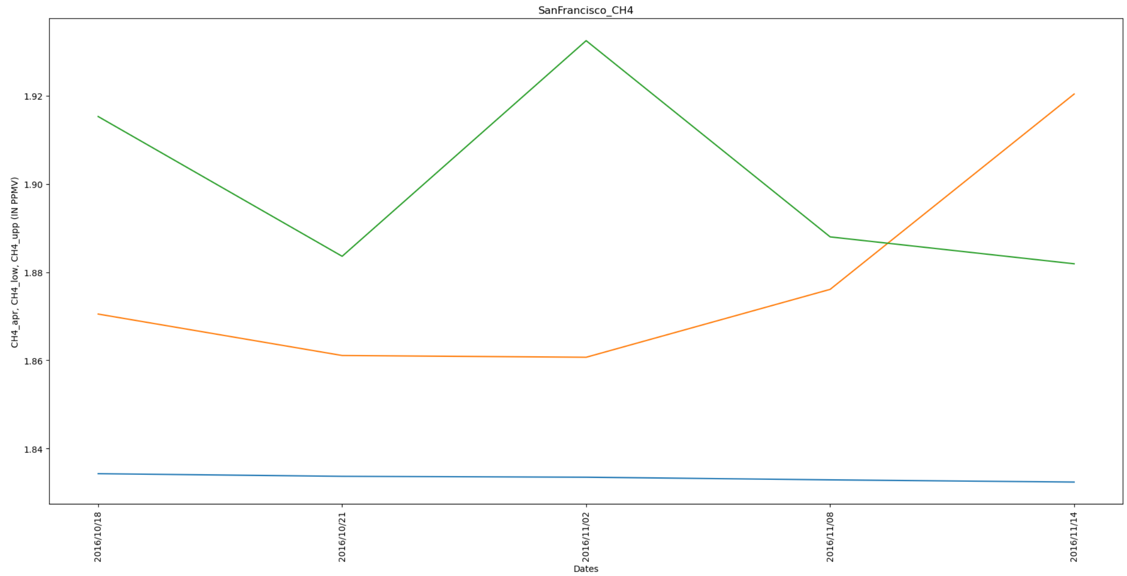
[126]:

	Dates	XCO2_apr	XCO2_low	XCO2_upp	XCH4_apr	XCH4_low	XCH4_upp
0	2016/10/18	401.7400	404.3901	399.5240	1.8343	1.8705	1.9153
1	2016/10/21	401.7837	402.3574	399.1347	1.8337	1.8611	1.8836
2	2016/11/02	402.5571	408.6408	400.1798	1.8335	1.8607	1.9325
3	2016/11/08	402.8543	406.8935	398.2899	1.8329	1.8761	1.8880
4	2016/11/14	403.1475	407.4052	396.8061	1.8324	1.9204	1.8819

5.1 Methane Statistics

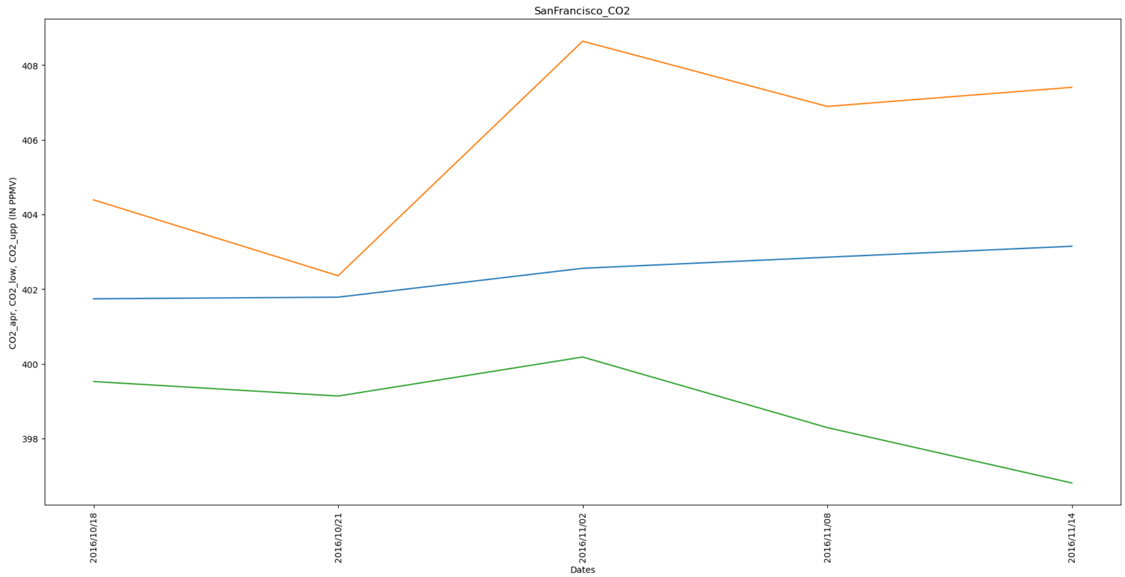
[127]:

```
plt.figure(figsize=(22,10))    # Set the size of your plot. It will determine␣
↪the relative size of all the labels.
plt.xticks(rotation=90)
plt.plot(data_blue['Dates'],data_blue['XCH4_apr'],label="Blue Led")
plt.plot(data_blue['Dates'],data_blue['XCH4_low'],label="Red")
plt.plot(data_blue['Dates'],data_blue['XCH4_upp'],label="Green")# Plot a curve.
plt.xlabel("Dates")
plt.ylabel("CH4_apr, CH4_low, CH4_upp (IN PPMV)")
plt.title("SanFrancisco_CH4")
plt.show()
```



5.2 CO2 Statistics

[128]:
```
plt.figure(figsize=(22,10))    # Set the size of your plot. It will determine␣
↪the relative size of all the labels.
plt.xticks(rotation=90)
plt.plot(data_blue['Dates'],data_blue['XCO2_apr'],label="Blue Led")
plt.plot(data_blue['Dates'],data_blue['XCO2_low'],label="Red")
plt.plot(data_blue['Dates'],data_blue['XCO2_upp'],label="Green")# Plot a curve.
plt.xlabel("Dates")
plt.ylabel("CO2_apr, CO2_low, CO2_upp (IN PPMV)")
plt.title("SanFrancisco_CO2")
plt.show()
```



5.2.1 In our project we have taken 4 megacities and 1 airport for our study. The case study done on San Fransisco display the levels of CO2 for the 2 layers of troposphere over a period of around a month.