Repeat_Assignment_3_west

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	S-FLAG	OBS-TIME
0	91	USS0019L03S	19800101	PRCP	0	NaN	NaN	T	NaN
1	465	USC00048446	19800101	PRCP	0	NaN	NaN	0	800.0
2	490	USC00040115	19800101	PRCP	0	NaN	NaN	0	800.0
3	735	USW00023110	19800101	TMAX	172	NaN	NaN	X	NaN
4	736	USW00023110	19800101	TMIN	89	NaN	NaN	X	NaN

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

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

Out[4]: (19800101, 20091231)

Weather data already in relevant time periods (1980-2009).

In [5]:
```
stations_within_10miles_west_alameda = pd.read_csv("../group_assignment4/Intermediate_
```

In [6]:
```
len(stations_within_10miles_west_alameda)
```

Out[6]: 76

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

In [8]:
```
#Merge the 'stations_within_10miles_west_alameda' with 'weather' on station's ID.
#Get rid of all stations not appearing in 'stations_within_10miles_west_alameda'
merged = stations_within_10miles_west_alameda.merge(weather_new, on='ID', how='left')
merged.head()
```

Out[8]:

	Unnamed: 0_x	Unnamed: 0.1	ID	LATITUDE	LONGITUDE	ELEVATION	\
0	0	0	US1CAAL0001	37.8123	-122.216	113.4	
1	0	0	US1CAAL0001	37.8123	-122.216	113.4	
2	0	0	US1CAAL0001	37.8123	-122.216	113.4	
3	0	0	US1CAAL0001	37.8123	-122.216	113.4	
4	0	0	US1CAAL0001	37.8123	-122.216	113.4	

	STATE	NAME	GSN FLAG	HCN/CRN FLAG	WMO ID	INVDIST	\
0	CA	PIEDMONT 1.0 SE	NaN	NaN	NaN	0.130528	
1	CA	PIEDMONT 1.0 SE	NaN	NaN	NaN	0.130528	
2	CA	PIEDMONT 1.0 SE	NaN	NaN	NaN	0.130528	
3	CA	PIEDMONT 1.0 SE	NaN	NaN	NaN	0.130528	
4	CA	PIEDMONT 1.0 SE	NaN	NaN	NaN	0.130528	

	Unnamed: 0_y	YEARMONTHDAY	ELEMENT	DATA VALUE	M-FLAG	Q-FLAG	S-FLAG	\
0	27876749.0	20081007.0	PRCP	0.0	NaN	NaN	N	
1	27978273.0	20081008.0	PRCP	0.0	NaN	NaN	N	
2	28079583.0	20081009.0	PRCP	0.0	NaN	NaN	N	
3	28179978.0	20081010.0	PRCP	0.0	NaN	NaN	N	
4	28279413.0	20081011.0	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 west 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        210537
HCN/CRN FLAG    191612
WMO ID          179863
INVDIST              0
Unnamed: 0_y        42
YEARMONTHDAY        42
ELEMENT             42
DATA VALUE          42
M-FLAG          148296
Q-FLAG          209765
S-FLAG              42
OBS-TIME         63161
dtype: int64
```

There are 42 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 duplicate data:

```
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

In [12]: stations_weather.tail()

Out[12]:         Unnamed: 0_x  Unnamed: 0.1           ID  LATITUDE  LONGITUDE  \
         210532            75          2714  USW00093228   37.6542   -122.115
         210533            75          2714  USW00093228   37.6542   -122.115
         210534            75          2714  USW00093228   37.6542   -122.115
         210535            75          2714  USW00093228   37.6542   -122.115
         210536            75          2714  USW00093228   37.6542   -122.115

                 ELEVATION STATE                 NAME  GSN FLAG HCN/CRN FLAG   WMO ID  \
         210532       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN  72585.0
         210533       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN  72585.0
         210534       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN  72585.0
         210535       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN  72585.0
         210536       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN  72585.0

                  INVDIST  Unnamed: 0_y  YEARMONTHDAY ELEMENT  DATA VALUE M-FLAG  \
         210532  0.115491    36648671.0    20091229.0    PRCP        10.0    NaN
         210533  0.115491    36751411.0    20091230.0    TMAX       139.0    NaN
         210534  0.115491    36751413.0    20091230.0    PRCP         0.0      T
         210535  0.115491    36853346.0    20091231.0    TMAX       167.0    NaN
```

```
        210536  0.115491    36853348.0    20091231.0    PRCP         0.0    NaN

               Q-FLAG S-FLAG  OBS-TIME
        210532    NaN      0    2400.0
        210533    NaN      0    2400.0
        210534    NaN      0    2400.0
        210535    NaN      0    2400.0
        210536    NaN      0    2400.0
```

No duplicates

Convert 'Data Value' to correct unit:

```
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 + 3
                                                   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  \
         210532            75          2714  USW00093228   37.6542   -122.115
         210533            75          2714  USW00093228   37.6542   -122.115
         210534            75          2714  USW00093228   37.6542   -122.115
         210535            75          2714  USW00093228   37.6542   -122.115
         210536            75          2714  USW00093228   37.6542   -122.115

                 ELEVATION STATE                  NAME  GSN FLAG HCN/CRN FLAG   WMO ID  \
         210532       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN  72585.0
         210533       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN  72585.0
         210534       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN  72585.0
         210535       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN  72585.0
         210536       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN  72585.0

                  INVDIST  Unnamed: 0_y  YEARMONTHDAY ELEMENT  DATA VALUE M-FLAG  \
         210532  0.115491    36648671.0    20091229.0    PRCP       10.00    NaN
         210533  0.115491    36751411.0    20091230.0    TMAX       57.02    NaN
         210534  0.115491    36751413.0    20091230.0    PRCP        0.00      T
         210535  0.115491    36853346.0    20091231.0    TMAX       62.06    NaN
         210536  0.115491    36853348.0    20091231.0    PRCP        0.00    NaN
```

```
        Q-FLAG S-FLAG  OBS-TIME
210532     NaN      0    2400.0
210533     NaN      0    2400.0
210534     NaN      0    2400.0
210535     NaN      0    2400.0
210536     NaN      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']], c
        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 - ter
        # if there's no shared date
```

```
                else:
                    mu[station_idx] = mu_old[station_idx]
```

```
In [17]: mu
```

```
Out[17]: array([-2.18711192, -4.68155761, -3.71250249, -3.59665333, -1.76479692,
                -1.41395145, -6.3991625 , -5.85571902, -4.41693154, -2.29110703,
                -7.53886538, -0.09187452, -4.1022289 , -1.97586929,  1.24371679,
                -1.83921429, -0.25119838, -2.69485846, -7.30562926, -1.60065139])
```

county adjusted C

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

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

```
Out[19]: -3.218216138639562
```

Adjust the 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  USC00040693   -2.187112
         1  USC00043244   -4.681558
         2  USC00045915   -3.712502
         3  USC00046144   -3.596653
         4  USC00046336   -1.764797
```

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

```
Out[21]:         Unnamed: 0_x  Unnamed: 0.1           ID  LATITUDE  LONGITUDE  \
         210490            75          2714  USW00093228   37.6542   -122.115
         210491            75          2714  USW00093228   37.6542   -122.115
         210492            75          2714  USW00093228   37.6542   -122.115
         210493            75          2714  USW00093228   37.6542   -122.115
         210494            75          2714  USW00093228   37.6542   -122.115

                 ELEVATION STATE                 NAME  GSN FLAG HCN/CRN FLAG  \
         210490       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN
         210491       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN
         210492       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN
         210493       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN
         210494       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN
```

```
                ...       INVDIST  Unnamed: 0_y  YEARMONTHDAY  ELEMENT DATA VALUE  \
210490          ...      0.115491    36648671.0    20091229.0      PRCP       10.00
210491          ...      0.115491    36751411.0    20091230.0      TMAX       57.02
210492          ...      0.115491    36751413.0    20091230.0      PRCP        0.00
210493          ...      0.115491    36853346.0    20091231.0      TMAX       62.06
210494          ...      0.115491    36853348.0    20091231.0      PRCP        0.00

        M-FLAG Q-FLAG S-FLAG OBS-TIME  station_mu
210490     NaN    NaN      0   2400.0   -1.600651
210491     NaN    NaN      0   2400.0   -1.600651
210492       T    NaN      0   2400.0   -1.600651
210493     NaN    NaN      0   2400.0   -1.600651
210494     NaN    NaN      0   2400.0   -1.600651

[5 rows x 21 columns]
```

In [22]:
```
# adjust tmax
station_adjusted_west['DATA VALUE'] = np.where(station_adjusted_west['ELEMENT'] == 'TM
                                               station_adjusted_west['DATA VALUE']+ statio
                                               station_adjusted_west['DATA VALUE'])
```

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

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

Out[24]:
```
        Unnamed: 0_x  Unnamed: 0.1           ID  LATITUDE  LONGITUDE  \
210490            75          2714  USW00093228   37.6542   -122.115
210491            75          2714  USW00093228   37.6542   -122.115
210492            75          2714  USW00093228   37.6542   -122.115
210493            75          2714  USW00093228   37.6542   -122.115
210494            75          2714  USW00093228   37.6542   -122.115

        ELEVATION STATE                 NAME  GSN FLAG HCN/CRN FLAG  \
210490       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN
210491       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN
210492       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN
210493       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN
210494       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN

                ...       INVDIST  Unnamed: 0_y  YEARMONTHDAY  ELEMENT DATA VALUE  \
210490          ...      0.115491    36648671.0    20091229.0      PRCP    1.000000
210491          ...      0.115491    36751411.0    20091230.0      TMAX   58.637565
210492          ...      0.115491    36751413.0    20091230.0      PRCP    0.000000
210493          ...      0.115491    36853346.0    20091231.0      TMAX   63.677565
210494          ...      0.115491    36853348.0    20091231.0      PRCP    0.000000
```

```
        M-FLAG Q-FLAG S-FLAG OBS-TIME  station_mu
210490     NaN    NaN      0   2400.0   -1.600651
210491     NaN    NaN      0   2400.0   -1.600651
210492       T    NaN      0   2400.0   -1.600651
210493     NaN    NaN      0   2400.0   -1.600651
210494     NaN    NaN      0   2400.0   -1.600651

[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  \
210528            75          2714  USW00093228   37.6542   -122.115
210530            75          2714  USW00093228   37.6542   -122.115
210532            75          2714  USW00093228   37.6542   -122.115
210534            75          2714  USW00093228   37.6542   -122.115
210536            75          2714  USW00093228   37.6542   -122.115

        ELEVATION STATE                  NAME  GSN FLAG HCN/CRN FLAG   WMO ID  \
210528       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN  72585.0
210530       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN  72585.0
210532       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN  72585.0
210534       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN  72585.0
210536       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN  72585.0

         INVDIST  Unnamed: 0_y  YEARMONTHDAY ELEMENT  DATA VALUE M-FLAG  \
210528  0.115491    36444853.0    20091227.0    PRCP         3.0    NaN
210530  0.115491    36546191.0    20091228.0    PRCP         0.0      T
210532  0.115491    36648671.0    20091229.0    PRCP        10.0    NaN
210534  0.115491    36751413.0    20091230.0    PRCP         0.0      T
210536  0.115491    36853348.0    20091231.0    PRCP         0.0    NaN

       Q-FLAG S-FLAG  OBS-TIME
210528    NaN      0    2400.0
210530    NaN      0    2400.0
210532    NaN      0    2400.0
210534    NaN      0    2400.0
210536    NaN      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([ -76.1787234 ,  -61.44      ,  -27.05714286,  -44.60816327,
                 -34.2       ,  -59.86666667,  -94.2       ,  -88.6375    ,
                -102.4       ,  -72.15652174, -108.5       ,  -75.2       ,
                 -72.7       ,  -41.7       ,  -54.82      ,  -30.14      ,
                 -73.68      ,  -16.2       ,  -39.74      ,  -83.4972973 ,
                 -25.43356401,  -45.64      ,  -16.11836735,  -19.60934821,
                 -13.8       ,  -13.34      ])
```

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

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

```
Out[30]: -51.404646743200956

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  US1CAAL0001       -76.178723
         1  US1CAAL0002       -61.440000
         2  US1CAAL0003       -27.057143
         3  US1CAAL0004       -44.608163
         4  US1CAAL0016       -34.200000

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  \
         210490            75          2714  USW00093228   37.6542   -122.115
         210491            75          2714  USW00093228   37.6542   -122.115
         210492            75          2714  USW00093228   37.6542   -122.115
         210493            75          2714  USW00093228   37.6542   -122.115
         210494            75          2714  USW00093228   37.6542   -122.115

                 ELEVATION STATE                 NAME  GSN FLAG HCN/CRN FLAG  \
         210490       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN
         210491       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN
         210492       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN
         210493       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN
         210494       13.1    CA  HAYWARD AIR TERMINAL       NaN          NaN

                         ...          INVDIST  Unnamed: 0_y  YEARMONTHDAY  ELEMENT  \
         210490          ...         0.115491    36648671.0    20091229.0     PRCP
         210491          ...         0.115491    36751411.0    20091230.0     TMAX
         210492          ...         0.115491    36751413.0    20091230.0     PRCP
         210493          ...         0.115491    36853346.0    20091231.0     TMAX
         210494          ...         0.115491    36853348.0    20091231.0     PRCP

                DATA VALUE  M-FLAG Q-FLAG S-FLAG OBS-TIME  station_prcp_mu
         210490      10.00     NaN    NaN      0   2400.0           -13.34
         210491      57.02     NaN    NaN      0   2400.0           -13.34
         210492       0.00       T    NaN      0   2400.0           -13.34
         210493      62.06     NaN    NaN      0   2400.0           -13.34
         210494       0.00     NaN    NaN      0   2400.0           -13.34

         [5 rows x 21 columns]

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

In [34]:
```
# adjust and convert precipitation
station_adjusted_west['DATA VALUE'] = np.where(station_adjusted_west['ELEMENT'] == 'PR
                                               (station_adjusted_west['DATA VALUE']+ static
                                               station_adjusted_west['DATA VALUE'])
```

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

Out[35]:

	Unnamed: 0_x	Unnamed: 0.1	ID	LATITUDE	LONGITUDE	\
210490	75	2714	USW00093228	37.6542	-122.115	
210491	75	2714	USW00093228	37.6542	-122.115	
210492	75	2714	USW00093228	37.6542	-122.115	
210493	75	2714	USW00093228	37.6542	-122.115	
210494	75	2714	USW00093228	37.6542	-122.115	

	ELEVATION	STATE	NAME	GSN FLAG	HCN/CRN FLAG	\
210490	13.1	CA	HAYWARD AIR TERMINAL	NaN	NaN	
210491	13.1	CA	HAYWARD AIR TERMINAL	NaN	NaN	
210492	13.1	CA	HAYWARD AIR TERMINAL	NaN	NaN	
210493	13.1	CA	HAYWARD AIR TERMINAL	NaN	NaN	
210494	13.1	CA	HAYWARD AIR TERMINAL	NaN	NaN	

	...	Unnamed: 0_y	YEARMONTHDAY	ELEMENT	DATA VALUE	\
210490	...	36648671.0	20091229.0	PRCP	3.906465	
210491	...	36751411.0	20091230.0	TMAX	58.637565	
210492	...	36751413.0	20091230.0	PRCP	3.806465	
210493	...	36853346.0	20091231.0	TMAX	63.677565	
210494	...	36853348.0	20091231.0	PRCP	3.806465	

	M-FLAG	Q-FLAG	S-FLAG	OBS-TIME	station_mu	station_prep_mu
210490	NaN	NaN	0	2400.0	-1.600651	-13.34
210491	NaN	NaN	0	2400.0	-1.600651	-13.34
210492	T	NaN	0	2400.0	-1.600651	-13.34
210493	NaN	NaN	0	2400.0	-1.600651	-13.34
210494	NaN	NaN	0	2400.0	-1.600651	-13.34

[5 rows x 22 columns]

In [36]:
```
station_adjusted_west.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_west = pd.read_csv("../group_assignment4/Intermediate_data/station_ad
station_adjusted_west.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	0	US1CAAL0001	37.8123	-122.216	

```
1           1            0            0  US1CAAL0001   37.8123   -122.216
2           2            0            0  US1CAAL0001   37.8123   -122.216
3           3            0            0  US1CAAL0001   37.8123   -122.216
4           4            0            0  US1CAAL0001   37.8123   -122.216

   ELEVATION STATE             NAME  GSN FLAG       ...       Unnamed: 0_y  \
0      113.4    CA  PIEDMONT 1.0 SE       NaN       ...         27876749.0
1      113.4    CA  PIEDMONT 1.0 SE       NaN       ...         27978273.0
2      113.4    CA  PIEDMONT 1.0 SE       NaN       ...         28079583.0
3      113.4    CA  PIEDMONT 1.0 SE       NaN       ...         28179978.0
4      113.4    CA  PIEDMONT 1.0 SE       NaN       ...         28279413.0

   YEARMONTHDAY  ELEMENT  DATA VALUE  M-FLAG Q-FLAG  S-FLAG OBS-TIME  \
0    20081007.0     PRCP   -2.477408     NaN    NaN       N      NaN
1    20081008.0     PRCP   -2.477408     NaN    NaN       N      NaN
2    20081009.0     PRCP   -2.477408     NaN    NaN       N      NaN
3    20081010.0     PRCP   -2.477408     NaN    NaN       N      NaN
4    20081011.0     PRCP   -2.477408     NaN    NaN       N      NaN

  station_mu station_prep_mu
0        NaN      -76.178723
1        NaN      -76.178723
2        NaN      -76.178723
3        NaN      -76.178723
4        NaN      -76.178723

[5 rows x 23 columns]
```

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

```
Out[38]: (210495, 23)
```

```
In [39]: station_TMAX_west = station_adjusted_west[station_adjusted_west['ELEMENT'] == 'TMAX']
         station_PRCP_west = station_adjusted_west[station_adjusted_west['ELEMENT'] == 'PRCP']
         station_TMAX_west['wi_val'] = station_TMAX_west['INVDIST'] * station_TMAX_west['DATA
         station_PRCP_west['wi_val'] = station_PRCP_west['INVDIST'] * station_PRCP_west['DATA
```

```
/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_west = station_TMAX_west.groupby('YEARMONTHDAY', as_index = False).agg(
         weighted_PRCP_west = station_PRCP_west.groupby('YEARMONTHDAY', as_index = False).agg(
         weighted_PRCP_west.tail()
```

```
Out[40]:       YEARMONTHDAY    wi_val
         10953    20091227.0  1.257676
         10954    20091228.0  0.116013
         10955    20091229.0  0.521392
         10956    20091230.0 -0.251306
         10957    20091231.0 -0.076925
```

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

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

```
Out[42]: count    10958.000000
         mean        67.750418
         std          9.920834
         min         36.743499
         25%         59.951389
         50%         67.328337
         75%         74.478568
         max        102.557214
         dtype: float64
```

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

```
Out[43]: 19800101.0    57.677027
         19800102.0    53.598224
         19800103.0    55.736733
         19800104.0    54.838987
         19800105.0    55.297114
         dtype: float64
```

```
In [44]: # Warning: Extremely long runtime!!!!!! Don't repeatedly run
         dict_date_PRCP_west = {}
         for i in station_PRCP_west['YEARMONTHDAY']:
             only_i = station_PRCP_west[station_PRCP_west['YEARMONTHDAY'] == i]
             dict_date_PRCP_west[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_west = pd.Series(data = dict_date_PRCP_west)
         series_date_PRCP_west.describe()

Out[45]: count    10958.000000
         mean         1.043671
         std          0.645362
         min         -0.118448
         25%          0.496040
         50%          0.875073
         75%          1.403604
         max         10.859776
         dtype: float64

In [46]: series_date_PRCP_west.head()

Out[46]: 20081007.0    0.921491
         20081008.0    0.923847
         20081009.0    0.921491
         20081010.0    0.921491
         20081011.0    0.921491
         dtype: float64

In [47]: #Put bias-adjusted temperature into bins
         import math
         #define bins
         TMAX_df_west = pd.DataFrame({'DATE':series_date_TMAX_west.index, 'TMAX':series_date_TM
         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','70-7
         TMAX_df_west['temp_bins'] = pd.cut(TMAX_df_west['TMAX'], temp_bins, labels = group_nam
         TMAX_df_west.tail()

Out[47]:             DATE       TMAX temp_bins
         10953  19881222.0  52.198033    50-59F
         10954  19881223.0  59.313743    50-59F
         10955  19881229.0  47.342599    40-49F
         10956  19890106.0  50.292007    50-59F
         10957  19910201.0  60.453999    60-69F

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_west = pd.DataFrame({'DATE':series_date_PRCP_west.index, 'PRCP':series_date_PR
         PRCP_df_west['prcp_bins'] = pd.cut(PRCP_df_west['PRCP'], prcp_bins, labels = group_nam
         PRCP_df_west.tail()

Out[48]:             DATE      PRCP prcp_bins
         10953  19921224.0  1.098822     1-4mm
         10954  19921225.0  1.098822     1-4mm
```

```
10955  19921226.0  1.098822     1-4mm
10956  19921227.0  1.098822     1-4mm
10957  19921231.0  1.119603     1-4mm
```

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

```
In [50]: PRCP_df_west['YearMonth'] = PRCP_df_west['DATE'].astype(str).str[:6]
         PRCP_data_west = PRCP_df_west.pivot_table(index=['prcp_bins'],
                                    columns='YearMonth',
                                    values='PRCP',
                                    fill_value = 0,
                                    aggfunc='count').unstack().to_frame().reset_index()
```

```
In [51]: TMAX_data_west.head()
```

```
Out[51]:   YearMonth temp_bins   0
         0    198001    30-39F   0
         1    198001    40-49F   0
         2    198001    50-59F  25
         3    198001    60-69F   6
         4    198001    70-79F   0
```

```
In [52]: PRCP_data_west.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  28
```

```
In [53]: TMAX_data_west.to_csv("TMAX_data_west.csv")
         PRCP_data_west.to_csv("PRCP_data_west.csv")
```