Step 4: Select one sample per hour for the stats calculation done in step 3

Do notice that, usually the satellite offers a full scan of the continent in an interval of 15 minutes. But, the weather station collects data hourly. Therefore, it would be the best for us to select the sample according to the weather station data collection schedule.

In [1]:

```
import os
import pandas as pd
import scipy
import numpy as np
from tqdm import tqdm
```

TO-DO:

Change the directory for the input file if needed.

In [2]:

```
os.getcwd()

## TO-DO: Change the directory if needed
os.chdir("/srv/scratch/NOAA/GOES_Hourly_Statistics/stats_result/")
os.getcwd()
```

Out[2]:

```
'/srv/scratch/NOAA/GOES_Hourly_Statistics/stats_result'
```

In [3]:

```
lake = pd.read_csv('03_opencv_lat_lon_15f16s_lake_0.csv', dtype={'Date': str
lake
```

Out[3]:

	Date	Time	Mean	Centroid_lon	Centroid_lat	Std_lon	Std_lat	Skewnes
0	20151001	0000	0.001967	-86.745549	43.885599	0.646480	1.212252	0.64
1	20151001	0030	0.001977	-86.739336	43.893524	0.632930	1.222021	0.57
2	20151001	0100	0.001764	-86.749381	43.939543	0.628686	1.202172	0.61
3	20151001	0115	0.001848	-86.741998	43.834050	0.632403	1.238987	0.63
4	20151001	0130	0.002057	-86.736437	43.914970	0.643772	1.221739	0.68
...	...	...	...	...	...	...	...	
15682	20160331	2245	0.333690	-86.774138	43.987717	0.665368	1.211079	0.63
15683	20160331	2300	0.263741	-86.778112	43.942255	0.659147	1.215728	0.63
15684	20160331	2315	0.208805	-86.795057	43.866143	0.657376	1.225620	0.69

	Date	Time	Mean	Centroid_lon	Centroid_lat	Std_lon	Std_lat	Skewnes
15685	20160331	2330	0.141006	-86.806141	43.834040	0.650572	1.209149	0.74
15686	20160331	2345	0.086758	-86.874062	43.660057	0.613668	1.186940	0.87

15687 rows × 17 columns

In [4]:

```
# Add a ' to reduce error in pandas reading the Time elements with leading 0

lake['Time'] = lake['Time'].apply(lambda x:"'"+x)
```

In [5]:

```
h_list = []
m_list = []
for t in lake['Time']:
    h_list.append(t[1:3])
    m_list.append(t[-2:])
lake['Hour'] = h_list
lake['Min'] = m_list
lake
```

```
Out[5]:
            Date   Time      Mean  Centroid_lon  Centroid_lat   Std_lon   Std_lat  Skewnes

    0   20151001  '0000  0.001967    -86.745549     43.885599  0.646480  1.212252      0.6

    1   20151001  '0030  0.001977    -86.739336     43.893524  0.632930  1.222021      0.5

    2   20151001  '0100  0.001764    -86.749381     43.939543  0.628686  1.202172      0.6

    3   20151001   '0115  0.001848    -86.741998     43.834050  0.632403  1.238987      0.6

    4   20151001  '0130  0.002057    -86.736437     43.914970  0.643772  1.221739      0.6

  ...        ...    ...       ...           ...           ...       ...       ...

15682   20160331  '2245  0.333690    -86.774138     43.987717  0.665368  1.211079      0.6

15683   20160331  '2300  0.263741    -86.778112     43.942255  0.659147  1.215728      0.6

15684   20160331  '2315  0.208805    -86.795057     43.866143  0.657376  1.225620      0.6
```

	Date	Time	Mean	Centroid_lon	Centroid_lat	Std_lon	Std_lat	Skewne
15685	20160331	'2330	0.141006	-86.806141	43.834040	0.650572	1.209149	0.7
15686	20160331	'2345	0.086758	-86.874062	43.660057	0.613668	1.186940	0.8

15687 rows × 19 columns

In [6]:

```
lake[lake['Min'] == '00'] #3103
```

Out[6]:

	Date	Time	Mean	Centroid_lon	Centroid_lat	Std_lon	Std_lat	Skewnes
0	20151001	'0000	0.001967	-86.745549	43.885599	0.646480	1.212252	0.64
2	20151001	'0100	0.001764	-86.749381	43.939543	0.628686	1.202172	0.6
6	20151001	'0200	0.001927	-86.729101	43.923259	0.652894	1.245747	0.5
10	20151001	'0300	0.001974	-86.736051	43.964181	0.650661	1.226098	0.6
13	20151001	'0400	0.002177	-86.735131	43.908864	0.642794	1.240158	0.6
...	...	...	...	...	...	...	...	
15667	20160331	'1800	0.930147	-86.750086	43.973649	0.647680	1.197397	0.6
15670	20160331	'1900	0.915581	-86.717445	43.966764	0.631820	1.192493	0.6
15676	20160331	'2100	0.570950	-86.550784	44.329053	0.671158	1.213310	0.3

	Date	Time	Mean	Centroid_lon	Centroid_lat	Std_lon	Std_lat	Skewnes
15679	20160331	'2200	0.472322	-86.681743	44.195860	0.685584	1.171544	0.5
15683	20160331	'2300	0.263741	-86.778112	43.942255	0.659147	1.215728	0.6

4353 rows × 19 columns

In [7]:
```
len(lake['Date'].unique())
```

Out[7]:
```
183
```

In [8]:
```
list_da = []
list_ti = []
list_m = []
list_cx = []
list_cy = []
list_sx = []
list_sy = []
list_skx = []
list_sky = []
list_kx = []
list_ky = []
list_cloud_pixel_num = []
list_non_nan_num = []
list_percentage_cloud_cover = []
list_len =[]
list_case = []
```

In [9]:
```
# a = lake[(lake['Date'] == 20171231) & (lake['Hour'] == '23')& (lake['Min']
# a
```

In [10]:
```
# a['Date'].values[0]
```

In [11]:
```
# if ((len(a) == 1) & ((a['Min']== '45').bool())):
#     print('exist')
# else:
#     print('no-exist')
```

In [12]:
```
# lake = lake.iloc[2:6, :]
# lake
```

In [13]:

```
for d in tqdm(lake['Date'].unique()):
#     print(d)
    for t1 in (lake[lake['Date'] == d]['Hour']).unique():
        t1_v = lake[(lake['Date'] == d) & (lake['Hour'] == t1)]
        t1_n = len(lake[(lake['Date'] == d) & (lake['Hour'] == t1)])
        n00 = len(lake[(lake['Date'] == d) & (lake['Hour'] == t1)& (lake['Mi
        n15 = len(lake[(lake['Date'] == d) & (lake['Hour'] == t1)& (lake['Mi
        n30 = len(lake[(lake['Date'] == d) & (lake['Hour'] == t1)& (lake['Mi
        n45 = len(lake[(lake['Date'] == d) & (lake['Hour'] == t1)& (lake['Mi
#         print(n00)
        list_len.append(t1_n)
        found = 0
        while found < 1:
            if n00 == 1:
                row_f = lake[(lake['Date'] == d) & (lake['Hour'] == t1)& (la
                list_da.append(row_f['Date'].values[0])
                list_ti.append(row_f['Time'].values[0])
                list_m.append(row_f['Mean'].values[0])

                list_cx.append(row_f['Centroid_lon'].values[0])
                list_cy.append(row_f['Centroid_lat'].values[0])
                list_sx.append(row_f['Std_lon'].values[0])
                list_sy.append(row_f['Std_lat'].values[0])
                list_skx.append(row_f['Skewness_lon'].values[0])
                list_sky.append(row_f['Skewness_lat'].values[0])
                list_kx.append(row_f['Kurtosis_lon'].values[0])
                list_ky.append(row_f['Kurtosis_lat'].values[0])
                list_cloud_pixel_num.append(row_f['Cloud_Cover_Count'].value
                list_non_nan_num.append(row_f['Total_Lake_Pixel'].values[0])
                list_percentage_cloud_cover.append(row_f['Percent_Cloud_Cove
                list_case.append('s00')
#                 print('1---'+ str(row_f['Date'].values[0])+ row_f['Time'].
                found = found +1
#                 print(found)
            else:
                if ((n00 == 0) & (n15 == 0) & (n45 == 0)):
                    list_da.append(t1_v['Date'].values[0])
                    list_ti.append('N/A')
                    list_m.append('N/A')

                    list_cx.append('N/A')
                    list_cy.append('N/A')
                    list_sx.append('N/A')
                    list_sy.append('N/A')
                    list_skx.append('N/A')
                    list_sky.append('N/A')
                    list_kx.append('N/A')
                    list_ky.append('N/A')
                    list_cloud_pixel_num.append('N/A')
                    list_non_nan_num.append('N/A')
                    list_percentage_cloud_cover.append('N/A')
                    list_case.append('s30')
#                     print('2---'+ str(row_f['Date'].values[0]) + row_f['Ti
                    found = found +1
                elif((n00 == 0) & (n15 == 1) & (n45 == 0)):
                    row_f = lake[(lake['Date'] == d) & (lake['Hour'] == t1)&
                    list_da.append(row_f['Date'].values[0])
                    list_ti.append(row_f['Time'].values[0])
                    list_m.append(row_f['Mean'].values[0])
                    list_cx.append(row_f['Centroid_lon'].values[0])
                    list_cy.append(row_f['Centroid_lat'].values[0])
                    list_sx.append(row_f['Std_lon'].values[0])
                    list_sy.append(row_f['Std_lat'].values[0])
                    list_skx.append(row_f['Skewness_lon'].values[0])
                    list_sky.append(row_f['Skewness_lat'].values[0])
                    list_kx.append(row_f['Kurtosis_lon'].values[0])
                    list_ky.append(row_f['Kurtosis_lat'].values[0])
                    list_cloud_pixel_num.append(row_f['Cloud_Cover_Count'].v
                    list_non_nan_num.append(row_f['Total_Lake_Pixel'].values
                    list_percentage_cloud_cover.append(row_f['Percent_Cloud_
                    list_case.append('s15')
#                     print('3---' + str(row_f['Date'].values[0])+ row_f['Ti
                    found = found +1
                elif((n00 == 0) & (n15 == 0) & (n45 == 1)):
                    row_f = lake[(lake['Date'] == d) & (lake['Hour'] == t1)&
                    list_da.append(row_f['Date'].values[0])
                    list_ti.append(row_f['Time'].values[0])
                    list_m.append(row_f['Mean'].values[0])
                    list_cx.append(row_f['Centroid_lon'].values[0])
                    list_cy.append(row_f['Centroid_lat'].values[0])
                    list_sx.append(row_f['Std_lon'].values[0])
                    list_sy.append(row_f['Std_lat'].values[0])
                    list_skx.append(row_f['Skewness_lon'].values[0])
                    list_sky.append(row_f['Skewness_lat'].values[0])
                    list_kx.append(row_f['Kurtosis_lon'].values[0])
                    list_ky.append(row_f['Kurtosis_lat'].values[0])
                    list_cloud_pixel_num.append(row_f['Cloud_Cover_Count'].v
                    list_non_nan_num.append(row_f['Total_Lake_Pixel'].values
                    list_percentage_cloud_cover.append(row_f['Percent_Cloud_
                    list_case.append('s45')
#                     print('4---' + str(row_f['Date'].values[0])+ row_f['Ti
                    found = found +1
                else:
                    row_f = lake[(lake['Date'] == d) & (lake['Hour'] == t1)&
                    row_f_2 = lake[(lake['Date'] == d) & (lake['Hour'] == t1
#                     print(row_f)
                    list_da.append(row_f['Date'].values[0])
                    list_ti.append(row_f['Time'].values[0])
                    list_m.append(((row_f['Mean'].values[0]) + (row_f_2['Mea

                    list_cx.append(((row_f['Centroid_lon'].values[0]) + (row
                    list_cy.append(((row_f['Centroid_lat'].values[0]) + (row
                    list_sx.append(((row_f['Std_lon'].values[0]) + (row_f_2[
                    list_sy.append(((row_f['Std_lat'].values[0]) + (row_f_2[
                    list_skx.append(((row_f['Skewness_lon'].values[0])+ (row
                    list_sky.append(((row_f['Skewness_lat'].values[0])+ (row
                    list_kx.append(((row_f['Kurtosis_lon'].values[0])+(row_f
                    list_ky.append(((row_f['Kurtosis_lat'].values[0])+ (row_
                    list_cloud_pixel_num.append(((row_f['Cloud_Cover_Count']
                    list_non_nan_num.append(((row_f['Total_Lake_Pixel'].valu
                    list_percentage_cloud_cover.append(((row_f['Percent_Clou

                    list_case.append('s1545')
#                     print('5---' + str(row_f['Date'].values[0])+ row_f['Ti
                    found = found +1
```

```
100%|██████████| 183/183 [01:26<00:00,  2.13it/s]
```

In [14]:

```
print(len(list_da))
print(len(list_ti))
print(len(list_m))
print(len(list_cx))
print(len(list_cy))
print(len(list_sx))
print(len(list_sy))
print(len(list_skx))
print(len(list_sky))
print(len(list_kx))
print(len(list_ky))
print(len(list_cloud_pixel_num))
print(len(list_non_nan_num))
print(len(list_percentage_cloud_cover))
print(len(list_len))
print(len(list_case))
```

```
4390
4390
4390
4390
4390
4390
4390
4390
4390
4390
4390
4390
4390
4390
4390
4390
```

TO-DO: Change the directory for outputs

Please clearly label the result as it is coming from Step 4.

In [15]:

```
#columns = ['Filename', 'Mean', 'Std', 'Skewness']
data = {'Date':list_da, 'Time': list_ti,'Mean':list_m,
        'Centroid_lon':list_cx,'Centroid_lat':list_cy,
        'Std_lon':list_sx,'Std_lat':list_sy,'Skewness_lon':list_skx, 'Skewne
        'Kurtosis_lon': list_kx, 'Kurtosis_lat': list_ky,
        'Cloud_Cover_Count': list_cloud_pixel_num,
        'Total_Lake_Pixel': list_non_nan_num, 'Percent_Cloud_Cover': list_pe
        'Sample Number': list_len, 'Selected':list_case}
binned = pd.DataFrame(data = data)

## TO-DO: Change output file name.
binned.to_csv("04_binned_lat_lon_opencv_15f16s_lake_0.csv",index = False)
```

In [16]:

```
binned
```

Out[16]:

	Date	Time	Mean	Centroid_lon	Centroid_lat	Std_lon	Std_lat	Skewnes
0	20151001	'0000	0.001967	-86.745549	43.885599	0.646480	1.212252	0.64
1	20151001	'0100	0.001764	-86.749381	43.939543	0.628686	1.202172	0.61
2	20151001	'0200	0.001927	-86.729101	43.923259	0.652894	1.245747	0.59
3	20151001	'0300	0.001974	-86.736051	43.964181	0.650661	1.226098	0.69
4	20151001	'0400	0.002177	-86.735131	43.908864	0.642794	1.240158	0.60
...	...	...	...	...	...	...	...	
4385	20160331	'1900	0.915581	-86.717445	43.966764	0.631820	1.192493	0.65
4386	20160331	'2015	0.719529	-86.611805	44.082250	0.615790	1.251751	0.52
4387	20160331	'2100	0.570950	-86.550784	44.329053	0.671158	1.213310	0.32
4388	20160331	'2200	0.472322	-86.681743	44.195860	0.685584	1.171544	0.51
4389	20160331	'2300	0.263741	-86.778112	43.942255	0.659147	1.215728	0.63

4390 rows × 16 columns

In [ ]: