Step 1: Extract Imagery Data for Lake Michigan from GOES

This is the 1st step of the experiment. Before get started, you should obtain the visible band data generated by GOES Satellite service. The original file is in `.nc` format, and it contains the full scan of the continent. However, in our application, it is not needed. Therefore, we choose to use NOAA's Weather and Climate Toolkit (Viewer and Data Exporter) to extract the data for the area which we are interested in. The files exported by the toolkit is in `.csv` format.

In [1]:

```
!pip install scipy # Just in case.
```

```
Defaulting to user installation because normal site-packages is not writeab
le
Requirement already satisfied: scipy in /opt/tljh/user/lib/python3.8/site-p
ackages (1.10.1)
Requirement already satisfied: numpy<1.27.0,>=1.19.5 in /home/jupyter-dogg
o/.local/lib/python3.8/site-packages (from scipy) (1.23.5)
```

In [2]:

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

In [3]:

```
from scipy.stats import skew
```

TO-DO:

In this part, please change the directory path to the `.csv` files generated by NOAA's toolkit.

In [4]:

```
os.getcwd()
os.chdir('GOES_Hourly_Statistics/raw_csv/2010Fall_2011Spring')
os.getcwd()
```

Out[4]:

```
'/srv/scratch/NOAA/GOES_Hourly_Statistics/raw_csv/2010Fall_2011Spring'
```

In [5]:

```
len(os.listdir())
```

Out[5]:

```
15050
```

In [6]:

```
filename = os.listdir()
len(filename)
```

Out[6]:

```
15050
```

In [7]:

```
filename[0:10]
```

Out[7]:

```
['goes11.2010.12.10.0300.v01.nc-var1-t0.csv',
 'goes11.2010.12.12.1615.v01.nc-var1-t0.csv',
 'goes11.2010.12.11.0115.v01.nc-var1-t0.csv',
 'goes13.2011.03.08.2145.v01.nc-var1-t0.csv',
 'goes13.2011.02.18.1430.v01.nc-var1-t0.csv',
 'goes13.2011.01.07.1300.v01.nc-var1-t0.csv',
 'goes13.2011.03.21.0630.v01.nc-var1-t0.csv',
 'goes11.2010.12.18.0645.v01.nc-var1-t0.csv',
 'goes13.2011.02.04.1415.v01.nc-var1-t0.csv',
 'goes11.2010.10.22.2030.v01.nc-var1-t0.csv']
```

Important: Sort is a must for linux system.

In [8]:

```
filename.sort()
```

In [9]:

```
## Quick inspection of the order.

filename[0:10]
```

Out[9]:

```
['goes11.2010.10.01.0000.v01.nc-var1-t0.csv',
 'goes11.2010.10.01.0030.v01.nc-var1-t0.csv',
 'goes11.2010.10.01.0045.v01.nc-var1-t0.csv',
 'goes11.2010.10.01.0100.v01.nc-var1-t0.csv',
 'goes11.2010.10.01.0115.v01.nc-var1-t0.csv',
 'goes11.2010.10.01.0130.v01.nc-var1-t0.csv',
 'goes11.2010.10.01.0145.v01.nc-var1-t0.csv',
 'goes11.2010.10.01.0200.v01.nc-var1-t0.csv',
 'goes11.2010.10.01.0215.v01.nc-var1-t0.csv',
 'goes11.2010.10.01.0230.v01.nc-var1-t0.csv']
```

TO-DO:

The `Lake_Partition.csv` contains a table of the boundary values which can be used to precisely extract the data from Lake Michigan area.

partition

In [10]:

```
## TO-DO: Change the directory if needed.
LP = pd.read_csv('/srv/scratch/NOAA/Lake_Partition.csv')
LP
```

Out[10]:

	latitude	first_bound	second_bound	left_zone	mid_zone	right_zone
0	40.06	-87.5465	-84.8170	2	3	4
1	40.10	-87.5465	-84.8170	2	3	4
2	40.14	-87.5465	-84.8170	2	3	4
3	40.18	-87.5465	-84.8170	2	3	4
4	40.22	-87.5465	-84.8170	2	3	4
...	...	...	...	...	...	...
193	47.78	-86.1781	-84.9473	6	6	6
194	47.82	-86.1781	-84.9473	6	6	6
195	47.86	-86.1781	-84.9473	6	6	6
196	47.90	-86.1781	-84.9473	6	6	6
197	47.94	-86.1781	-84.9473	6	6	6

198 rows × 6 columns

In [11]:

```
# Optional: Check the list of filenames one more time.
for i in filename[:3]:
    print(i)
```

```
goes11.2010.10.01.0000.v01.nc-var1-t0.csv
goes11.2010.10.01.0030.v01.nc-var1-t0.csv
goes11.2010.10.01.0045.v01.nc-var1-t0.csv
```

TO-DO:

Make sure to change the directory of the outputs!

In [12]:

```
!pwd
```

```
/srv/scratch/NOAA/GOES_Hourly_Statistics/raw_csv/2010Fall_2011Spring
```

In [13]:

```
#!mkdir /srv/scratch/NOAA/GOES_Hourly_Statistics/zone_0_2007Fall_2008Spring/
```

In [14]:

```
for fn in tqdm(filename[:2000]):
    file = pd.read_csv(fn)
    flp = pd.merge(file, LP, on = 'latitude')
    par_list = []
    for i in range(len(flp)):
        a = flp['longitude'][i]
        b = flp['first_bound'][i]
        c = flp['second_bound'][i]
        if a < b:
            par_list.append(flp['left_zone'][i])
        elif a > c:
            par_list.append(flp['right_zone'][i])
        else:
            par_list.append(flp['mid_zone'][i])
    flp['partition'] = par_list
    flp = flp.loc[:, ['value', 'datetime', 'latitude', 'longitude', 'partiti
    s_flp = flp[flp['partition'] == 0].reset_index().rename(columns={'index'

    ## TO-DO: Change the directory, clearly lable as "zone_0"
    s_flp.to_csv('/srv/scratch/NOAA/GOES_Hourly_Statistics/zone_0_2010Fall_2
#     print(fn)
```

```
100%|██████████| 2000/2000 [21:39<00:00,  1.54it/s]
```

In [15]:

```
print("End of current process.")
```

```
End of current process.
```

In [ ]:

In [ ]: