Assignment2

May 28, 2019

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` 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 [4]:
```
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[4]: <IPython.core.display.HTML object>

In [6]:
```
df = pd.read_csv('data/C2A2_data/BinnedCsvs_d400/fb441e62df2d58994928907a91
df.head()
```

Out[6]:

	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

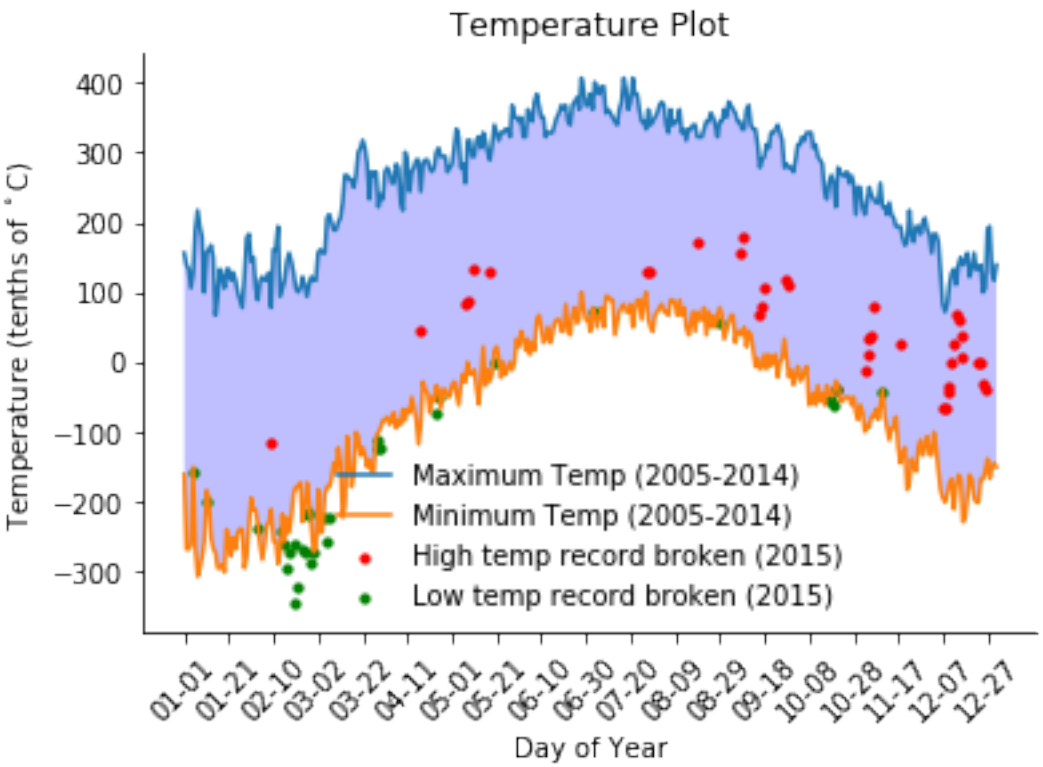
In [7]:
```
df['Year'], df['Month-Date'] = zip(*df['Date'].apply(lambda x: (x[:4], x[5:
df = df[df['Month-Date'] != '02-29']
df.head()
```

Out[7]:

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

In [8]:
```
import numpy as np

temp_max = df[(df['Element'] == 'TMAX') & (df['Year'] != '2015')].groupby('
temp_min = df[(df['Element'] == 'TMIN') & (df['Year'] != '2015')].groupby('
```

```
        temp_max_15 = df[(df['Element'] == 'TMAX') & (df['Year'] == '2015')].groupb
        temp_min_15 = df[(df['Element'] == 'TMIN') & (df['Year'] == '2015')].groupb

In [10]: broken_max = np.where(temp_max_15['Data_Value'] > temp_max['Data_Value'])
         broken_min = np.where(temp_min_15['Data_Value'] < temp_min['Data_Value'])

In [11]: print(broken_max)
         print(broken_min)

[ 39 106 126 127 130 137 207 209 230 249 250 258 259 260 270 271 292 305
 306 307 308 309 321 340 341 342 343 344 345 346 347 348 349 356 357 358
 359]
[  4  10  33  44  45  46  47  49  50  51  53  54  55  56  57  58  63  64
  65  86  87  88 113 114 139 183 239 289 290 291 292 313]


In [14]: plt.figure()

         plt.plot(temp_max.values, label='Maximum Temp (2005-2014)')
         plt.plot(temp_min.values, label='Minimum Temp (2005-2014)')

         plt.gca().fill_between(range(len(temp_min)), temp_min['Data_Value'],temp_m

         plt.xticks(range(0, len(temp_min), 20), temp_min.index[range(0, len(temp_m

         plt.scatter(broken_max, temp_min_15.iloc[broken_max], s=10, color='red', l
         plt.scatter(broken_min, temp_min_15.iloc[broken_min], s=10, color='green',

         plt.legend(frameon = False)

         plt.xlabel('Day of Year')
         plt.ylabel('Temperature (tenths of $^\circ$C)')
         plt.title('Temperature Plot')

         plt.gca().spines['top'].set_visible(False)
         plt.gca().spines['right'].set_visible(False)

         plt.show()
```


Temperature Plot
Temperature (tenths of °C)
400
300
200
100
0
−100
−200
−300
Maximum Temp (2005-2014)
Minimum Temp (2005-2014)
High temp record broken (2015)
Low temp record broken (2015)
01-01
01-21
02-10
03-02
03-22
04-11
05-01
05-21
06-10
06-30
07-20
08-09
08-29
09-18
10-08
10-28
11-17
12-07
12-27
Day of Year

In [ ]: