Assignment 2

Before working on this assignment please read these instructions fully. In the submission area, you will notice that you can click the link to **Preview the Grading** for each step of the assignment. This is the criteria that will be used for peer grading. Please familiarize yourself with the criteria before beginning the assignment.

An NOAA dataset has been stored in the file
`data/C2A2_data/BinnedCsvs_d25/d035233802c307b63e773fd6d0b925b4f447b38691b74f670fcb4647.csv`.
The data for this assignment comes from a subset of The National Centers for Environmental Information (NCEI)
Daily Global Historical Climatology Network (https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/readme.txt)
(GHCN-Daily). The GHCN-Daily is comprised of daily climate records from thousands of land surface stations across the globe.

Each row in the assignment datafile corresponds to a single observation.

The following variables are provided to you:

- **id** : station identification code
- **date** : date in YYYY-MM-DD format (e.g. 2012-01-24 = January 24, 2012)
- **element** : indicator of element type
    - TMAX : Maximum temperature (tenths of degrees C)
    - TMIN : Minimum temperature (tenths of degrees C)
- **value** : data value for element (tenths of degrees C)

For this assignment, you must:

1. Read the documentation and familiarize yourself with the dataset, then write some python code which returns a line graph of the record high and record low temperatures by day of the year over the period 2005-2014. The area between the record high and record low temperatures for each day should be shaded.
2. Overlay a scatter of the 2015 data for any points (highs and lows) for which the ten year record (2005-2014) record high or record low was broken in 2015.
3. Watch out for leap days (i.e. February 29th), it is reasonable to remove these points from the dataset for the purpose of this visualization.
4. Make the visual nice! Leverage principles from the first module in this course when developing your solution. Consider issues such as legends, labels, and chart junk.

The data you have been given is near **Tokyo, Tokyo, Japan**, and the stations the data comes from are shown on the map below.

```
In [3]:    import matplotlib.pyplot as plt
           import mplleaflet
           import pandas as pd

           def leaflet_plot_stations(binsize, hashid):

               df = pd.read_csv('data/C2A2_data/BinSize_d{}.csv'.format(binsize))

               station_locations_by_hash = df[df['hash'] == hashid]

               lons = station_locations_by_hash['LONGITUDE'].tolist()
               lats = station_locations_by_hash['LATITUDE'].tolist()

               plt.figure(figsize=(8,8))

               plt.scatter(lons, lats, c='r', alpha=0.7, s=200)

               return mplleaflet.display()

           leaflet_plot_stations(25,'d035233802c307b63e773fd6d0b925b4f447b38691b74f670fcb
           4647')
```
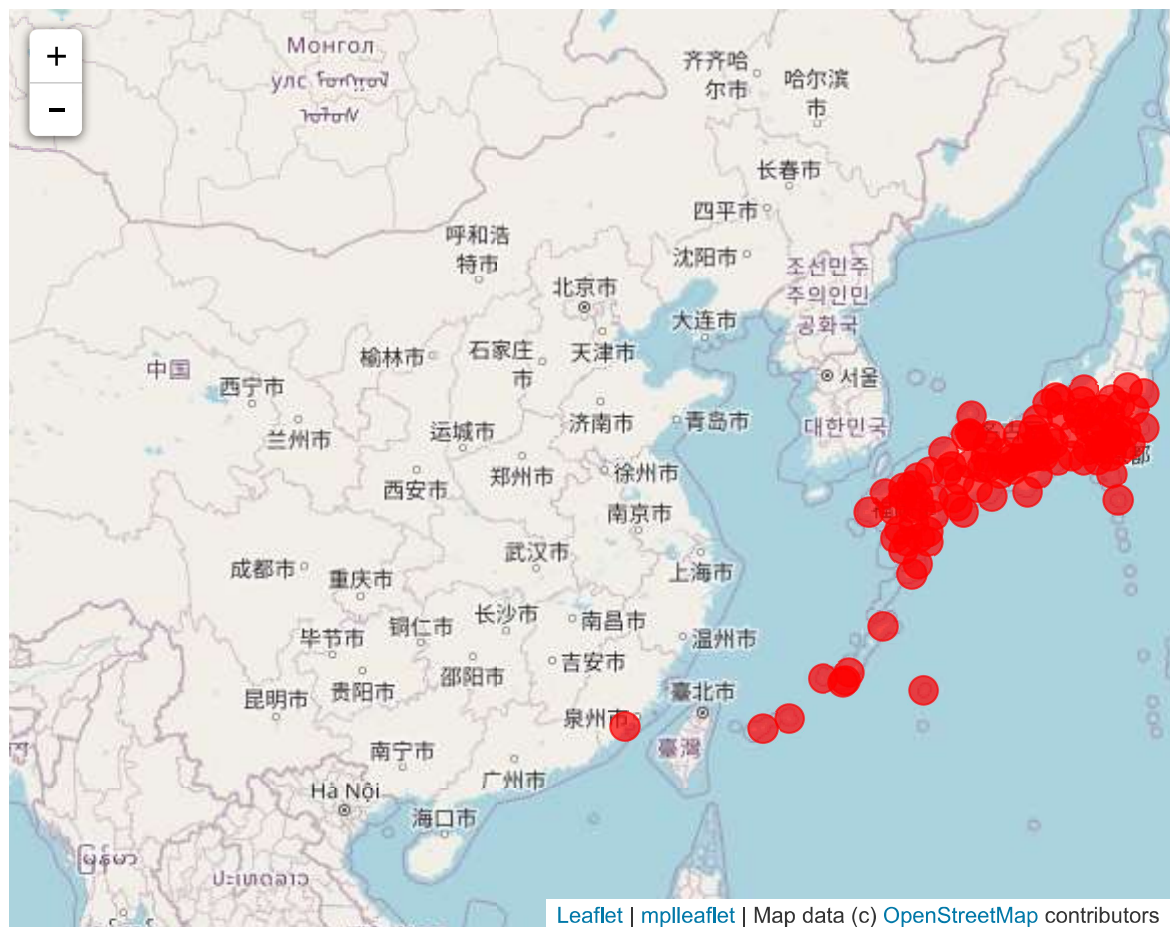
Out[3]:

```
In [3]:  import matplotlib.pyplot as plt
         #import mplleaflet
         import pandas as pd
         import numpy as np
         df = pd.read_csv("/home/sabodhapati/Data_Science_with_Python/Applied_Plotting/
         data/d035233802c307b63e773fd6d0b925b4f447b38691b74f670fcb4647.csv")
         df.head()
```

Out[3]:

	ID	Date	Element	Data_Value
0	JA000047827	2013-11-14	TMAX	206
1	JA000047843	2009-11-12	TMIN	157
2	JA000047610	2009-10-17	TMIN	89
3	JA000047843	2009-11-01	TMAX	202
4	JA000047640	2009-04-03	TMIN	-27

```
In [4]:  #df.shape : (597953, 4)
         df['Data_Value'] = df['Data_Value'] * 0.1 #outof coursera platform memrory
         df['Year'] = df['Date'].apply(lambda x: x[:4])
         df['Date2'] = df['Date'].apply(lambda x: x[-5:])
         df = df[df['Date2'] != '02-29']
         df_05_14 = df[~(df['Year'] == '2015')]
         df_15 = df[df['Year'] == '2015']
         df_05_14.head()
```

Out[4]:

	ID	Date	Element	Data_Value	Year	Date2
0	JA000047827	2013-11-14	TMAX	20.6	2013	11-14
1	JA000047843	2009-11-12	TMIN	15.7	2009	11-12
2	JA000047610	2009-10-17	TMIN	8.9	2009	10-17
3	JA000047843	2009-11-01	TMAX	20.2	2009	11-01
4	JA000047640	2009-04-03	TMIN	-2.7	2009	04-03

```
In [5]:  #temp = pd.DataFrame()
         max_0415 = df_05_14.groupby('Date2').agg({'Data_Value':np.max})
         min_0415 = df_05_14.groupby('Date2').agg({'Data_Value':np.min})
         max_15 = df_15.groupby('Date2').agg({'Data_Value':np.max})
         min_15 = df_15.groupby('Date2').agg({'Data_Value':np.min})
         all_max = pd.merge(max_0415.reset_index(), max_15.reset_index(), left_index=Tr
         ue, on = 'Date2')
         all_min = pd.merge(min_0415.reset_index(), min_15.reset_index(), left_index=Tr
         ue, on = 'Date2')
```

```
In [6]:  break_max = all_max[all_max['Data_Value_y'] > all_max['Data_Value_x']]
         break_min = all_min[all_min['Data_Value_y'] < all_min['Data_Value_x']]
         break_max.head()
```

Out[6]:

	Date2	Data_Value_x	Data_Value_y
4	01-05	24.1	24.2
25	01-26	23.8	24.5
91	04-02	28.0	29.8
93	04-04	28.3	28.8
106	04-17	27.8	28.1

```
In [7]:  %matplotlib inline
         plt.figure(figsize=(16,10))

         plt.plot(max_0415.values, c = 'red', label ='Record High')
         plt.plot(min_0415.values, c = 'blue', label ='Record Low')

         plt.xlabel('Day', fontsize=20)
         plt.ylabel('Temperature', fontsize=20)
         plt.title('Ten Year Record (2005-2014) Was Broken in 2015', fontsize=25)

         plt.scatter(break_max.index.tolist(), break_max['Data_Value_y'].values, c = 'b
         lack', label = "Broken High in 2015")
         plt.scatter(break_min.index.tolist(), break_min['Data_Value_y'].values, c = 'g
         reen', label = "Broken Low in 2015")

         plt.gca().fill_between(range(len(max_0415)),
                            np.array(max_0415.values.reshape(len(min_0415.values
         ),)),
                            np.array(min_0415.values.reshape(len(min_0415.values
         ),)),
                            facecolor='#2F99B4',
                            alpha=0.25)

         plt.gca().spines['top'].set_visible(False)
         plt.gca().spines['right'].set_visible(False)
         plt.legend(loc = 8, fontsize=18, frameon = False)
         plt.show()
```
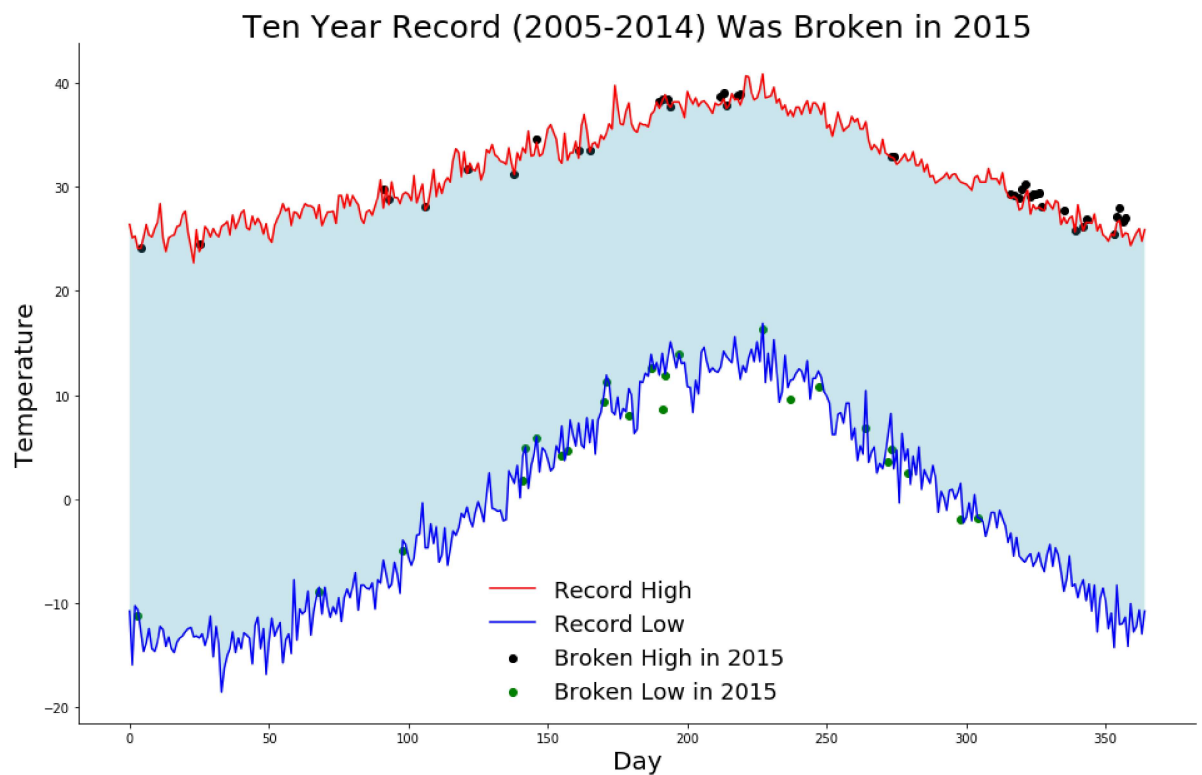

Ten Year Record (2005-2014) Was Broken in 2015

this link is to the data file on your online jupyter

If you run the following code in one of the cells of Assignment 2 notebook on the online platform it will generate a link from which you can download the file

```
In [1]: file ='data/C2A2_data/BinnedCsvs_d400/fb441e62df2d58994928907a91895ec62c2c42e6
        cd075c2700843b89.csv'
        !cp "$file" .
        from IPython.display import HTML
        link = '<a href="{0}" download>Click here to download {0}</a>'
        HTML(link.format(file.split('/')[-1]))
```

Out[1]: Click here to download fb441e62df2d58994928907a91895ec62c2c42e6cd075c2700843b89.csv
        (fb441e62df2d58994928907a91895ec62c2c42e6cd075c2700843b89.csv)
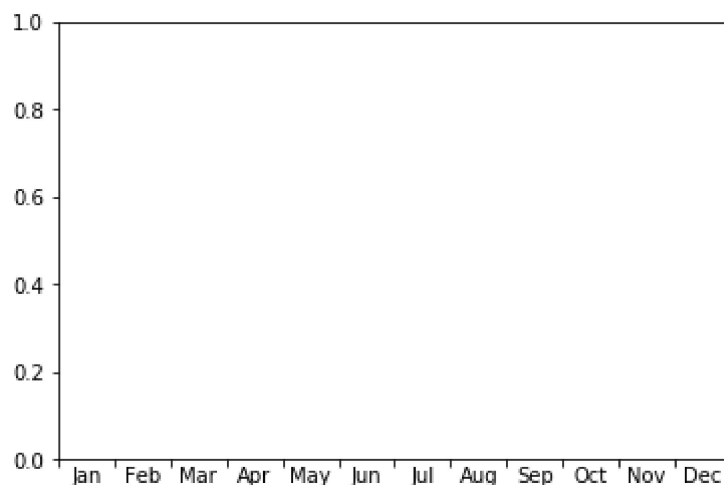
```
In [8]: x = np.arange(0, 365)

        labels = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct'
        , 'Nov', 'Dec']
        ticks = np.arange(min(x), max(x)+len(x)/12, len(x)/12)
```

```
In [9]: minor_ticks = ticks + (len(x)/12)/2
        minor_ticks = minor_ticks[:len(minor_ticks)-1]

        ax = plt.gca()

        ax.set_xticks(ticks)
        ax.set_xticklabels('')
        ax.set_xticks(minor_ticks, minor = True)
        ax.set_xticklabels(labels, minor = True)
        ax.tick_params(axis='x', which = 'minor', length= 0)
```

```
In [10]: minor_ticks
```

```
Out[10]: array([  15.20833333,    45.625     ,    76.04166667,  106.45833333,
                 136.875     ,  167.29166667,  197.70833333,  228.125     ,
                 258.54166667,  288.95833333,  319.375     ,  349.79166667])
```

```
In [11]:
```

Object `set_xticklabels` not found.