Repeat_Assignment_3_east

December 6, 2018

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt

In [2]: weather = pd.read_csv("../group_assignment3/weather.csv")

In [3]: weather.head()

Out[3]:    Unnamed: 0           ID  YEARMONTHDAY ELEMENT  DATA VALUE M-FLAG Q-FLAG  \
        0          91  USS0019L03S      19800101    PRCP           0    NaN    NaN
        1         465  USC00048446      19800101    PRCP           0    NaN    NaN
        2         490  USC00040115      19800101    PRCP           0    NaN    NaN
        3         735  USW00023110      19800101    TMAX         172    NaN    NaN
        4         736  USW00023110      19800101    TMIN          89    NaN    NaN

          S-FLAG  OBS-TIME
        0      T       NaN
        1      0     800.0
        2      0     800.0
        3      X       NaN
        4      X       NaN
```

0.0.1 Retrive the weather data for the relevant time periods for stations within 10 miles of East Alameda

```
In [4]: #check the time range of weather data
        (min(weather['YEARMONTHDAY'].unique()), max(weather['YEARMONTHDAY'].unique()))

Out[4]: (19800101, 20091231)

In [5]: stations_within_10miles_east_alameda = pd.read_csv("../group_assignment4/Intermediate_

In [6]: len(stations_within_10miles_east_alameda)

Out[6]: 30

In [7]: #Remove 'TMIN' from weather data.
        weather_new = weather[weather['ELEMENT'] != 'TMIN']
```

```
In [8]: #Merge the 'stations_within_10miles_east_alameda' with 'weather' on station's ID.
        #Get rid of all stations not appearing in 'stations_within_10miles_east_alameda'
        merged = stations_within_10miles_east_alameda.merge(weather_new, on='ID', how='left')
        merged.head()
```

```
Out[8]:    Unnamed: 0_x  Unnamed: 0.1           ID  LATITUDE  LONGITUDE  ELEVATION  \
        0             0             3  US1CAAL0004   37.6483  -121.8745      107.0
        1             0             3  US1CAAL0004   37.6483  -121.8745      107.0
        2             0             3  US1CAAL0004   37.6483  -121.8745      107.0
        3             0             3  US1CAAL0004   37.6483  -121.8745      107.0
        4             0             3  US1CAAL0004   37.6483  -121.8745      107.0

          STATE                NAME  GSN FLAG HCN/CRN FLAG  WMO ID  INVDIST  \
        0    CA  PLEASANTON 1.8 SSE       NaN          NaN     NaN  0.13654
        1    CA  PLEASANTON 1.8 SSE       NaN          NaN     NaN  0.13654
        2    CA  PLEASANTON 1.8 SSE       NaN          NaN     NaN  0.13654
        3    CA  PLEASANTON 1.8 SSE       NaN          NaN     NaN  0.13654
        4    CA  PLEASANTON 1.8 SSE       NaN          NaN     NaN  0.13654

           Unnamed: 0_y  YEARMONTHDAY ELEMENT  DATA VALUE M-FLAG Q-FLAG S-FLAG  \
        0    17849812.0    20080701.0    PRCP         0.0    NaN    NaN      N
        1    17949981.0    20080702.0    PRCP         0.0    NaN    NaN      N
        2    18051494.0    20080703.0    PRCP         0.0    NaN    NaN      N
        3    18153195.0    20080704.0    PRCP         0.0    NaN    NaN      N
        4    18253026.0    20080705.0    PRCP         0.0    NaN    NaN      N

           OBS-TIME
        0       NaN
        1       NaN
        2       NaN
        3       NaN
        4       NaN
```

0.0.2 Identify the stations that meet Ranson's criteria in east Alameda for inclusion in each year:

Missing values:

```
In [9]: #Check how many missing values in each column
        merged.isnull().sum()
```

```
Out[9]: Unnamed: 0_x        0
        Unnamed: 0.1        0
        ID                  0
        LATITUDE            0
        LONGITUDE           0
        ELEVATION           0
        STATE               0
        NAME                0
```

```
GSN FLAG        85575
HCN/CRN FLAG    63846
WMO ID          85575
INVDIST             0
Unnamed: 0_y       17
YEARMONTHDAY       17
ELEMENT            17
DATA VALUE         17
M-FLAG          54715
Q-FLAG          84899
S-FLAG             17
OBS-TIME        23264
dtype: int64
```

There are 17 rows with missing 'ELEMENT' and 'DATA VALUE'. It is probably because some stations in 'stations.csv' don't have corresponding records in 'weather.csv', so we remove missing values.

check duplicates:

```
In [10]: #Drop missing values in 'ELEMENT', 'DATA VALUE'
         stations_weather = merged.dropna(subset=['ELEMENT', 'DATA VALUE'])
```

```
In [11]: #Check the length before and after dropping duplicates
         len(stations_weather) == len(stations_weather.drop_duplicates(['ID', 'YEARMONTHDAY',
```

```
Out[11]: True
```

No duplicates

```
In [12]: stations_weather.tail()
```

```
Out[12]:        Unnamed: 0_x  Unnamed: 0.1           ID  LATITUDE  LONGITUDE  \
         85570            29          2657  USW00023285   37.6928  -121.8144
         85571            29          2657  USW00023285   37.6928  -121.8144
         85572            29          2657  USW00023285   37.6928  -121.8144
         85573            29          2657  USW00023285   37.6928  -121.8144
         85574            29          2657  USW00023285   37.6928  -121.8144

                ELEVATION STATE               NAME  GSN FLAG HCN/CRN FLAG  WMO ID  \
         85570      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     NaN
         85571      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     NaN
         85572      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     NaN
         85573      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     NaN
         85574      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     NaN

                 INVDIST  Unnamed: 0_y  YEARMONTHDAY ELEMENT  DATA VALUE M-FLAG Q-FLAG  \
         85570  0.143148    36575687.0    20091229.0    PRCP         3.0    NaN    NaN
         85571  0.143148    36678341.0    20091230.0    TMAX       150.0    NaN    NaN
         85572  0.143148    36678343.0    20091230.0    PRCP         0.0      T    NaN
```

```
        85573  0.143148    36780952.0    20091231.0     TMAX       150.0     NaN     NaN
        85574  0.143148    36780954.0    20091231.0     PRCP         0.0     NaN     NaN

               S-FLAG  OBS-TIME
        85570       0    2400.0
        85571       0    2400.0
        85572       0    2400.0
        85573       0    2400.0
        85574       0    2400.0
```

Covert 'Data value' temperature to correct value

```
In [13]: #Convert unit of TMAX data value to Fahrenheit which Ranson used:
         stations_weather['DATA VALUE'] = np.where(stations_weather['ELEMENT'] == 'TMAX',
                                                   (stations_weather['DATA VALUE']/10)* 1.8 +
                                                   stations_weather['DATA VALUE'])
```

```
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html
  after removing the cwd from sys.path.
```

```
In [14]: stations_weather.tail()
```

```
Out[14]:        Unnamed: 0_x  Unnamed: 0.1           ID  LATITUDE  LONGITUDE  \
        85570             29          2657  USW00023285   37.6928  -121.8144
        85571             29          2657  USW00023285   37.6928  -121.8144
        85572             29          2657  USW00023285   37.6928  -121.8144
        85573             29          2657  USW00023285   37.6928  -121.8144
        85574             29          2657  USW00023285   37.6928  -121.8144

               ELEVATION STATE               NAME  GSN FLAG HCN/CRN FLAG  WMO ID  \
        85570      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     NaN
        85571      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     NaN
        85572      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     NaN
        85573      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     NaN
        85574      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     NaN

                INVDIST  Unnamed: 0_y  YEARMONTHDAY ELEMENT  DATA VALUE M-FLAG Q-FLAG  \
        85570  0.143148    36575687.0    20091229.0    PRCP         3.0    NaN    NaN
        85571  0.143148    36678341.0    20091230.0    TMAX        59.0    NaN    NaN
        85572  0.143148    36678343.0    20091230.0    PRCP         0.0      T    NaN
        85573  0.143148    36780952.0    20091231.0    TMAX        59.0    NaN    NaN
        85574  0.143148    36780954.0    20091231.0    PRCP         0.0    NaN    NaN

               S-FLAG  OBS-TIME
```

```
85570      0    2400.0
85571      0    2400.0
85572      0    2400.0
85573      0    2400.0
85574      0    2400.0
```

Bias adjustment for each weather station

```
In [15]: stations_temp = stations_weather[stations_weather.ELEMENT=='TMAX']
```

```
In [16]: import random
         # set seed
         np.random.seed(101)

         ID = stations_temp.ID.unique()
         # Create the starting mu
         mu = np.zeros(len(ID))
         mu_old = np.zeros(len(ID))

         while abs(mu_old-mu).any() > 0.01 or mu_old.all() == 0:
             # update mu
             mu_old = mu
             # get reference station
             ref_idx = np.random.choice(np.arange(len(ID)),1)
             ref = ID[ref_idx][0]
             station_i = stations_temp[stations_temp.ID == ref]

             # other station
             for j in ID:
                 station_idx = np.arange(len(ID))[j == ID]
                 station_j = stations_temp[stations_temp.ID == j]
                 # get the date that both stations reported
                 shared_date = station_i[['YEARMONTHDAY']].merge(station_j[['YEARMONTHDAY']], o
                 n = len(shared_date)

                 # if there are shared dates
                 if n>0:
                     temp_i = 0
                     for day in shared_date.YEARMONTHDAY:
                         temp_i = temp_i + float(station_i[station_i.YEARMONTHDAY==day]['DATA V

                     temp_j = 0
                     for day in shared_date.YEARMONTHDAY:
                         temp_j = temp_j + float(station_j[station_j.YEARMONTHDAY==day]['DATA V

                     mu[station_idx] = mu_old[station_idx]+float(temp_i+mu_old[ref_idx]*n - tem
                 # if there's no shared date
                 else:
                     mu[station_idx] = mu_old[station_idx]
```

```
In [17]: mu

Out[17]: array([ 3.4620058 , -0.04247438, -1.08785784,  6.0323726 ,  1.28016482,
                 5.3127417 ,  6.47611683,  9.56719641,  1.1221162 ])
```

county adjusted C

```
In [18]: w = stations_temp.groupby('ID').mean().INVDIST

In [19]: C = sum(w * mu)/sum(w)
         C

Out[19]: 3.385098512586172
```

Adjusted temperature value

```
In [20]: #Construct a dataframe for mu of stations
         d = {'ID': ID, 'station_mu': mu}
         mu_df = pd.DataFrame(data=d)
         mu_df.head()

Out[20]:             ID  station_mu
         0  USC00043244    3.462006
         1  USC00044997   -0.042474
         2  USC00049001   -1.087858
         3  USR0000CCLV    6.032373
         4  USR0000CLVR    1.280165

In [21]: station_adjusted_east = stations_weather.merge(mu_df, on='ID', how='left')
         station_adjusted_east.tail()

Out[21]:        Unnamed: 0_x  Unnamed: 0.1           ID  LATITUDE  LONGITUDE  \
         85553            29          2657  USW00023285   37.6928  -121.8144
         85554            29          2657  USW00023285   37.6928  -121.8144
         85555            29          2657  USW00023285   37.6928  -121.8144
         85556            29          2657  USW00023285   37.6928  -121.8144
         85557            29          2657  USW00023285   37.6928  -121.8144

                ELEVATION STATE               NAME  GSN FLAG HCN/CRN FLAG     ...       \
         85553      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     ...
         85554      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     ...
         85555      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     ...
         85556      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     ...
         85557      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     ...

                 INVDIST  Unnamed: 0_y  YEARMONTHDAY  ELEMENT DATA VALUE  M-FLAG  \
         85553  0.143148    36575687.0    20091229.0     PRCP         3.0     NaN
         85554  0.143148    36678341.0    20091230.0     TMAX        59.0     NaN
         85555  0.143148    36678343.0    20091230.0     PRCP         0.0       T
```

```
85556  0.143148    36780952.0    20091231.0     TMAX       59.0     NaN
85557  0.143148    36780954.0    20091231.0     PRCP        0.0     NaN

      Q-FLAG S-FLAG OBS-TIME  station_mu
85553    NaN      0   2400.0    1.122116
85554    NaN      0   2400.0    1.122116
85555    NaN      0   2400.0    1.122116
85556    NaN      0   2400.0    1.122116
85557    NaN      0   2400.0    1.122116

[5 rows x 21 columns]
```

```
In [22]: # adjust tmax
         station_adjusted_east['DATA VALUE'] = np.where(station_adjusted_east['ELEMENT'] == 'T
                                                        station_adjusted_east['DATA VALUE']+ statio
                                                        station_adjusted_east['DATA VALUE'])
```

```
In [23]: # convert precipitation
         station_adjusted_east['DATA VALUE'] = np.where(station_adjusted_east['ELEMENT'] == 'P
                                                        (station_adjusted_east['DATA VALUE']/10),
                                                        station_adjusted_east['DATA VALUE'])
```

```
In [24]: station_adjusted_east.tail()
```

```
Out[24]:       Unnamed: 0_x  Unnamed: 0.1           ID  LATITUDE  LONGITUDE  \
         85553            29          2657  USW00023285   37.6928  -121.8144
         85554            29          2657  USW00023285   37.6928  -121.8144
         85555            29          2657  USW00023285   37.6928  -121.8144
         85556            29          2657  USW00023285   37.6928  -121.8144
         85557            29          2657  USW00023285   37.6928  -121.8144

                ELEVATION STATE               NAME  GSN FLAG HCN/CRN FLAG     ...       \
         85553      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     ...
         85554      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     ...
         85555      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     ...
         85556      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     ...
         85557      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     ...

                 INVDIST  Unnamed: 0_y  YEARMONTHDAY  ELEMENT DATA VALUE  M-FLAG  \
         85553  0.143148    36575687.0    20091229.0     PRCP   0.300000     NaN
         85554  0.143148    36678341.0    20091230.0     TMAX  56.737018     NaN
         85555  0.143148    36678343.0    20091230.0     PRCP   0.000000       T
         85556  0.143148    36780952.0    20091231.0     TMAX  56.737018     NaN
         85557  0.143148    36780954.0    20091231.0     PRCP   0.000000     NaN

               Q-FLAG S-FLAG OBS-TIME  station_mu
         85553    NaN      0   2400.0    1.122116
         85554    NaN      0   2400.0    1.122116
         85555    NaN      0   2400.0    1.122116
         85556    NaN      0   2400.0    1.122116
         85557    NaN      0   2400.0    1.122116

         [5 rows x 21 columns]
```

bias adjustment for the Prcp

```
In [25]: stations_prcp = stations_weather[stations_weather.ELEMENT=='PRCP']

In [26]: stations_prcp.tail()

Out[26]:        Unnamed: 0_x  Unnamed: 0.1           ID  LATITUDE  LONGITUDE  \
         85566            29          2657  USW00023285   37.6928  -121.8144
         85568            29          2657  USW00023285   37.6928  -121.8144
         85570            29          2657  USW00023285   37.6928  -121.8144
         85572            29          2657  USW00023285   37.6928  -121.8144
         85574            29          2657  USW00023285   37.6928  -121.8144

                ELEVATION STATE               NAME  GSN FLAG HCN/CRN FLAG  WMO ID  \
         85566      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     NaN
         85568      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     NaN
         85570      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     NaN
         85572      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     NaN
         85574      119.8    CA  LIVERMORE MUNI AP       NaN          NaN     NaN

                 INVDIST  Unnamed: 0_y  YEARMONTHDAY ELEMENT  DATA VALUE M-FLAG Q-FLAG  \
         85566  0.143148    36373607.0    20091227.0    PRCP        25.0    NaN    NaN
         85568  0.143148    36473956.0    20091228.0    PRCP        18.0    NaN    NaN
         85570  0.143148    36575687.0    20091229.0    PRCP         3.0    NaN    NaN
         85572  0.143148    36678343.0    20091230.0    PRCP         0.0      T    NaN
         85574  0.143148    36780954.0    20091231.0    PRCP         0.0    NaN    NaN

               S-FLAG  OBS-TIME
         85566       0    2400.0
         85568       0    2400.0
         85570       0    2400.0
         85572       0    2400.0
         85574       0    2400.0

In [27]: import random
         # set seed
         np.random.seed(101)

         ID = stations_prcp.ID.unique()
         # Create the starting mu
         mu = np.zeros(len(ID))
         mu_old = np.zeros(len(ID))

         while abs(mu_old-mu).any() > 0.01 or mu_old.all() == 0:
```

```
    # update mu
    mu_old = mu
    # get reference station
    ref_idx = np.random.choice(np.arange(len(ID)),1)
    ref = ID[ref_idx][0]
    station_i = stations_prcp[stations_prcp.ID == ref]

    # other station
    for j in ID:
        station_idx = np.arange(len(ID))[j == ID]
        station_j = stations_prcp[stations_prcp.ID == j]
        # get the date that both stations reported
        shared_date = station_i[['YEARMONTHDAY']].merge(station_j[['YEARMONTHDAY']],
        n = len(shared_date)

        # if there are shared dates
        if n>0:
            prcp_i = 0
            for day in shared_date.YEARMONTHDAY:
                prcp_i = prcp_i + float(station_i[station_i.YEARMONTHDAY==day]['DATA

            prcp_j = 0
            for day in shared_date.YEARMONTHDAY:
                prcp_j = prcp_j + float(station_j[station_j.YEARMONTHDAY==day]['DATA

            mu[station_idx] = mu_old[station_idx]+float(prcp_i+mu_old[ref_idx]*n - pr
        # if there's no shared date
        else:
            mu[station_idx] = mu_old[station_idx]
```

```
In [28]: mu
```

```
Out[28]: array([ -5.6893477 , -26.13431924,   0.72261292,  -3.21957172,
                  2.76907655,  -2.23792284,  -0.63431924,  -2.19957723])
```

```
In [29]: w = stations_prcp.groupby('ID').mean().INVDIST
```

```
In [30]: C = sum(w * mu)/sum(w)
         C
```

```
Out[30]: -3.8054063454983824
```

```
In [31]: #Construct a dataframe for mu of stations
         d = {'ID': ID, 'station_prcp_mu': mu}
         prcp_mu_df = pd.DataFrame(data=d)
         prcp_mu_df.head()
```

```
Out[31]:             ID  station_prcp_mu
         0  US1CAAL0004        -5.689348
```

```
        1  US1CASC0027       -26.134319
        2  US1CASJ0007         0.722613
        3  USC00043244        -3.219572
        4  USC00044508         2.769077
```

```
In [32]: station_adjusted_prcp = stations_weather.merge(prcp_mu_df, on='ID', how='left')
         station_adjusted_prcp.tail()
```

```
Out[32]:        Unnamed: 0_x  Unnamed: 0.1           ID  LATITUDE  LONGITUDE  \
         85553            29          2657  USW00023285   37.6928  -121.8144
         85554            29          2657  USW00023285   37.6928  -121.8144
         85555            29          2657  USW00023285   37.6928  -121.8144
         85556            29          2657  USW00023285   37.6928  -121.8144
         85557            29          2657  USW00023285   37.6928  -121.8144

                ELEVATION STATE               NAME  GSN FLAG HCN/CRN FLAG  \
         85553      119.8    CA  LIVERMORE MUNI AP       NaN          NaN
         85554      119.8    CA  LIVERMORE MUNI AP       NaN          NaN
         85555      119.8    CA  LIVERMORE MUNI AP       NaN          NaN
         85556      119.8    CA  LIVERMORE MUNI AP       NaN          NaN
         85557      119.8    CA  LIVERMORE MUNI AP       NaN          NaN

                     ...          INVDIST  Unnamed: 0_y  YEARMONTHDAY  ELEMENT  \
         85553       ...         0.143148    36575687.0    20091229.0     PRCP
         85554       ...         0.143148    36678341.0    20091230.0     TMAX
         85555       ...         0.143148    36678343.0    20091230.0     PRCP
         85556       ...         0.143148    36780952.0    20091231.0     TMAX
         85557       ...         0.143148    36780954.0    20091231.0     PRCP

               DATA VALUE  M-FLAG Q-FLAG S-FLAG OBS-TIME  station_prcp_mu
         85553        3.0     NaN    NaN      0   2400.0        -2.199577
         85554       59.0     NaN    NaN      0   2400.0        -2.199577
         85555        0.0       T    NaN      0   2400.0        -2.199577
         85556       59.0     NaN    NaN      0   2400.0        -2.199577
         85557        0.0     NaN    NaN      0   2400.0        -2.199577

         [5 rows x 21 columns]
```

```
In [33]: station_adjusted_east['station_prep_mu']=station_adjusted_prcp['station_prcp_mu']
```

```
In [34]: # adjust and convert precipitation
         station_adjusted_east['DATA VALUE'] = np.where(station_adjusted_east['ELEMENT'] == 'P
                                                  (station_adjusted_east['DATA VALUE']+ statio
                                                  station_adjusted_east['DATA VALUE'])
```

```
In [35]: station_adjusted_east.tail()
```

```
Out[35]:        Unnamed: 0_x  Unnamed: 0.1           ID  LATITUDE  LONGITUDE  \
         85553            29          2657  USW00023285   37.6928  -121.8144
```

```
85554            29          2657  USW00023285   37.6928  -121.8144
85555            29          2657  USW00023285   37.6928  -121.8144
85556            29          2657  USW00023285   37.6928  -121.8144
85557            29          2657  USW00023285   37.6928  -121.8144

       ELEVATION STATE               NAME  GSN FLAG HCN/CRN FLAG  \
85553      119.8    CA  LIVERMORE MUNI AP       NaN          NaN
85554      119.8    CA  LIVERMORE MUNI AP       NaN          NaN
85555      119.8    CA  LIVERMORE MUNI AP       NaN          NaN
85556      119.8    CA  LIVERMORE MUNI AP       NaN          NaN
85557      119.8    CA  LIVERMORE MUNI AP       NaN          NaN

            ...         Unnamed: 0_y  YEARMONTHDAY  ELEMENT  DATA VALUE  \
85553       ...           36575687.0    20091229.0     PRCP    0.190583
85554       ...           36678341.0    20091230.0     TMAX   56.737018
85555       ...           36678343.0    20091230.0     PRCP    0.160583
85556       ...           36780952.0    20091231.0     TMAX   56.737018
85557       ...           36780954.0    20091231.0     PRCP    0.160583

      M-FLAG  Q-FLAG S-FLAG OBS-TIME station_mu  station_prep_mu
85553    NaN     NaN      0   2400.0   1.122116        -2.199577
85554    NaN     NaN      0   2400.0   1.122116        -2.199577
85555      T     NaN      0   2400.0   1.122116        -2.199577
85556    NaN     NaN      0   2400.0   1.122116        -2.199577
85557    NaN     NaN      0   2400.0   1.122116        -2.199577

[5 rows x 22 columns]
```

```
In [36]: station_adjusted_east.to_csv("../group_assignment4/Intermediate_data/station_adjusted_
```

0.0.3 Please refer to above step for how the dataset is generated.

```
In [37]: station_adjusted_east = pd.read_csv("../group_assignment4/Intermediate_data/station_ad
         station_adjusted_east.head()
```

```
/anaconda3/lib/python3.6/site-packages/IPython/core/interactiveshell.py:3020: DtypeWarning: Col
  interactivity=interactivity, compiler=compiler, result=result)
```

```
Out[37]:    Unnamed: 0  Unnamed: 0_x  Unnamed: 0.1           ID  LATITUDE  LONGITUDE  \
         0           0             0             3  US1CAAL0004   37.6483  -121.8745
         1           1             0             3  US1CAAL0004   37.6483  -121.8745
         2           2             0             3  US1CAAL0004   37.6483  -121.8745
         3           3             0             3  US1CAAL0004   37.6483  -121.8745
         4           4             0             3  US1CAAL0004   37.6483  -121.8745

            ELEVATION STATE                NAME  GSN FLAG       ...        Unnamed: 0_y  \
         0      107.0    CA  PLEASANTON 1.8 SSE       NaN       ...          17849812.0
         1      107.0    CA  PLEASANTON 1.8 SSE       NaN       ...          17949981.0
```

```
2      107.0    CA  PLEASANTON 1.8 SSE       NaN       ...         18051494.0
3      107.0    CA  PLEASANTON 1.8 SSE       NaN       ...         18153195.0
4      107.0    CA  PLEASANTON 1.8 SSE       NaN       ...         18253026.0

   YEARMONTHDAY  ELEMENT  DATA VALUE  M-FLAG Q-FLAG  S-FLAG OBS-TIME  \
0    20080701.0     PRCP   -0.188394     NaN    NaN       N      NaN
1    20080702.0     PRCP   -0.188394     NaN    NaN       N      NaN
2    20080703.0     PRCP   -0.188394     NaN    NaN       N      NaN
3    20080704.0     PRCP   -0.188394     NaN    NaN       N      NaN
4    20080705.0     PRCP   -0.188394     NaN    NaN       N      NaN

  station_mu station_prep_mu
0        NaN       -5.689348
1        NaN       -5.689348
2        NaN       -5.689348
3        NaN       -5.689348
4        NaN       -5.689348

[5 rows x 23 columns]
```

```
In [38]: station_adjusted_east.shape
```

```
Out[38]: (85558, 23)
```

```
In [39]: station_TMAX_east = station_adjusted_east[station_adjusted_east['ELEMENT'] == 'TMAX']
         station_PRCP_east = station_adjusted_east[station_adjusted_east['ELEMENT'] == 'PRCP']
         station_TMAX_east['wi_val'] = station_TMAX_east['INVDIST'] * station_TMAX_east['DATA V
         station_PRCP_east['wi_val'] = station_PRCP_east['INVDIST'] * station_PRCP_east['DATA V
```

```
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html
  This is separate from the ipykernel package so we can avoid doing imports until
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html
  after removing the cwd from sys.path.
```

```
In [40]: weighted_TMAX_east = station_TMAX_east.groupby('YEARMONTHDAY', as_index = False).agg(
         weighted_PRCP_east = station_PRCP_east.groupby('YEARMONTHDAY', as_index = False).agg(
         weighted_PRCP_east.tail()
```

```
Out[40]:        YEARMONTHDAY    wi_val
         10953    20091227.0  0.491856
         10954    20091228.0  0.137103
         10955    20091229.0  0.173879
         10956    20091230.0  0.118014
         10957    20091231.0  0.089301
```

```
In [41]: # Warning: Extremely long runtime!!!!!! Don't repeatedly run
         dict_date_TMAX_east = {}
         for i in station_TMAX_east['YEARMONTHDAY']:
             only_i = station_TMAX_east[station_TMAX_east['YEARMONTHDAY'] == i]
             dict_date_TMAX_east[i] = sum(only_i['wi_val'])/sum(only_i['INVDIST'].unique())
```

```
In [42]: #sanity check
         series_date_TMAX_east = pd.Series(data = dict_date_TMAX_east)
         series_date_TMAX_east.describe()
```

```
Out[42]: count    10954.000000
         mean        69.835278
         std         13.843576
         min         28.878357
         25%         58.497655
         50%         68.871849
         75%         80.563947
         max        108.447896
         dtype: float64
```

```
In [43]: series_date_TMAX_east.head()
```

```
Out[43]: 19960601.0    88.730105
         19960602.0    96.781007
         19960603.0    95.283871
         19960604.0    88.685542
         19960605.0    95.541548
         dtype: float64
```

```
In [44]: # Warning: Extremely long runtime!!!!!! Don't repeatedly run
         dict_date_PRCP_east = {}
         for i in station_PRCP_east['YEARMONTHDAY']:
             only_i = station_PRCP_east[station_PRCP_east['YEARMONTHDAY'] == i]
             dict_date_PRCP_east[i] = sum(only_i['wi_val'])/sum(only_i['INVDIST'].unique())
```

```
In [45]: #sanity check, NorCal is dry so this makes sense
         series_date_PRCP_east = pd.Series(data = dict_date_PRCP_east)
         series_date_PRCP_east.describe()
```

```
Out[45]: count    10958.000000
         mean         0.309173
         std          0.364971
         min         -0.067553
         25%          0.178843
         50%           0.214986
         75%           0.299419
         max           5.418305
         dtype: float64
```

```
In [46]: series_date_PRCP_east.head()
```

```
Out[46]: 20080701.0    0.096698
         20080702.0    0.096698
         20080703.0    0.096698
         20080704.0    0.096698
         20080705.0    0.096698
         dtype: float64
```

0.0.4 Put bias-adjusted temperature into bins

```
In [47]: import math
         #define bins
         TMAX_df_east = pd.DataFrame({'DATE':series_date_TMAX_east.index, 'TMAX':series_date_T
         temp_bins = [-math.inf,10,20,30,40,50,60,70,80,90,100,math.inf]
         group_names_temp = ['<10F','10-19F','20-29F','30-39F','40-49F','50-59F','60-69F','70-
         TMAX_df_east['temp_bins'] = pd.cut(TMAX_df_east['TMAX'], temp_bins, labels = group_na
         TMAX_df_east.tail()
```

```
Out[47]:              DATE       TMAX temp_bins
         10949  19941126.0  50.528402    50-59F
         10950  19941127.0  48.179049    40-49F
         10951  19941128.0  48.284221    40-49F
         10952  19941129.0  51.784204    50-59F
         10953  19941130.0  54.934867    50-59F
```

```
In [48]: #Put precipitation into bins
         prcp_bins = [-math.inf,0.000001,5,15,30,math.inf]
         group_names_prcp = ['0mm','1-4mm','5-14mm','15-29mm','>30mm']
         PRCP_df_east = pd.DataFrame({'DATE':series_date_PRCP_east.index, 'PRCP':series_date_P
         PRCP_df_east['prcp_bins'] = pd.cut(PRCP_df_east['PRCP'], prcp_bins, labels = group_na
         PRCP_df_east.tail()
```

```
Out[48]:              DATE      PRCP prcp_bins
         10953  19941126.0  0.827109     1-4mm
         10954  19941127.0  0.317109     1-4mm
         10955  19941128.0  0.317109     1-4mm
         10956  19941129.0  0.317109     1-4mm
         10957  19941130.0  0.317109     1-4mm
```

```
In [49]: TMAX_df_east['YearMonth'] = TMAX_df_east['DATE'].astype(str).str[:6]
         TMAX_data_east = TMAX_df_east.pivot_table(index=['temp_bins'],
                                       columns='YearMonth',
                                       values='TMAX',
                                       fill_value = 0,
                                       aggfunc='count').unstack().to_frame().reset_index()
```

```
In [50]: TMAX_data_east.head()

Out[50]:   YearMonth temp_bins   0
         0    198001    20-29F   0
         1    198001    30-39F   0
         2    198001    40-49F   8
         3    198001    50-59F  20
         4    198001    60-69F   3
```

```
In [51]: PRCP_df_east['YearMonth'] = PRCP_df_east['DATE'].astype(str).str[:6]
         PRCP_data_east = PRCP_df_east.pivot_table(index=['prcp_bins'],
                                      columns='YearMonth',
                                      values='PRCP',
                                      fill_value = 0,
                                      aggfunc='count').unstack().to_frame().reset_index()
```

```
In [52]: PRCP_data_east.head()

Out[52]:   YearMonth prcp_bins   0
         0    198001       0mm   0
         1    198001     1-4mm  31
         2    198001    5-14mm   0
         3    198002       0mm   0
         4    198002     1-4mm  29
```

```
In [53]: TMAX_data_east.to_csv('TMAX_data_east.csv')
         PRCP_data_east.to_csv('PRCP_data_east.csv')
```