Assignment2

September 5, 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 [2]: source=pd.read_csv('data/C2A2_data/BinnedCsvs_d400/fb441e62df2d58994928907a
        source.head()
```

```
Out[2]:            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 [25]: source.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 165085 entries, 0 to 165084
Data columns (total 4 columns):
ID            165085 non-null object
Date          165085 non-null object
Element       165085 non-null object
Data_Value    165085 non-null int64
dtypes: int64(1), object(3)
memory usage: 5.0+ MB
```

```
In [29]: source.shape
```

```
Out[29]: (165002, 6)
```

In [30]:
```
source['ID'].value_counts().sum()
```

Out[30]: 165002

In [31]:
```
source['Year'] = source['Date'].apply(lambda x: x[:4])
source['Date2'] = source['Date'].apply(lambda x: x[-5:])
source = source[source['Date2'] != '02-29']
df_data = source[~(source['Year'] == '2015')]
df_data.head()
```

Out[31]:

	ID	Date	Element	Data_Value	Year	Date2
0	USW00094889	2014-11-12	TMAX	2.2	2014	11-12
1	USC00208972	2009-04-29	TMIN	5.6	2009	04-29
2	USC00200032	2008-05-26	TMAX	27.8	2008	05-26
3	USC00205563	2005-11-11	TMAX	13.9	2005	11-11
4	USC00200230	2014-02-27	TMAX	-10.6	2014	02-27

In [36]:
```
import numpy as np
df_2015 = source[source['Year'] == '2015']
max_data1 = df_data.groupby('Date2').agg({'Data_Value':np.max})
min_data1 = df_data.groupby('Date2').agg({'Data_Value':np.min})
max_2015 = df_2015.groupby('Date2').agg({'Data_Value':np.max})
min_2015 = df_2015.groupby('Date2').agg({'Data_Value':np.min})
all_max = pd.merge(max_data1.reset_index(), max_2015.reset_index(), left_i
all_min = pd.merge(min_data1.reset_index(), min_2015.reset_index(), left_i
```

In [37]:
```
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[37]:

	Date2	Data_Value_x	Data_Value_y
39	02-09	7.8	8.3
106	04-17	24.4	27.8
126	05-07	25.6	30.6
127	05-08	31.7	33.3
130	05-11	29.4	30.6

In [65]:
```
import seaborn as sns
plt.figure(figsize=(15,15))
plt.plot(max_data1.values, c = 'red', label ='Record High')
plt.plot(min_data1.values, c='blue', label ='Record Low')

plt.gca().fill_between(range(len(max_data1)),
                       np.array(max_data1.values.reshape(len(min_data1.val
                       np.array(min_data1.values.reshape(len(min_data1.val
                       facecolor='blue',
                       alpha=0.2)

plt.xlabel('Day', fontsize=20)
```

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

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

plt.scatter(break_max.index.tolist(), break_max['Data_Value_y'].values, c
plt.scatter(break_min.index.tolist(), break_min['Data_Value_y'].values, c
plt.show()
```