Yearly Weather Data Plotting

Mini-project

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/fb441e62df2d58994928907a91895ec62c2c42e6cd075c2700843b89`. 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)](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 **Ann Arbor, Michigan, United States**, and the stations the data comes from are shown on the map below.

In [3]:

```python
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

```
In [ ]:
```

```python
import matplotlib.pyplot as plt
import mplleaflet
import pandas as pd
import numpy as np

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,'fb441e62df2d58994928907a91895ec62c2c42e6cd075c2700843b89')
```

```
In [4]:
```

```python
# Load temperature data
df = pd.read_csv('temperature_data.csv')
df.Data_Value = df.Data_Value/10
df.sort_values(['Date','Element'],inplace=True)
df.head()
```

```
Out[4]:
```

	ID	Date	Element	Data_Value
1860	USC00202308	2005-01-01	TMAX	15.0
2073	USC00203712	2005-01-01	TMAX	14.4
3058	USC00205822	2005-01-01	TMAX	12.8
10073	USW00014833	2005-01-01	TMAX	3.3
17153	USC00207320	2005-01-01	TMAX	15.0

In [5]:

```python
# Process Maximum Daily Temperatures

# Step 1: Prepare Max temp data
tmax = df.groupby(['Date','Element']).max()
tmax = pd.DataFrame(pd.pivot_table(tmax, index=[tmax.index.get_level_values(0)], columns=[tmax.index.get_level_values(1)], values=['Data_Value'])['Data_Value']['TMAX'])
tmax.index = pd.to_datetime(tmax.index)
tmax['Year'] = tmax.index.year
tmax['Month'] = tmax.index.month
tmax['Day'] = tmax.index.day
tmonth = tmax['Month']

# Step 2: Remove Leap days and year 2015
tmax2015 = tmax[(tmax.Year == 2015)]
tmax2015 = tmax2015.drop('Year', axis=1).set_index(['Month','Day'])
tmax = tmax[~((tmax.Month == 2) & (tmax.Day == 29)) & (tmax.Year != 2015)]
tmax.reset_index(inplace=True)
tmax.drop(['Date','Year'], axis=1, inplace=True)
tmax = tmax.groupby(['Month','Day']).agg(max)
tmax.reset_index(inplace=True)
```

In [6]:

```python
# Process Minimum Daily Temperatures

# Step 1: Prepare Min temp data
tmin = df.groupby(['Date','Element']).min()
tmin = pd.DataFrame(pd.pivot_table(tmin, index=[tmin.index.get_level_values(0)], columns=[tmin.index.get_level_values(1)], values=['Data_Value'])['Data_Value']['TMIN'])
tmin.index = pd.to_datetime(tmin.index)
tmin['Year'] = tmin.index.year
tmin['Month'] = tmin.index.month
tmin['Day'] = tmin.index.day

# Step 2: Remove Leap days and year 2015
tmin2015 = tmin[(tmin.Year == 2015)]
tmin2015 = tmin2015.drop('Year', axis=1).set_index(['Month','Day'])
tmin = tmin[~((tmin.Month == 2) & (tmin.Day == 29)) & (tmin.Year != 2015)]
tmin.reset_index(inplace=True)
tmin.drop(['Date','Year'], axis=1, inplace=True)
tmin = tmin.groupby(['Month','Day']).agg(min)
tmin.reset_index(inplace=True)
```

In [7]:

```python
# Processing points for the Year 2015

tmax2015.reset_index(inplace=True)
tmax2015 = tmax2015[tmax2015.TMAX>tmax.TMAX]
tmin2015.reset_index(inplace=True)
tmin2015 = tmin2015[tmin2015.TMIN<tmin.TMIN]
```
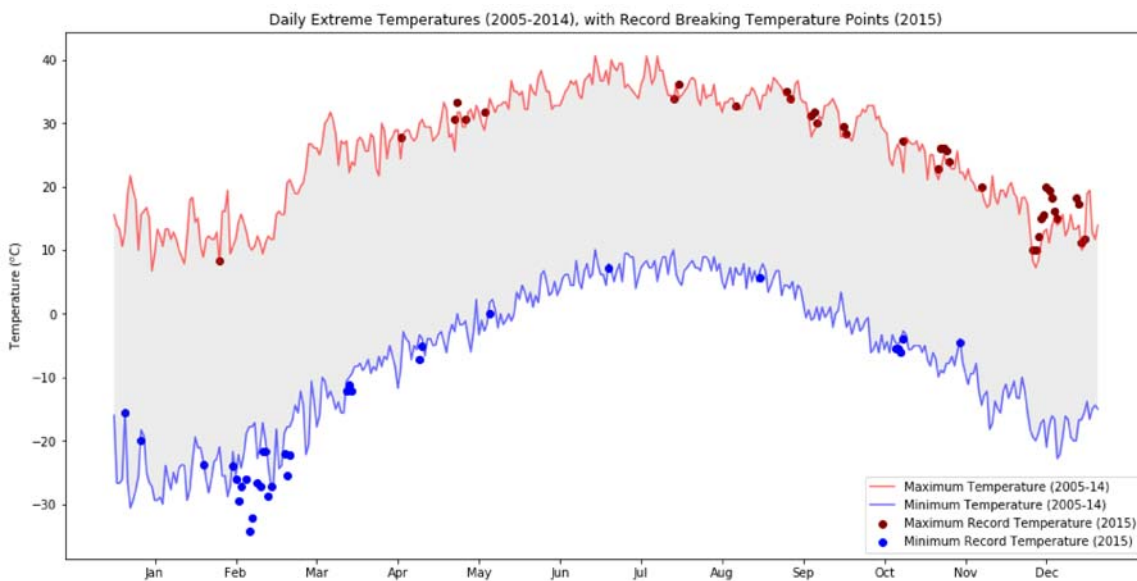
In [9]:

```python
# Plotting the Line Graph (2005-14) and Scatter Plot (2015)

plt.figure(figsize=(16,8))
plt.plot(tmax.reset_index()['TMAX'], color='red', alpha=0.6);
plt.plot(tmin.reset_index()['TMIN'], color='blue', alpha=0.6);
plt.scatter(x=tmax2015.index, y=tmax2015.TMAX, color='maroon');
plt.scatter(x=tmin2015.index, y=tmin2015.TMIN, color='blue');
plt.xticks(np.linspace(15,15 + 30*11 , num = 12),['Jan','Feb','Mar','Apr','May','Jun',
'Jul','Aug','Sep','Oct','Nov','Dec'])

plt.gca().fill_between(range(len(tmax)),
                        tmax['TMAX'], tmin['TMIN'],
                        facecolor='grey', alpha=0.15)

plt.ylabel('Temperature ($^o$C)')
plt.title('Daily Extreme Temperatures (2005-2014), with Record Breaking Temperature Poi
nts (2015)')
# add a Legend with legend entries
plt.legend(['Maximum Temperature (2005-14)',
            'Minimum Temperature (2005-14)',
            'Maximum Record Temperature (2015)',
            'Minimum Record Temperature (2015)']);
```


In [ ]:

In [ ]:

In [ ]:

In [ ]: