Assignment2

June 28, 2021

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.

```python
In [8]:  %matplotlib inline
         import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
         from datetime import datetime
         from matplotlib.pyplot import figure
         from matplotlib import dates, pyplot
         df = pd.read_csv("data/C2A2_data/BinnedCsvs_d400/fb441e62df2d58994928907a91
         df['Data_Value'] = df['Data_Value'].apply(lambda x: x / 10)
         df['Year'] = df['Date'].apply(lambda x: x[:4])
         df['Month-Day'] = df['Date'].apply(lambda x: x[5:])
         df['Day'] = df['Date'].apply(lambda x: x[8:])
         df = df[df['Month-Day'] != '02-29']

In [9]:  #sorting the dataframe according to date
         sort_df = df.sort_values("Date")

         #collecting all TMAXs and TMINs between 2005 and 2014
         sort_df = sort_df[~sort_df['Date'].str.startswith('2015')]

         df_TMAX = sort_df.where(sort_df["Element"] == "TMAX").dropna()
         df_TMIN = sort_df.where(sort_df["Element"] == "TMIN").dropna()

         #collecting maxes and mins for each day
         df_TMAX_grouped = df_TMAX[["Month-Day", "Data_Value", "Date"]].groupby("Mor
         df_TMIN_grouped = df_TMIN[["Month-Day", "Data_Value", "Date"]].groupby("Mor

In [10]: #plotting the data
         df_TMAX_grouped_values = np.array(df_TMAX_grouped["Data_Value"])
         df_TMIN_grouped_values = np.array(df_TMIN_grouped["Data_Value"])
         df_TMAX_grouped_dates = (np.array(df_TMAX_grouped["Date"]))
         df_TMIN_grouped_dates = (np.array(df_TMIN_grouped["Date"]))

In [11]: sort_df_2015 = df[df['Date'].str.startswith('2015')]
         sort_df_2015 = sort_df_2015.sort_values("Date")
         df_TMAX_2015 = sort_df_2015.where(sort_df_2015["Element"] == "TMAX").dropr
         df_TMIN_2015 = sort_df_2015.where(sort_df_2015["Element"] == "TMIN").dropr

         #collecting maxes and mins for each day
         df_TMAX_grouped_2015 = df_TMAX_2015[["Month-Day", "Data_Value", "Date"]].g
         df_TMIN_grouped_2015 = df_TMIN_2015[["Month-Day", "Data_Value", "Date"]].g

In [12]: df_TMAX_grouped_dates_2015 = np.array(pd.to_datetime(df_TMAX_grouped_2015[
         df_TMIN_grouped_dates_2015 = np.array(pd.to_datetime(df_TMAX_grouped_2015[
         df_TMAX_grouped_values_2015 = np.array(df_TMAX_grouped_2015["Data_Value"])
         df_TMIN_grouped_values_2015 = np.array(df_TMIN_grouped_2015["Data_Value"])


         new_max_values = np.array(df_TMAX_grouped_2015["Data_Value"].where(df_TMAX
```

2

```
        new_min_values = np.array(df_TMIN_grouped_2015["Data_Value"].where(df_TMIN
        new_max_dates =  np.array(df_TMAX_grouped_2015["Date"].where(df_TMAX_group
        new_min_dates = np.array(df_TMIN_grouped_2015["Date"].where(df_TMIN_groupe
```

In [13]:
```python
#plotting the data
dates = np.array(pd.date_range(start="01-01-2015",end="12-31-2015").to_pyd
plt.figure(figsize = (12, 12), dpi = 60)
plt.plot(dates, df_TMAX_grouped_values, label = "Record high, 2005 - 2014'
plt.plot(dates, df_TMIN_grouped_values, label = "Record low, 2005 - 2014")
plt.scatter(dates, new_max_values, c = 'r', label = "New high, 2015")
plt.scatter(dates, new_min_values, c = 'g', label = "New low, 2015")
plt.gca().fill_between(dates, df_TMAX_grouped_values, df_TMIN_grouped_valu
plt.legend()
plt.xlabel("Dates")
plt.ylabel("Temperature in $^{\circ}C$")
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.title("Temperature Summary from 2005 - 2014 and 2015 in Michigan")
```
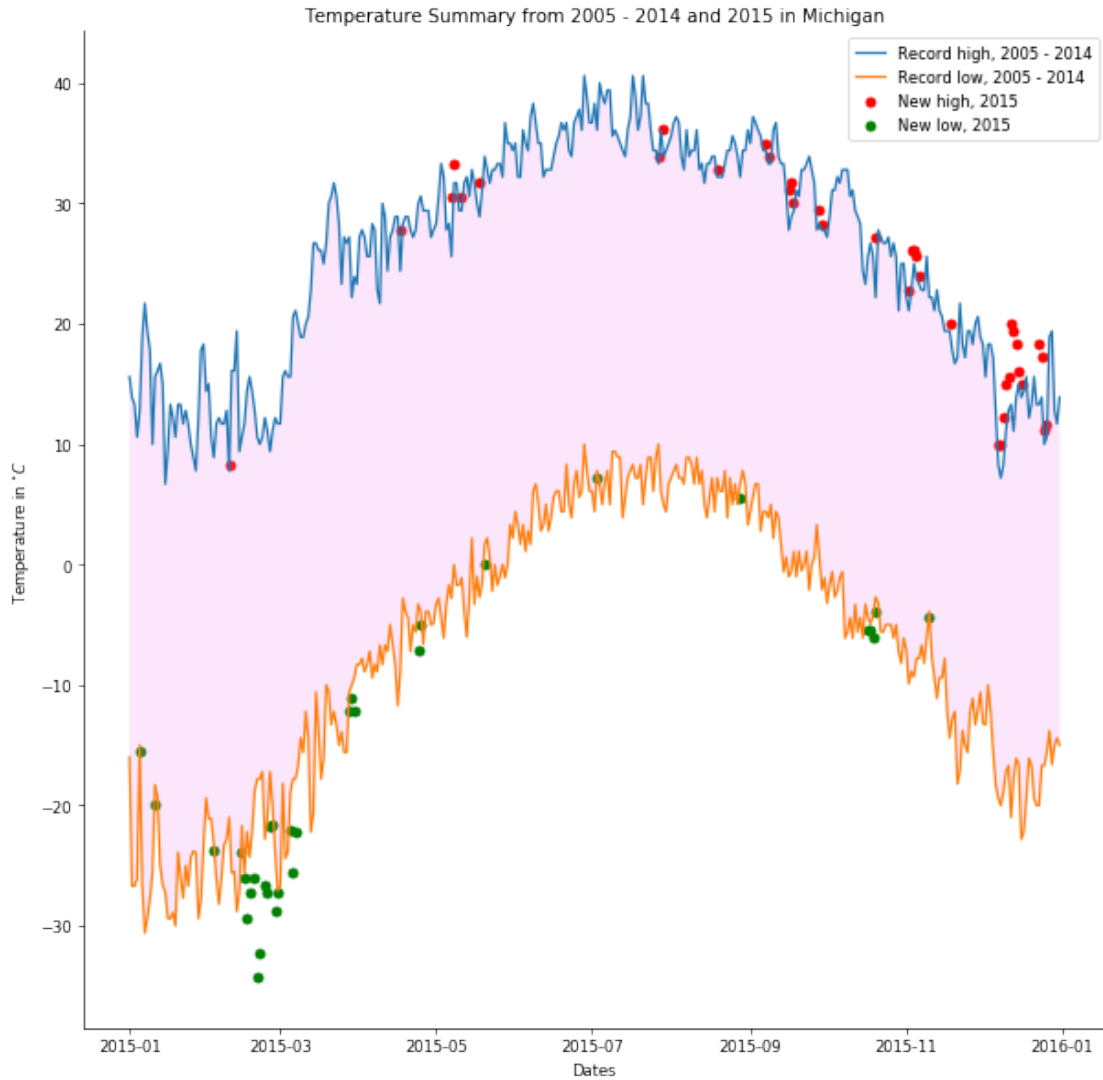
Out[13]: <matplotlib.text.Text at 0x7faacfc1cd30>

Temperature Summary from 2005 - 2014 and 2015 in Michigan

```
In [14]: 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,'fb441e62df2d58994928907a91895ec62c2c42e6cd075c2
```

Out[14]: <IPython.core.display.HTML object>