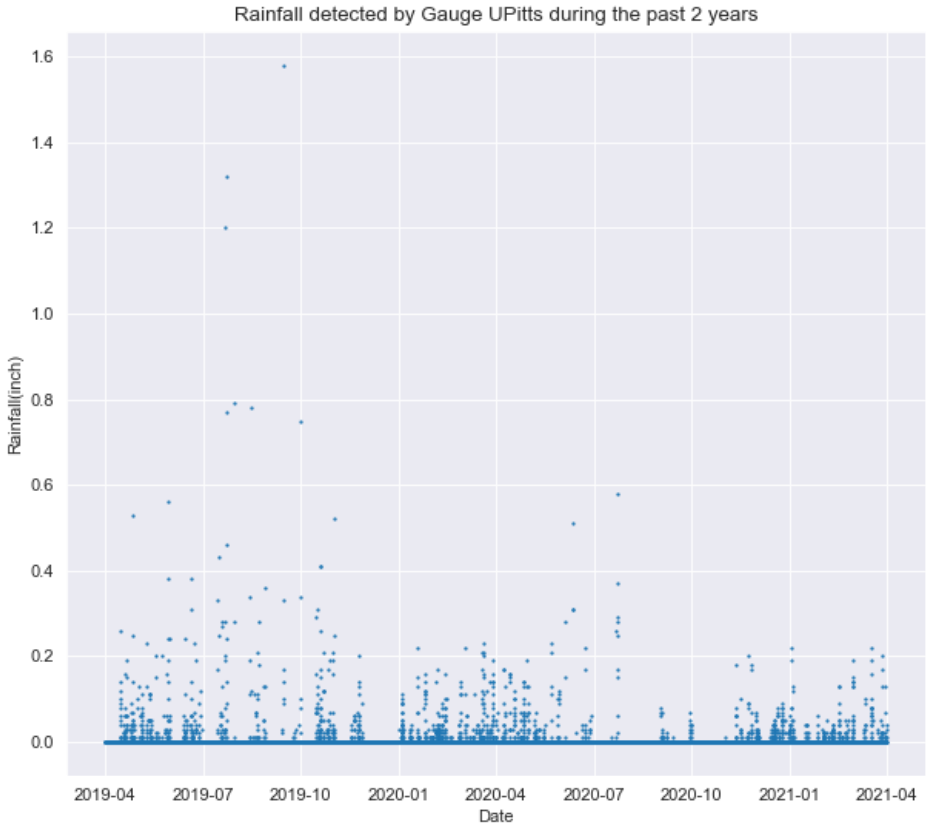
1. Warming up to Time-Series Data Again (15%)

a.

```
In [154]: result['Date'] = result['Date'].astype('datetime64')
          result = result.sort_values(by="Date")
          result = result.set_index("Date")
          result_hourly = result.resample("1H").sum()

In [155]: import matplotlib.pyplot as plt
          from matplotlib.pyplot import figure
          %matplotlib inline

          fig = figure(figsize = (9,8),dpi = 80)
          x = np.array(result_hourly.index)
          y = result_hourly['University of Pittsburgh']
          y[y<0]=0
          y = np.array(y)
          plt.scatter(x,y,s=1)
          plt.xlabel("Date")
          plt.ylabel("Rainfall(inch)")
          plt.title("Rainfall detected by Gauge UPitts during the past 2 years")
          plt.show()
          fig.savefig("rainfall.png")
```



b.

To clean the data, I would first remove all the negative data points and Null data points. Then I would fit the data using learning algorithms. The model would then give me a prediction for every time instance. The prediction minus the real data would give me a residual vector. After I acquire the residual vector, I will calculate the mean and standard deviation of the residuals. Lastly, the outliers will be removed outliers using Chauvenet's criterion. I will repeat the process until no

more outliers are detected.

2. Water in the foundation: yikes! (60%)

(a)

```
In [575]: data_matrix = np.array(data[['Water Level(m)','Temperature(C)']].values,'float')
          beta_vector = np.ones(3651)
          alpha_vector = np.arange(0,3651,1)*144
          A_matrix = np.column_stack((alpha_vector,beta_vector,data_matrix))
          A_matrix.reshape(3651,4)
          pseudo_inverse_A_Matrix = np.linalg.pinv(A_matrix)

In [576]: y_vector = np.array(data['Strain(micro-strain)'].values,'float')

In [577]: W_Matrix = np.matmul(pseudo_inverse_A_Matrix,y_vector)

In [578]: print('Alpha=',W_Matrix[0],' micro-strain/min')
          print('Beta=',W_Matrix[1],'micro-strain')
          print('Gamma=',W_Matrix[3],'micro-strains/meter')
          print('Delta=',W_Matrix[2],'micro-strains/Celsius')
```

Alpha= -0.00042984263809532115 micro-strain/min

Beta= -439.50238120742614 micro-strain

Gamma= 1.4432743355313284 micro-strains/meter

Delta= -53.15723364050697 micro-strains/Celsius

(b)

```
In [579]: q_3000 = W_Matrix[0]*2999*144+W_Matrix[1]
          print('q_3000 =',q_3000, 'micro-strain')
```

q_3000 = -625.1325035247191 micro-strain

(c)

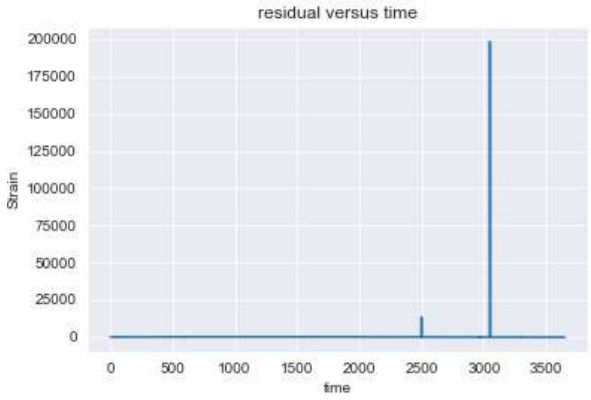
```
In [580]: q_5y = W_Matrix[0]*5*365*24*60+W_Matrix[1]
          print('q_5y=',q_5y, 'micro-strain')
```

q_5y= -1569.1288341219304 micro-strain

(d)

```
In [581]: # y is the prediction after linear regression
          y = np.matmul(A_matrix,W_Matrix)
          residual = y-data['Strain(micro-strain)']

In [589]: residual_plot = sns.lineplot(data = residual)
          sns.set_style("darkgrid")
          residual_plot.set_xlabel("minutes")
          residual_plot.set_ylabel("micro-strain")
          residual_plot.set_title("residual versus time")
          residual_plot.figure.savefig("residual versus time")
```



```
In [483]: print('The mean of the residual =',residual.mean())
          residual_mean = residual.mean()
          print('The std of the residual =',residual.std())
          residual_std = residual.std()
```

The mean of the residual = 3.5922829883142626e-12
The std of the residual = 3293.952960065999 (e)
(e)

```
In [669]: # iterate the process
          outlier_exist = True
          drop_index_total=[]
          while outlier_exist:
              number_of_rows = len(data)
              # build A_matrix for linear regression
              data_matrix = np.array(data[['Water Level(m)','Temperature(C)']].values,'float')
              beta_vector = np.ones(number_of_rows)
              alpha_vector = np.arange(0,number_of_rows,1)*144
              A_matrix = np.column_stack((alpha_vector,beta_vector,data_matrix))
              A_matrix.reshape(number_of_rows,4)
              # find the psuedo inverse of A_matrix
              pseudo_inverse_A_Matrix = np.linalg.pinv(A_matrix)
              # find the y vector
              y_vector = np.array(data['Strain(micro-strain)'].values,'float')
              # calculate the parameters for linear regression
              W_Matrix = np.matmul(pseudo_inverse_A_Matrix,y_vector)
              # predict y
              y = np.matmul(A_matrix,W_Matrix)
              # calculate residuals
              residual = y-data['Strain(micro-strain)']
              # mean and std of residuals
              print('The mean of the residual = ',residual.mean())
              residual_mean = residual.mean()
              print('The std of the residual = ',residual.std())
              residual_std = residual.std()
              # find the index with outlier datapoints and store it in drop_index[]
              drop_index = []
              threshold_high = residual_mean+3*residual_std
              threshold_low = residual_mean-3*residual_std
              for i in range(0,number_of_rows):
                  if i in drop_index_total:
                      continue
                  if residual[i] > thres_hold_high or residual[i] < threshold_low:
                      drop_index.append(i)
              #keep track of all the index dropped throught the iteration
              for i in range(0,len(drop_index)):
                  drop_index_total.append(drop_index[i])
                  print('The water level:',data['Water Level(m)'][drop_index[i]],
                        ',the Temperature:',data['Temperature(C)'][drop_index[i]],',the row is:',drop_index[i])
              #data set drop index
              data.drop(drop_index,inplace=True)
              print('-----------------------------------------------------')
              #if no more outliers than exit the iteration
              if not drop_index:
                  print('no more outliers')
                  outlier_exist = False
```

The mean of the residual = -1.6778227974623325e-12
The std of the residual = 3293.952960065999
The water level: -9.54688299960992, the Temperature: 4.17978837213146, the row is: 2499
The water level: -8.37139121446426, the Temperature: 2.10399519785036, the row is: 3049

The mean of the residual = 5.787398817541758e-14
The std of the residual = 14.009994054339375
The water level: -7.0630814881888, the Temperature: 0.455863021208531, the row is: 49
The water level: -8.43718097809429, the Temperature: 4.77281695916613, the row is: 199
The water level: -8.59082784081332, the Temperature: 9.05104789496236, the row

is: 349

The water level: -9.24008000788457, the Temperature: 13.4188135942759, the row is: 499

The water level: -9.19820277788709, the Temperature: 14.5260105408692, the row is: 599

The mean of the residual = -1.3106457010069391e-13

The std of the residual = 13.448810720019157

---

no more outliers

```
In [503]: print('Alpha=',W_Matrix[0],' micro-strain/min')
          print('Beta=',W_Matrix[1],'micro-strain')
          print('Gamma=',W_Matrix[3],'micro-strains/Celsius')
          print('Delta=',W_Matrix[2],'micro-strains/meter')
```

Alpha= -3.7920129517063086e-05 micro-strain/min

Beta= 13.226808599144258 micro-strain

Gamma= 1.5986416864931896 micro-strains/Celsius

Delta= -0.722926395653447 micro-strains/meter

(f)

```
In [550]: q_3000 = W_Matrix[0]*2000*144+W_Matrix[1]
          print('q_3000 =',q_3000, 'micro-strain')
          q_5y = W_Matrix[0]*5*365*24*60+W_Matrix[1]
          print('q_5y=',q_5y, 'micro-strain')
```

q_3000 = -3.15468735222435

q_5y= -66.51831369211841

(g)

```
In [555]: W_Matrix
          print('Alpha=',W_Matrix[0],' micro-strain/min')
          print('Beta=',W_Matrix[1],'micro-strain')
          print('Delta=',W_Matrix[2],'micro-strains/meter')
          q_3000 = W_Matrix[0]*2000*144+W_Matrix[1]
          print('q_3000 =',q_3000, 'micro-strain')
          q_5y = W_Matrix[0]*5*365*24*60+W_Matrix[1]
          print('q_5y=',q_5y, 'micro-strain')
```

Alpha= -8.03862856049973e-05 micro-strain/min

Beta= -39.338451803891246 micro-strain

Delta= -9.538011885057054 micro-strains/meter

q_3000 = -74.05375156012296 micro-strain

q_5y= -250.59361037382422 micro-strain

3. Wet Databases (15%)

4. Set Theory (10%)

The baseball players are a larger group.

$$\frac{1}{10} baseball\ players = \frac{1}{6} Dominicans$$

$$\frac{Baseball\ players}{Dominicans} = \frac{5}{3}$$