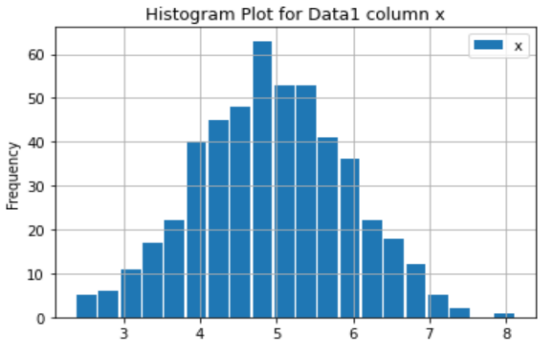
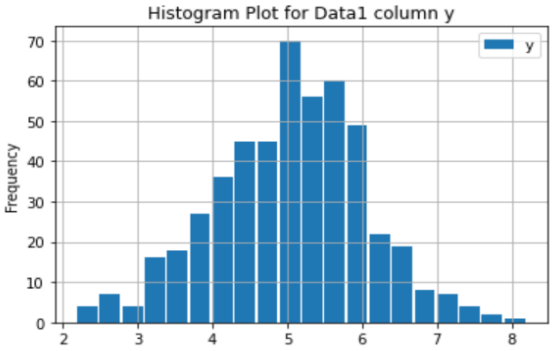
# CLL 788 Assignment 1 Report

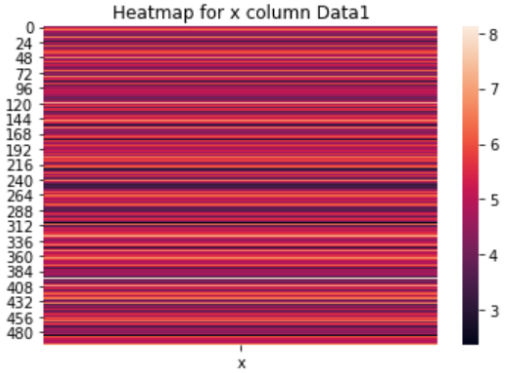
*Aditi Singh*
*2017CH10188*

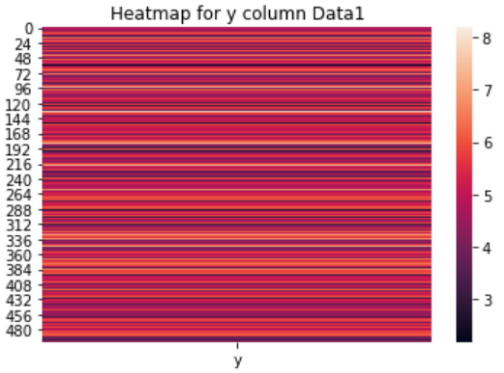
**Q1 Do data visualization using the data given**

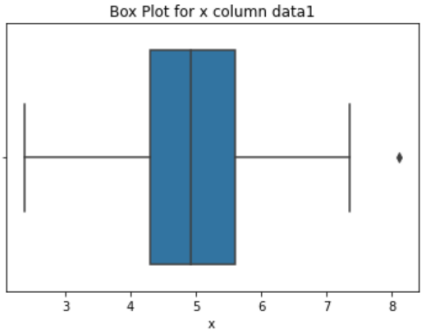
**a) Show scatter, histogram, heatmap, box plots (data_1).**

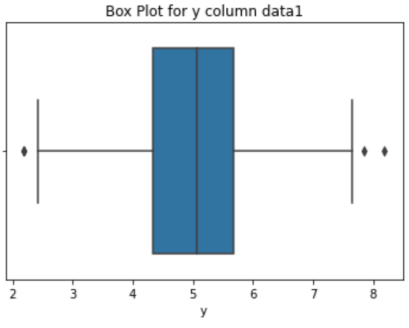



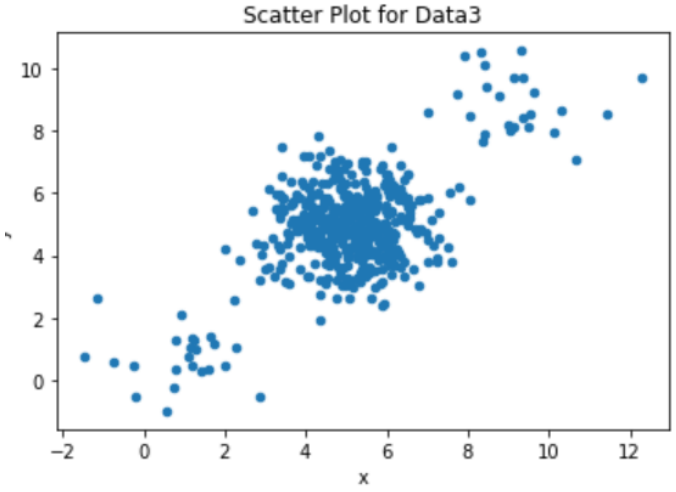



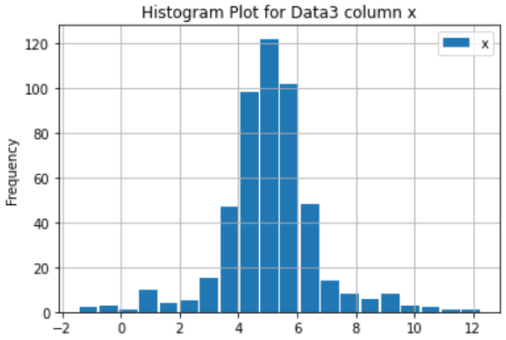



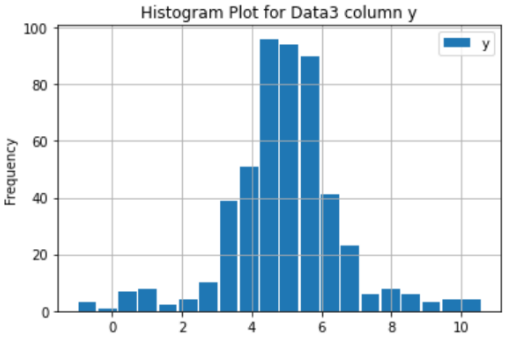



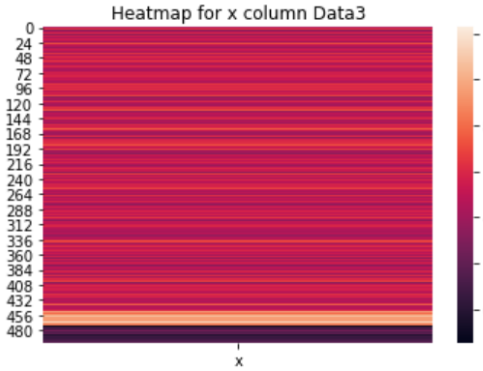



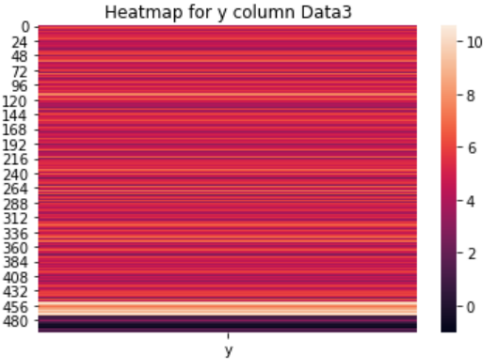
**b) Now perform the same for (data_3).**

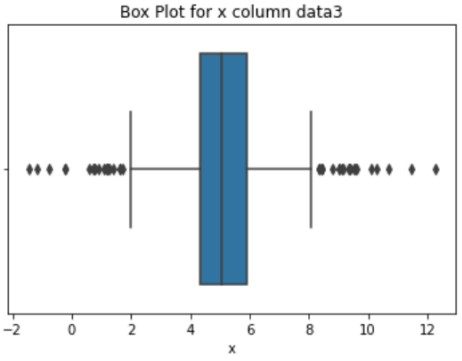


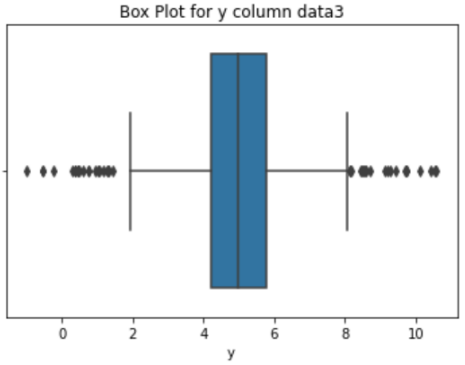




All the above plots were made using **matplotlib** and **seaborn** packages in python. Also one can note that in the box plots for both data 1 and data 3 and for both their columns one can see **outliers** which do not lie in the box (blue) region

**c) Calculate the statistics for both data sets.**

Statistics for Data1

```
The mean for data1 column x is 4.9397429184888235
The median for data1 column x is 4.924278397765319
The mode for data1 column x is 4.924278397765319
The variance for data1 column x is 0.9737801246861147

The mean for data1 column y is 5.042983531876115
The median for data1 column y is 5.074768390017326
The mode for data1 column y is 5.074768390017326
The variance for data1 column y is 1.0164610197905017

The Range for data1 column x is 5.743406828227609
The Interquartile Range for column x for data 1 is -1.3032264593322562
The Range for data1 column y is 6.008928657297398
The Interquartile Range for column y for data 1 is -1.3509156935779743
```

| | x | y |
|---|---|---|
| **count** | 500.000000 | 500.000000 |
| **mean** | 4.939743 | 5.042984 |
| **std** | 0.986803 | 1.008197 |
| **min** | 2.373638 | 2.181180 |
| **25%** | 4.303987 | 4.331464 |
| **50%** | 4.924278 | 5.074768 |
| **75%** | 5.607214 | 5.682380 |
| **max** | 8.117045 | 8.190109 |

Statistics for Data3

```
The mean for data3 column x is 5.082467729231953
The median for data3 column x is 5.05487549200244
The mode for data3 column x is 5.05487549200244
The variance for data3 column x is 2.662558188756155

The mean for data3 column y is 4.952896424504495
The median for data3 column y is 4.978847078474789
The mode for data3 column y is 4.978847078474789
The variance for data3 column y is 2.634758133985419

The Range for data3 column x is 13.725427048471568
The Interquartile Range for column x for data 3 is -1.572622064630803
The Range for data3 column y is 11.593352800114099
The Interquartile Range for column y for data 3 is -1.350915693577974
```

| | x | y |
|---|---|---|
| **count** | 500.000000 | 500.000000 |
| **mean** | 5.082468 | 4.952896 |
| **std** | 1.631735 | 1.623194 |
| **min** | -1.458403 | -1.004100 |
| **25%** | 4.318981 | 4.218101 |
| **50%** | 5.054875 | 4.978847 |
| **75%** | 5.891603 | 5.782668 |
| **max** | 12.267025 | 10.589252 |

The analysis was done using **scipy** and **statistics** library in python

**d) Detect the outliers in data_3 using standard deviation approach and MAD approach**

Standard Deviation Approach

Used z score function from the statistics library to detect outliers. The cutoff was 3

```
The Outliers in Data 3 in first column is with index numbers
(array([450, 459, 462, 468, 471, 475, 476, 483, 488, 492], dtype=int64),)
```

```
(10.10393747643514,
 10.67798532964827,
 12.2670245277286,
 11.44965456902049,
 10.28782093813039,
 -1.458402520742969,
 -1.171853375119153,
 -0.763132811291513,
 -0.2198563833130491,
 -0.2439263484483771)
```

(The points in x column which are outliers)

```
The Outliers in Data 3 in second column is with index numbers
(array([454, 464, 470, 474, 488, 491, 493, 497], dtype=int64),)
```

```
(10.43902165955905,
 10.55061243676815,
 10.58925243952102,
 10.11551641326384,
 -0.5334956955315495,
 -0.2557338130229054,
 -1.004100360593079,
 -0.5528195925039532)
```

(The points in y columns which are outliers)

MAD Approach

Made a code for modified Zm score to find outliers in data3 .

```
The Outliers in Data 3 in first column is with index numbers
(array([450, 451, 452, 455, 456, 457, 458, 459, 461, 462, 463, 465, 466,
       468, 470, 471], dtype=int64),)
```

```
(10.10393747643514,
 9.14841306110033,
 9.630313376450168,
 9.37144942459267,
 9.005201225564582,
 9.039507424436444,
 10.67798532964827,
 9.510497893324324,
 12.2670245277286,
 9.5456448855292,
 9.375609831291062,
 8.780482242073463,
 11.44965456902049,
 9.321302250247287,
 10.28782093813039)
```

(The points in x column which are outliers using MAD Approach)

```
The Outliers in Data 3 in second column is with index numbers
(array([451, 452, 454, 460, 462, 463, 464, 465, 466, 467, 468, 470, 471,
       472, 474], dtype=int64),)
```

```
(9.745989964192137,
 9.265264240840457,
 10.43902165955905,
 9.219071559753377,
 9.717401494534483,
 8.5540601585995,
 10.55061243676815,
 9.749970745029252,
 9.127858533076473,
 8.581245337617629,
 8.529730294461498,
 10.58925243952102,
 8.693157145943976,
 9.431652444355333,
 10.11551641326384)
```

(The points in y column which are outliers using MAD Approach)

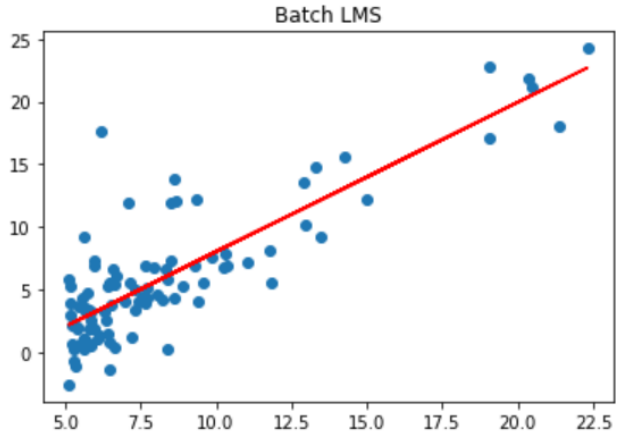
Thus we can see that there is a difference in both the approaches (Standard Deviation and MAD Approach) because though some points that are detected as outliers are the same for the cases but for other points it remains different. Which one approach to choose would depend upon the type of dataset we need to work upon and further advanced techniques would be needed to figure that out.

**Q2 You are CEO of a clothing company with outlets in many cities. You have decided to open an outlet in a new city. To help with the decision of selecting a city, you decide to look at population vs profit data and apply linear regression to see if any relation exists between population & profit with population being the independent variable.**

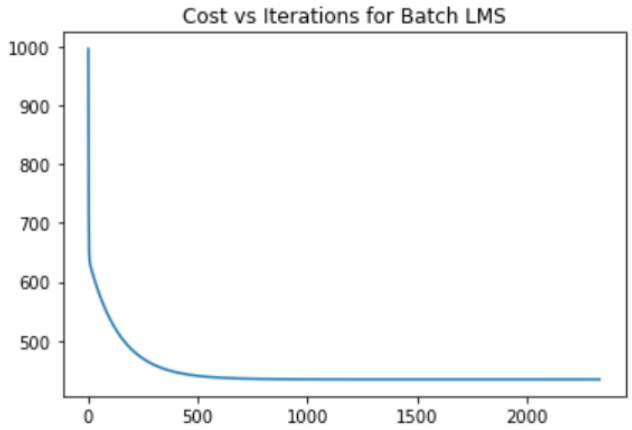
**(a) Apply Batch LMS, Stochastic LMS and Least Square closed form solution and compare the results. Plot the graphs of the obtained results and training data. Use the learning rate of 0.1. Analyze the results(Convergence time, accuracy etc.)(Don't use in-built packages.)**

**Batch LMS**

For alpha = 0.1 the solution didn't converge hence used **alpha = 0.001**
The convergence criterion used was **10^(-6)**
The value of theta0 (intercept) and theta1(coefficient) respectively obtained is
**-3.913816639814972 1.1929072987065412**



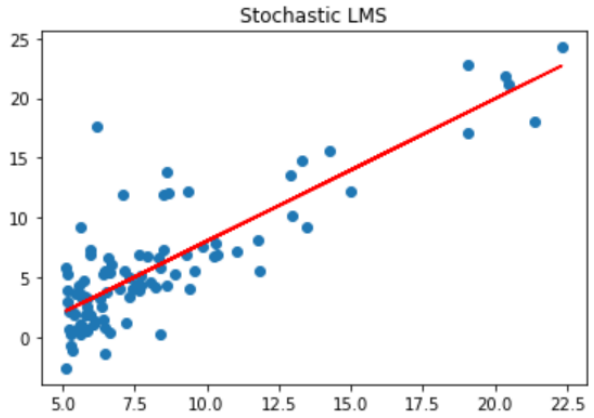

The red line in the above graph shows the fitted line computed using the theta values whereas blue dots shows the data points



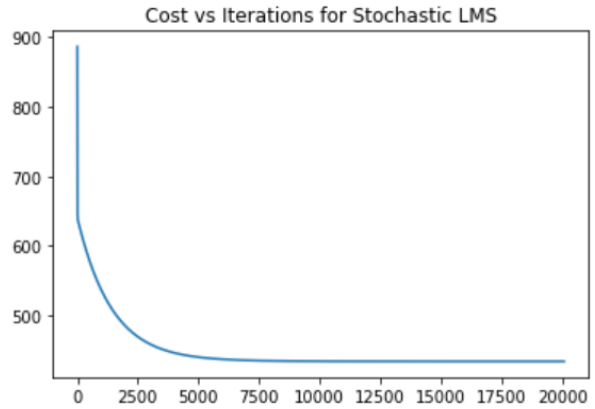

The above graph shows how the cost converges as we increase the number of iterations

**Stochastic LMS**

For alpha = 0.1 the solution didn't converge hence used **alpha = 0.001**
The convergence criterion used was **10^(-6)**
The value of theta0 (intercept) and theta1(coefficient) respectively obtained is
**-3.9110438365155096 1.1926309257447079**



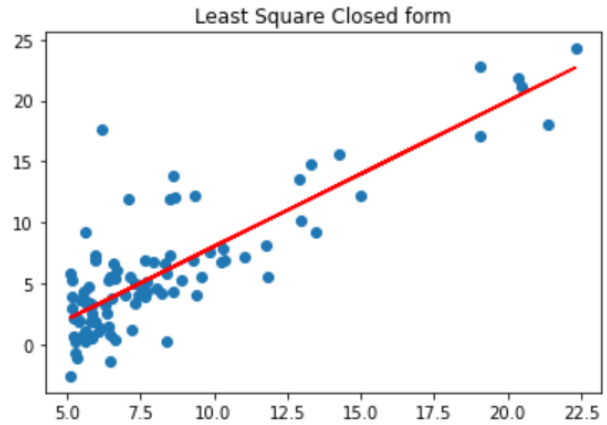

The red line in the above graph shows the fitted line computed using the theta values for stochastic LMS whereas blue dots shows the data points




The above graph shows how the cost converges as we increase the number of iterations. Notice that for batch LMS the iterations range is showing to be 2000 while for stochastic it is shown to be 20000 (**10 times more**) thus stochastic LMS is computationally more expensive.

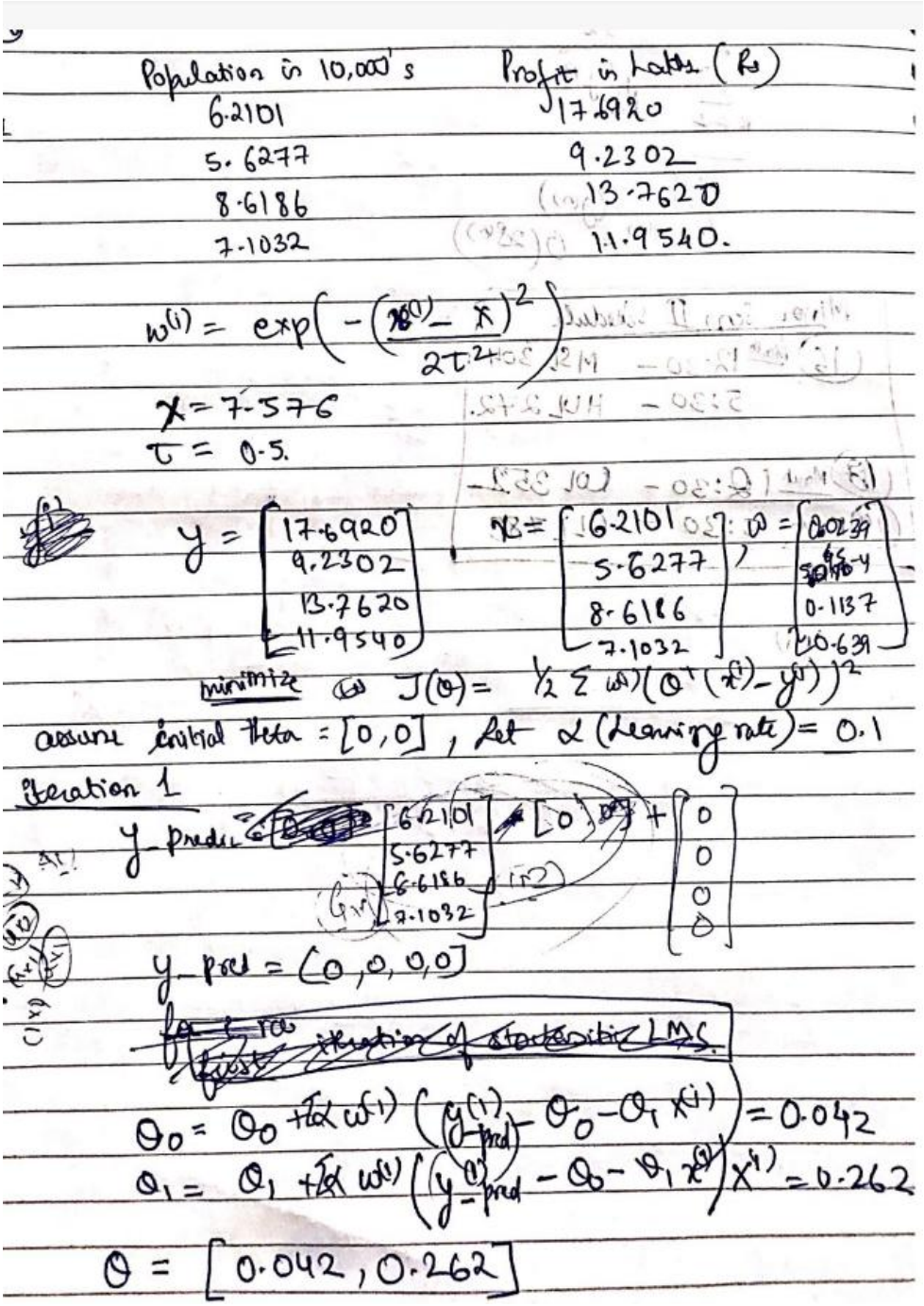
**Least Square closed form**

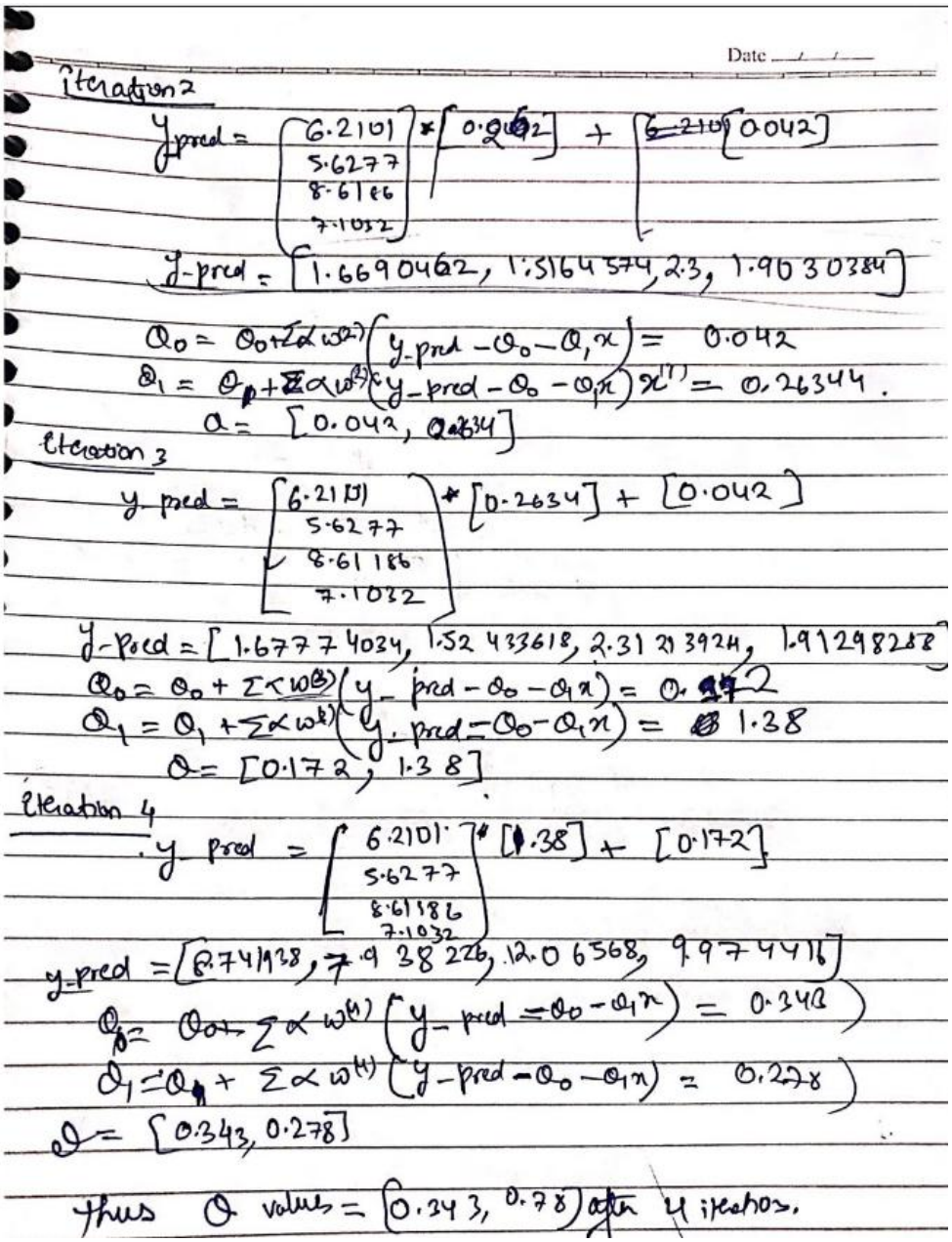
The value of theta0 (intercept) and theta1(coefficient) respectively obtained is **-3.91508424  1.19303364**



The red line in the above graph shows the fitted line computed using the theta values for Least Square closed form whereas blue dots shows the data points
Notice that in all the cases the intercept and coefficient values are very close to the least square closed form (which would give the most accurate result and the data points use are very less also so computationally also it is not expensive).

**(b) Manually perform the locally weighted least linear regression using the first four data points given in excel sheet. Query point is 7.576 and bandwidth parameter is 0.5. Perform four iterations by using stochastic LMS.**






**(c) Compare the results of Elastic net, Lasso and Ridge regression. (Use in-built packages)**

**Elastic Net**

The value of theta0 (intercept) and theta1(coefficient) respectively obtained is **-3.85418289 1.18566042**

**Lasso**

The value of theta0 (intercept) and theta1(coefficient) respectively obtained is **-3.85935614 1.18628674**

**Ridge**

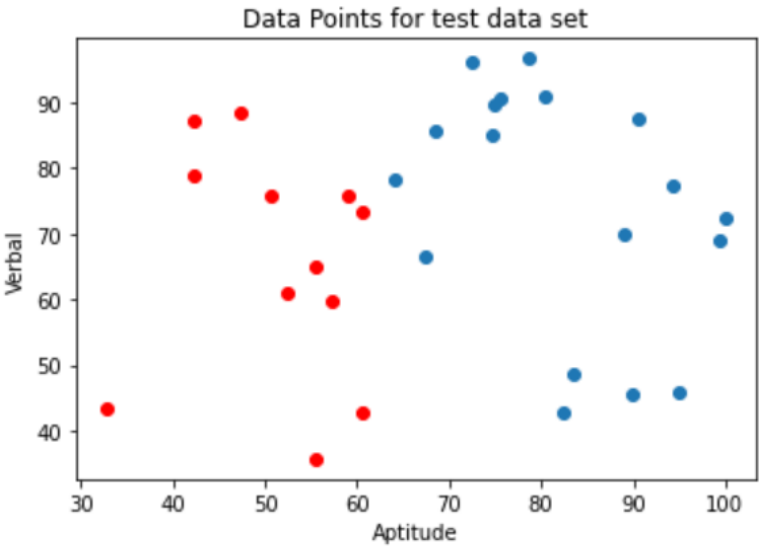
The value of theta0 (intercept) and theta1(coefficient) respectively obtained is **-3.91439887 1.19295067**

We infer that for the three cases, Elastic Net, Lasso and Ridge for (alpha=0.1) the most accurate data is represented by **Ridge Regression (very close to closed form solution)** whereas other two seem to be **computationally more expensive** and would need a lower alpha (learning rate) value to get a close value of intercept and coefficient

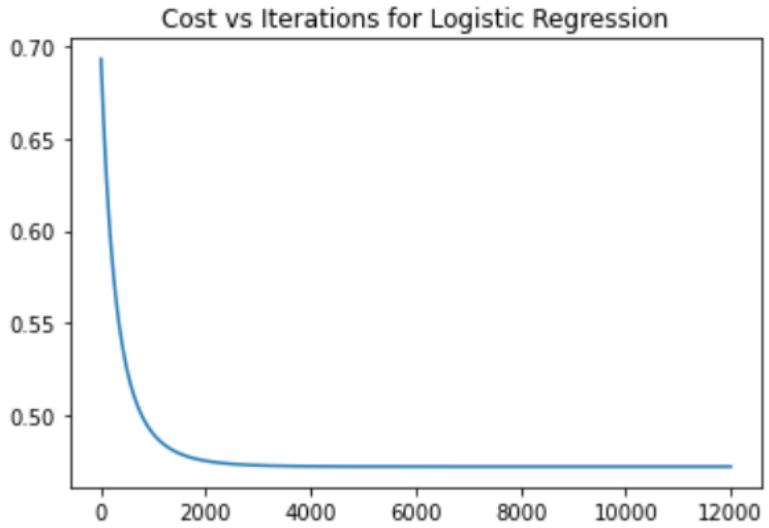
**Q3 A university conducts 2 exams – Aptitude & Verbal as its entrance test to a 2-year program. Based on the scores of these 2 papers, admission is given to students. University has not mentioned the exact criteria of selection. Based on historical data, you need to predict whether a student will get admission based on his/her scores in the 2 exams. Data is provided in q2train.csv & q2test.csv. Train.csv contains training data. First column contains the score of Aptitude exam, 2nd column contains the score of verbal exam and 3rd column indicates whether that student got admission or not. 0 indicates not selected whereas 1 means selected. q2test.csv contains test data.**

**(a) Apply logistic regression on training data with the first 2 columns as input data and the third column as output. Use any suitable learning rate. Now predict admission results on test data (q2test.csv) and print the result in output1.txt with every line of the text file containing either 0 or 1. Plot the results. (Don't use inbuilt packages.)**

Made a Logistic Regression Model and applied it for **alpha=0.0075** and **12000** iterations on scaled values of train data using **StandardScaler** function and used the model to predict labels for the scaled test values.
Predicted Labels are stored in the **'output.txt'** file.
To check the accuracy of the model, I used the **train_test_split** function to split my training data (x) into x_train (70% values) and x_test (30% values) dataset and the accuracy obtained on the x_test was approximately **85%.**
The obtained theta values **[theta0 (intercept),theta1 (coefficient 1),theta2 (coefficient 2)]** : **[0.05476858,1.6502387,0.31334178]**
The final converged cost is **0.474938**




In the above graph, the red data points show the points having 0 (fail) as predicted label while blue points show the points having 1 (fail) as predicted label.




The above graph shows how cost (error) declines as we increase the iterations, after 12000 iterations it was observed that the cost was the same upto 6 decimal places. Hence the converged plot was obtained.