Assignment2

June 12, 2020

1 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_d400/fb441e62df2d58994` This is the dataset to use for this assignment. Note: The data for this assignment comes from a subset of The National Centers for Environmental Information (NCEI) Daily Global Historical Climatology Network (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 **Ann Arbor, Michigan, United States**, and the stations the data comes from are shown on the map below.

In [1]:
```
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(400,'fb441e62df2d58994928907a91895ec62c2c42e6cd075c27
```

Out[1]: <IPython.core.display.HTML object>

In [87]:
```
dat=pd.read_csv('data/C2A2_data/BinnedCsvs_d400/fb441e62df2d58994928907a91
dat.head()
```

Out[87]:

	ID	Date	Element	Data_Value
0	USW00094889	2014-11-12	TMAX	22
1	USC00208972	2009-04-29	TMIN	56
2	USC00200032	2008-05-26	TMAX	278
3	USC00205563	2005-11-11	TMAX	139
4	USC00200230	2014-02-27	TMAX	-106

In [88]:
```
dat['Date'] = pd.DatetimeIndex(dat['Date']).date
lis = ['2922008','2922012'] #list of 29 Febs appearing in each leap year
dates = [datetime.datetime.strptime(i, '%d%m%Y').date() for i in lis] #cor
dat = dat[~dat['Date'].isin(dates)] #remove all 29 February dates
```

In [89]:
```
#creating a separate column for the month
dat['Month'] = pd.DatetimeIndex(dat['Date']).month
a= '31122014'
dat['Day']=pd.DatetimeIndex(dat['Date']).weekday
cutoff_date = datetime.datetime.strptime(a,'%d%m%Y').date()
#dataframe for values after 31 Dec 2014
df2 = dat[dat['Date'] <= cutoff_date]
#returning dates before cutoff date
df3 = dat[dat['Date'] > cutoff_date]
```

In [90]:
```
import numpy as np
df_min = df2[df2['Element'] =='TMIN'].groupby('Month').aggregate({'Data_Va
df_max = df2[df2['Element'] == 'TMAX'].groupby('Month').aggregate({'Data_V
```

```
In [91]: %matplotlib notebook
         plt.figure()
         plt.plot(df_max, '-o',label="Maximum Temp")
         plt.plot(df_min,'-o',label="Minimum Temp")
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.HTML object>
```

```
Out[91]: [<matplotlib.lines.Line2D at 0x7fb0952166d8>]
```

```
In [92]: x = df_max.index.values
         plt.fill_between(x,df_min.values.flatten(),df_max.values.flatten(), color=
         plt.xticks(list(df_max.index),['Jan','Feb','Mar','Apr','May','Jun','Jul','
```

```
Out[92]: ([<matplotlib.axis.XTick at 0x7fb0951fb2b0>,
           <matplotlib.axis.XTick at 0x7fb0952a9630>,
           <matplotlib.axis.XTick at 0x7fb0934eb2e8>,
           <matplotlib.axis.XTick at 0x7fb095320198>,
           <matplotlib.axis.XTick at 0x7fb0953128d0>,
           <matplotlib.axis.XTick at 0x7fb095305710>,
           <matplotlib.axis.XTick at 0x7fb0952fe198>,
           <matplotlib.axis.XTick at 0x7fb0952f74a8>,
           <matplotlib.axis.XTick at 0x7fb095264c88>,
           <matplotlib.axis.XTick at 0x7fb0953a5cc0>,
           <matplotlib.axis.XTick at 0x7fb09539ef28>,
           <matplotlib.axis.XTick at 0x7fb095396940>],
          <a list of 12 Text xticklabel objects>)
```

```
In [93]: plt.title('Min and Max temperatures in Ann Arbor Michigan United States be
         plt.xlabel('Years', fontsize=20)
         plt.ylabel('Temperatures(in degree-tenths)', fontsize=15)
         fig = plt.gcf()

         plt.tick_params(labelsize=10)
         plt.show()
```

```
In [94]: #superimpose scatterplot
         df3_max = df3[df3['Data_Value'] > df3['Month'].map(df_max['Data_Value'])]
         df4_min = df3[df3['Data_Value'] < df3['Month'].map(df_min['Data_Value'])]
         plt.scatter(df3_max.Month.tolist(),df3_max['Data_Value'],s=50,c='red',labe
         plt.scatter(df4_min.Month.tolist(),df4_min['Data_Value'],s=30,c='black',la
```

```
Out[94]: <matplotlib.collections.PathCollection at 0x7fb0952449e8>
```

```
In [95]: plt.legend()
```

```
Out[95]: <matplotlib.legend.Legend at 0x7fb0953203c8>
```

```
In [51]:
```