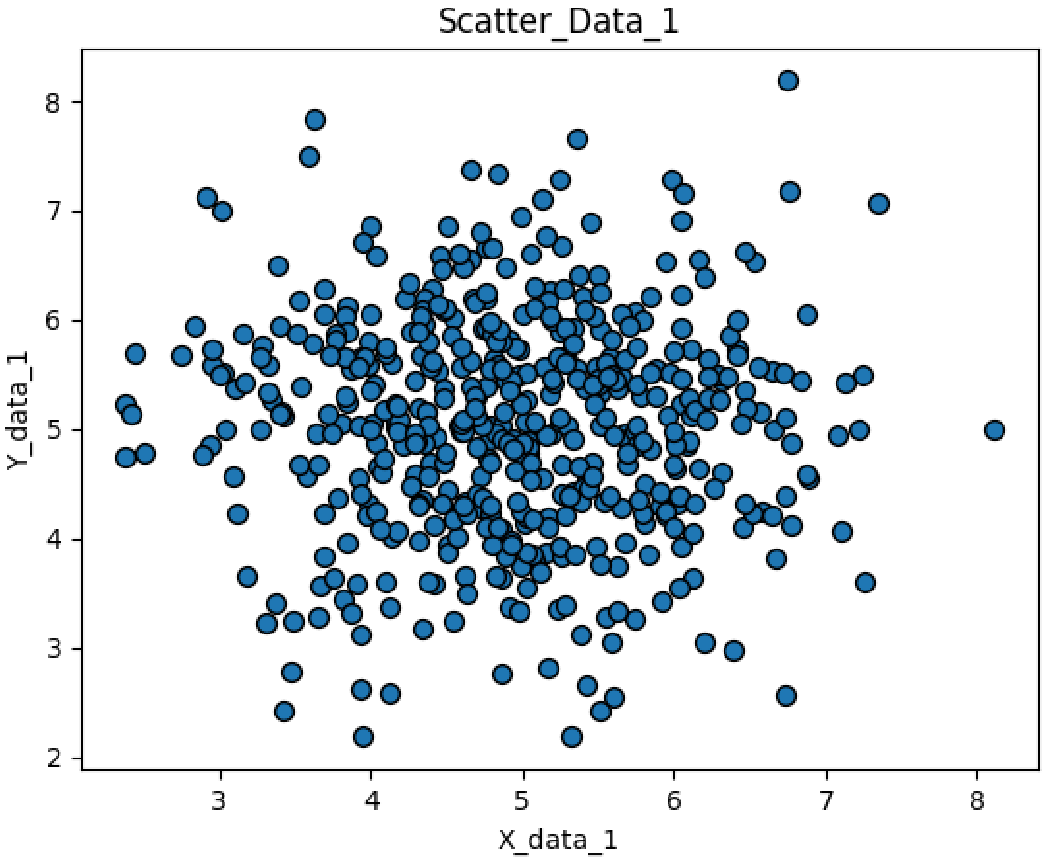
CLL788

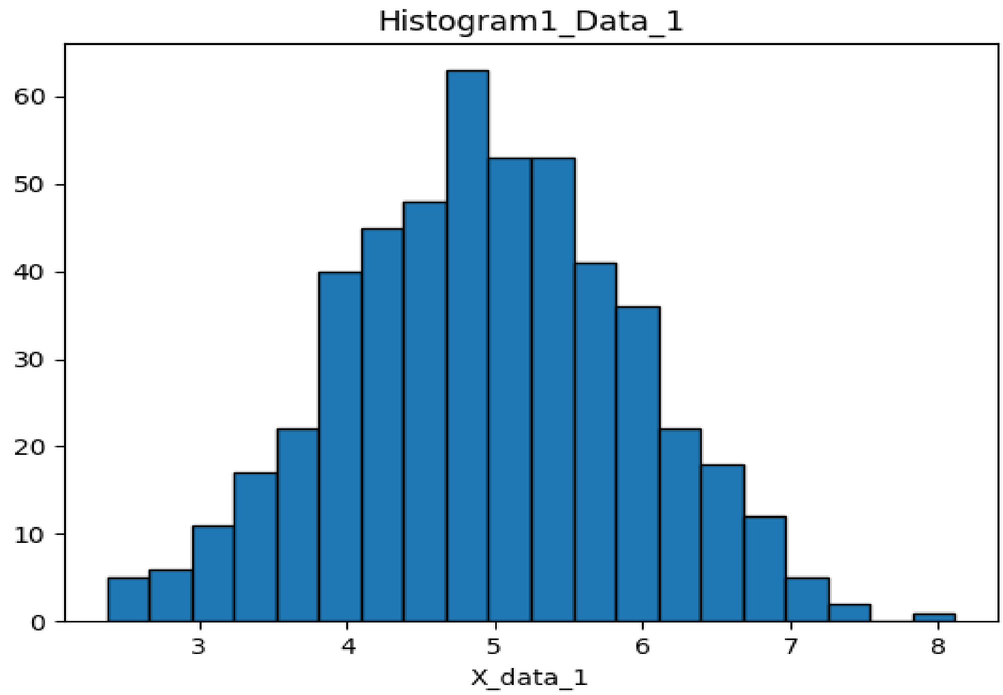
Assignment 1

1. The code for Q1 is uploaded as question1.py

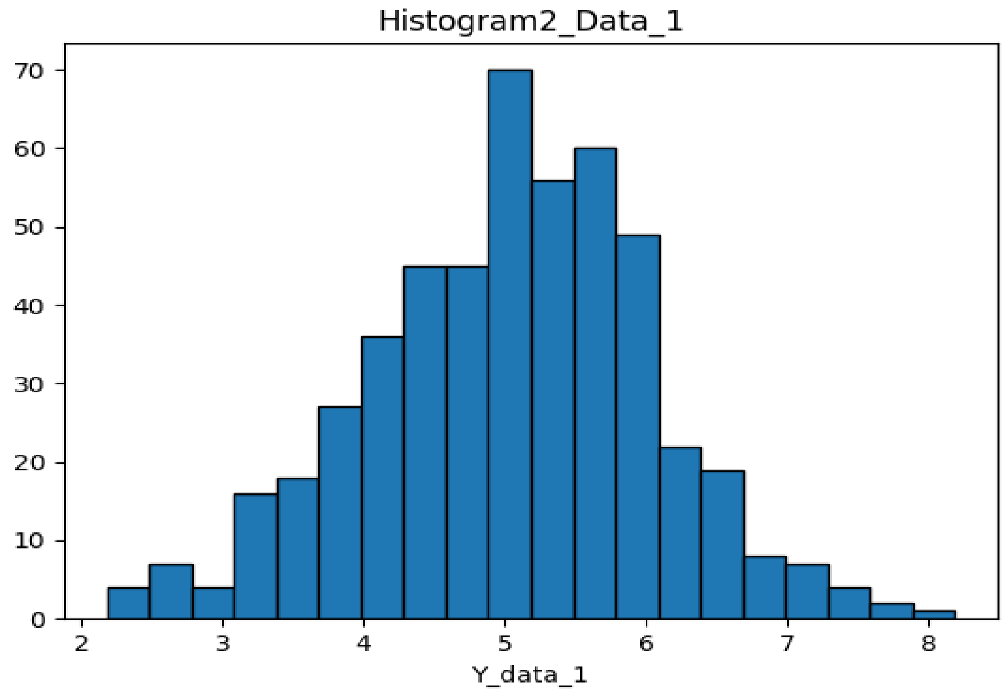
(i) The graphs obtained for Data1.xlsx are:



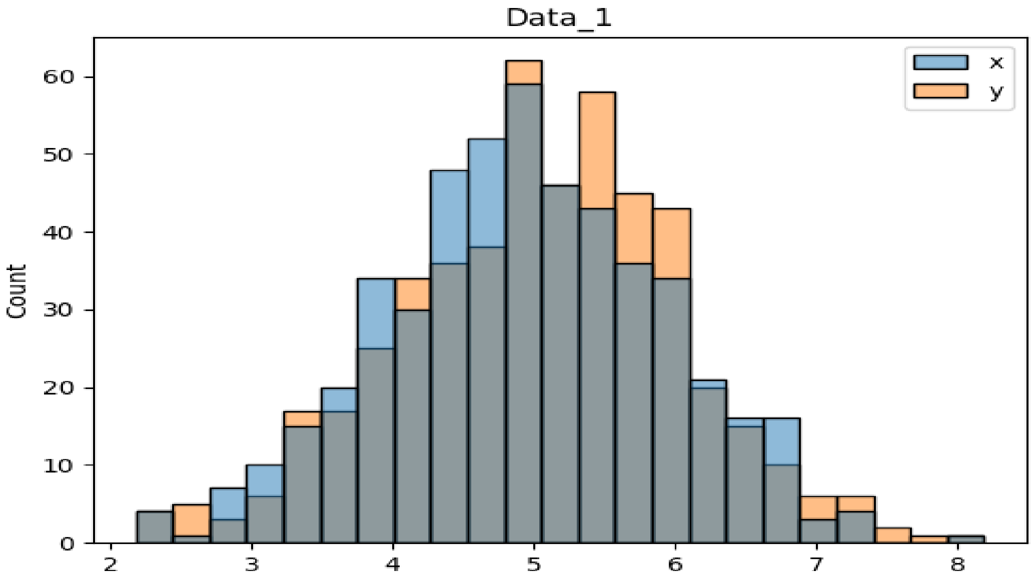
SCATTER PLOT



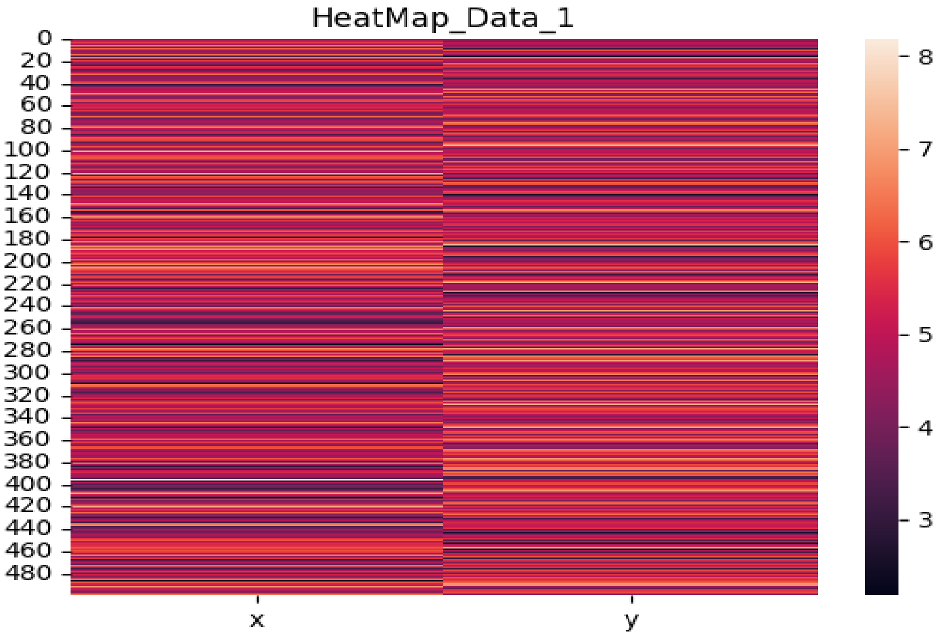
HISTOGRAM for X values in Data1



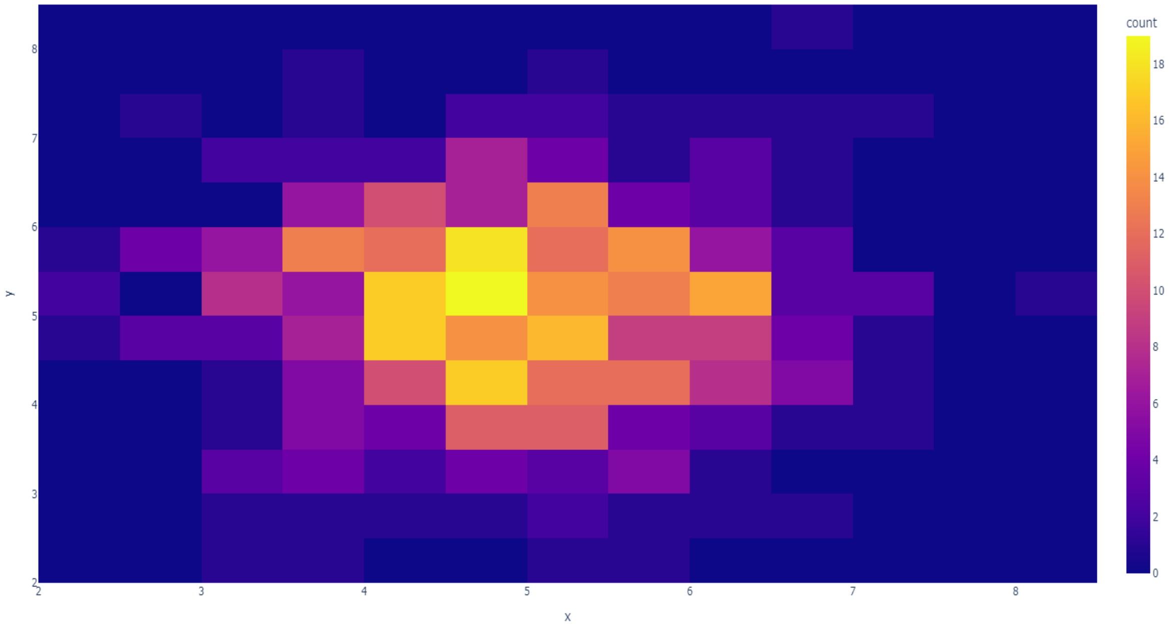
HISTOGRAM for Y values in Data1



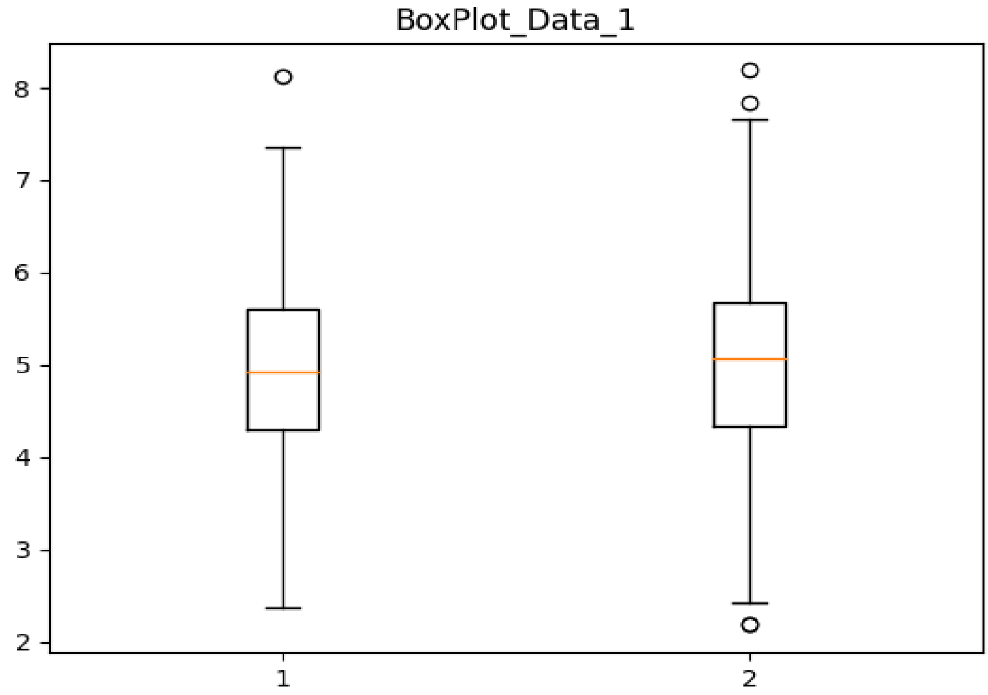
HISTOGRAM for Data1



HEATMAP for Data1

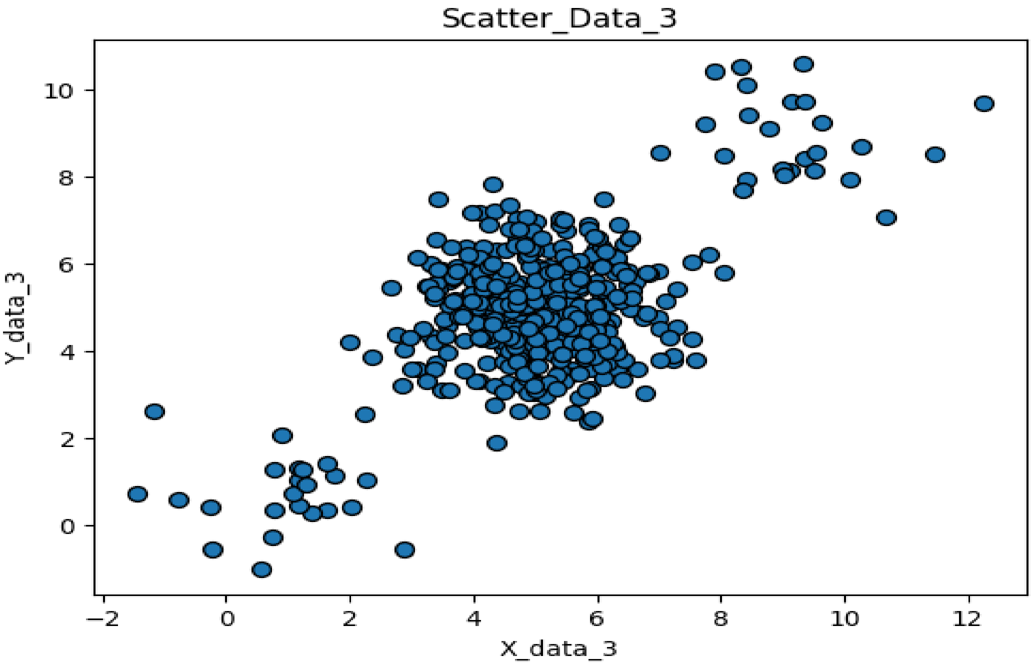


DENSITY MAP for Data1

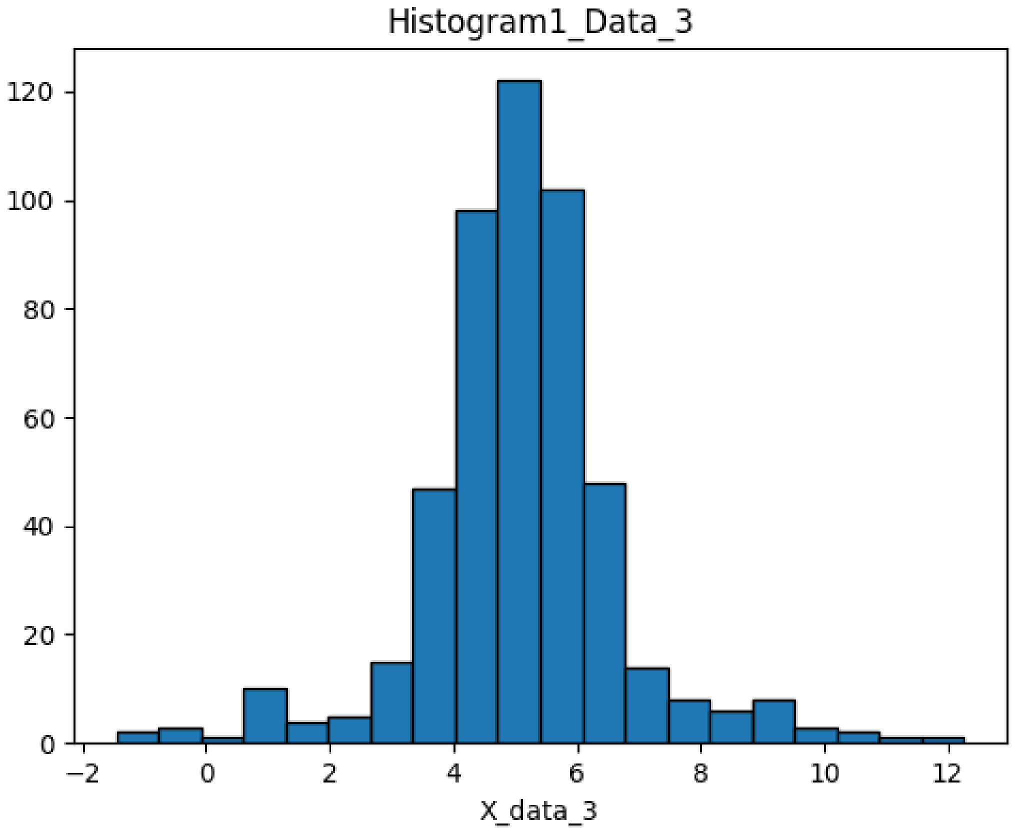


BOXPLOT for Data1 [1:x, 2: y]

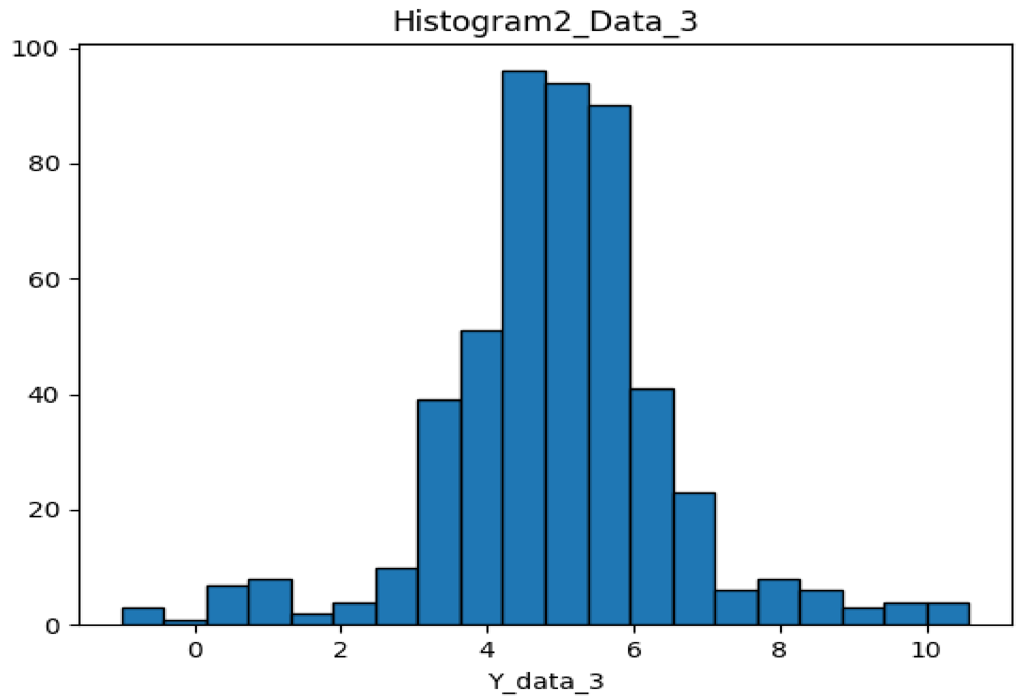
(ii) The graphs obtained for Data3.xlsx are:



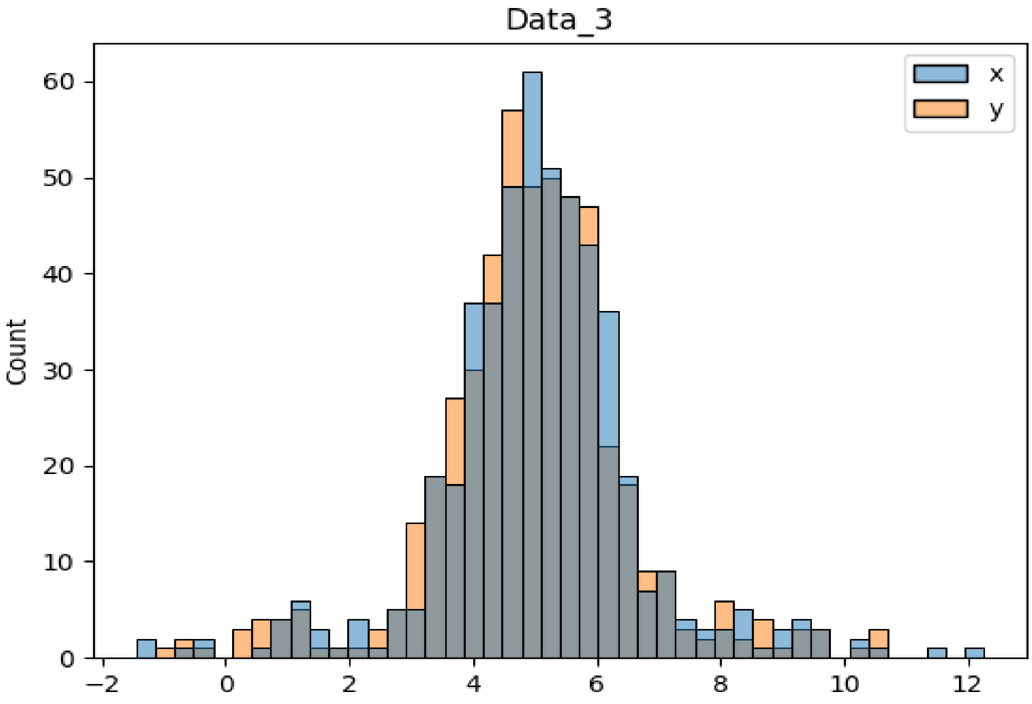
SCATTER PLOT for Data3



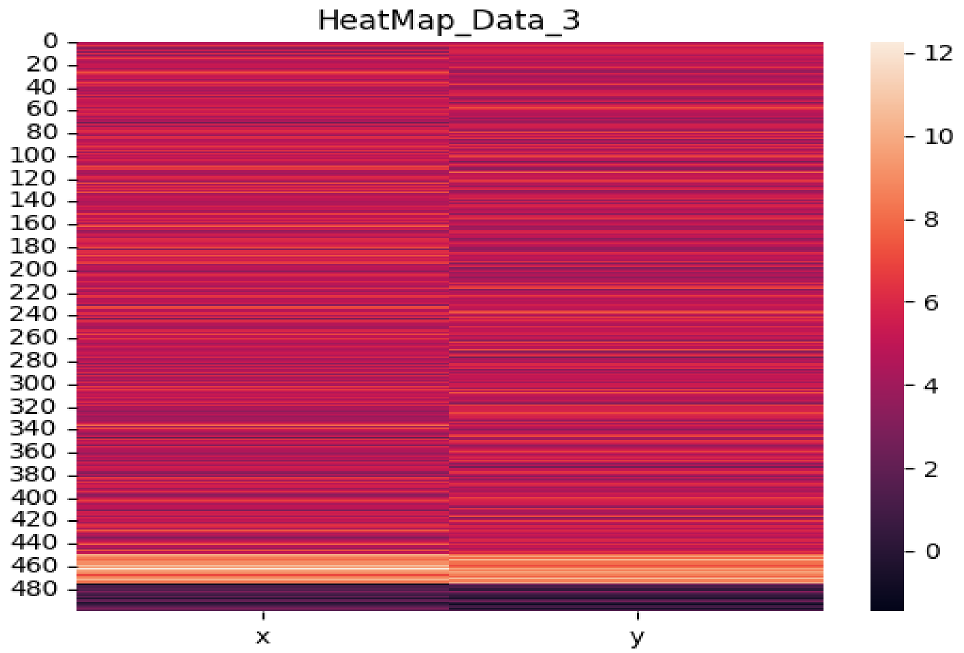
HISTOGRAM for x values in Data3



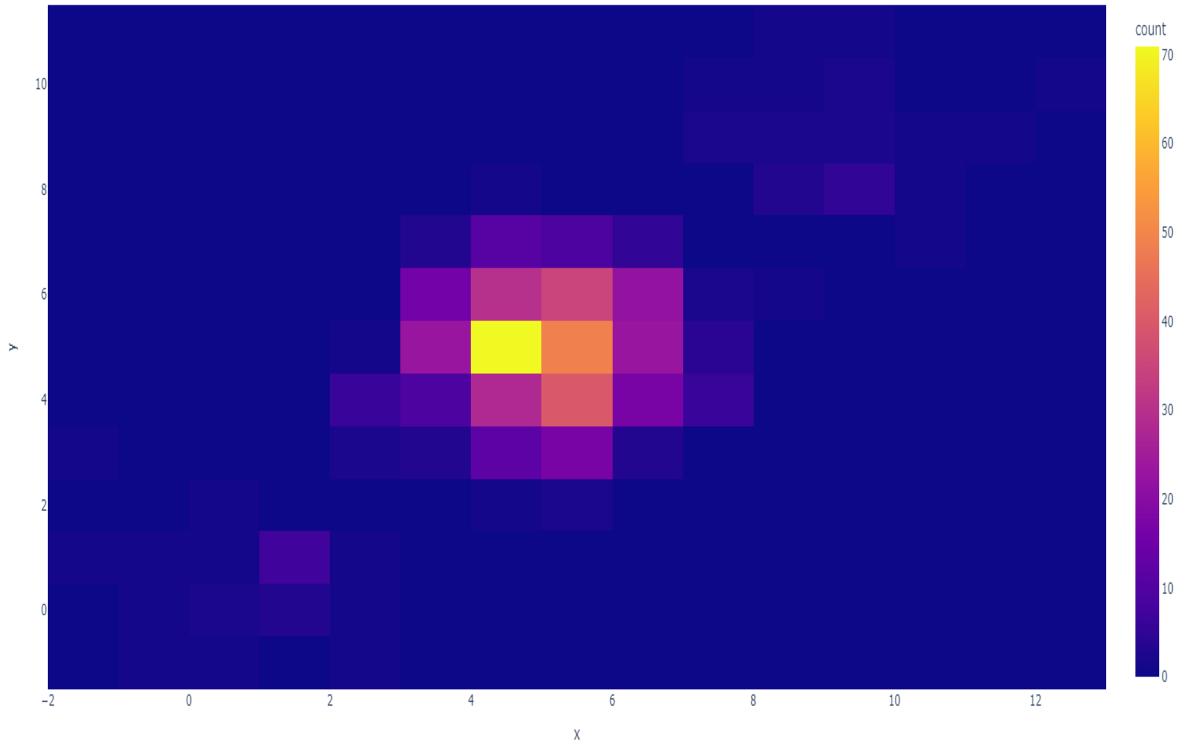
HISTOGRAM for y values in Data3



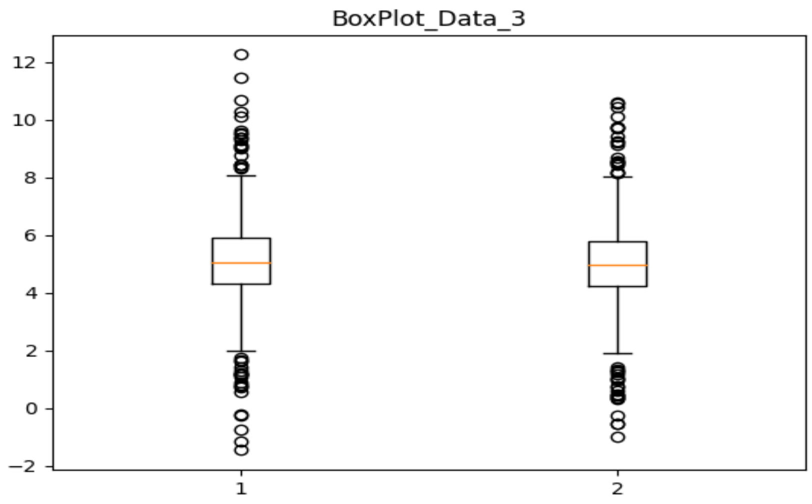
HISTOGRAM for Data3



HEATMAP for Data3



DENSITY MAP for Data3



BOXPLOT for Data3

(iii) The statistics calculated for data1 is as follows:

	x	y
Count	500.00	500.00
Mean	4.939743	5.042984
Standard Deviation	0.986803	1.008197
Variance	0.973780	1.016461
Minimum Value	2.373638	2.181180
25% Percentile	4.303987	4.331464
50% Percentile(Median)	4.924278	5.074768
75% Percentile	5.607214	5.682380
Maximum Value	8.117045	8.190109

The statistics calculated for data3 are as follows:

	x	y
Count	500.00	500.00
Mean	5.082468	4.952896
Standard Deviation	1.631735	1.623194
Variance	2.662559	2.634758
Minimum Value	-1.458403	-1.00410
25% Percentile	4.318981	4.218101
50% Percentile (Median)	5.054875	4.978847

75% Percentile	5.891603	5.782668
Maximum Value	12.267025	10.589252

(iv) I used the Z score 3 as threshold for detecting outliers through standard deviation approach. Similarly, in MAD approach also I considered Z_M score 3 to be the threshold for detecting outliers.

The outliers detected through standard deviation approach in Data3 are:

Index: 450 x: 10.10393747643514 y: 7.959553573726425

Index: 454 x: 7.906331287245181 y: 10.43902165955905

Index: 459 x: 10.67798532964827 y: 7.082585468001082

Index: 462 x: 12.2670245277286 y: 9.717401494534483

Index: 464 x: 8.329546889576005 y: 10.55061243676815

Index: 468 x: 11.44965456902049 y: 8.529730294461498

Index: 470 x: 9.321302250247287 y: 10.58925243952102

Index: 471 x: 10.28782093813039 y: 8.693157145943976

Index: 474 x: 8.4268435572467 y: 10.11551641326384

Index: 475 x: -1.458402520742969 y: 0.7395934657790222

Index: 476 x: -1.171853375119153 y: 2.63532019232785

Index: 483 x: -0.763132811291513 y: 0.5801206333486194

Index: 488 x: -0.2198563833130491 y: -0.5334956955315495

Index: 491 x: 0.7398399056096148 y: -0.2557338130229054

Index: 492 x: -0.2439263484483771 y: 0.4344124182250851

Index: 493 x: 0.5680693847459939 y: -1.004100360593079

Index: 497 x: 2.873079584466421 y: -0.5528195925039532

The outliers detected through MAD approach in Data3 are:

Index : 450 x: 10.10393747643514 y: 7.959553573726425

Index : 454 x: 7.906331287245181 y: 10.43902165955905

Index : 459 x: 10.67798532964827 y: 7.082585468001082

Index : 462 x: 12.2670245277286 y: 9.717401494534483

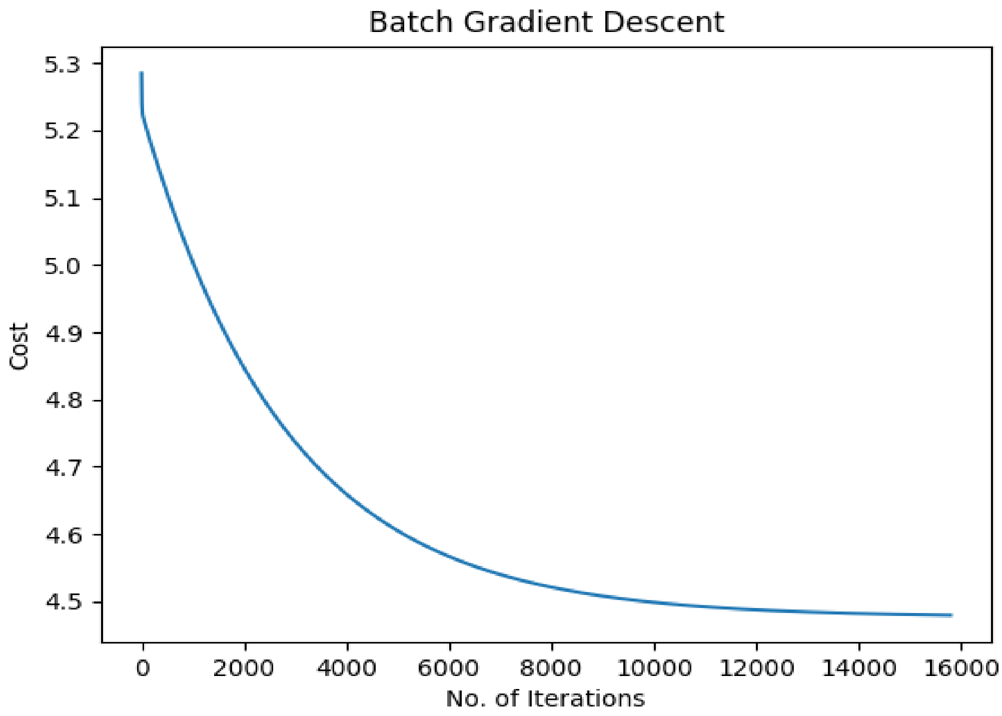
Index : 464 x: 8.329546889576005 y: 10.55061243676815

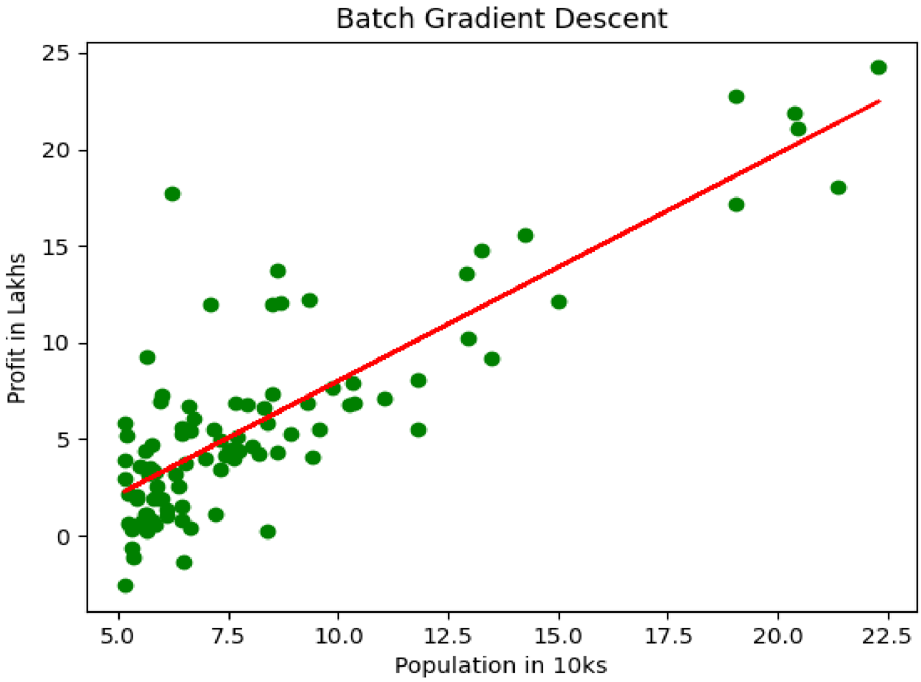
Index : 468 x: 11.44965456902049 y: 8.529730294461498

```
Index :  470  x:  9.321302250247287  y:  10.58925243952102
Index :  471  x:  10.28782093813039  y:  8.693157145943976
Index :  474  x:  8.4268435572467      y:  10.11551641326384
Index :  475  x:  -1.458402520742969  y:  0.7395934657790222
Index :  476  x:  -1.171853375119153  y:  2.63532019232785
Index :  483  x:  -0.763132811291513  y:  0.5801206333486194
Index :  488  x:  -0.2198563833130491  y:  -0.5334956955315495
Index :  491  x:  0.7398399056096148  y:  -0.2557338130229054
Index :  492  x:  -0.2439263484483771  y:  0.4344124182250851
Index :  493  x:  0.5680693847459939  y:  -1.004100360593079
Index :  497  x:  2.873079584466421  y:  -0.5528195925039532
```

2. The codes for Q2 is saved as question2.py

(a) The graphs for batch gradient descent are as follows:





Theta values for Batch Gradient Descent : [-3.73706744 1.17529022]

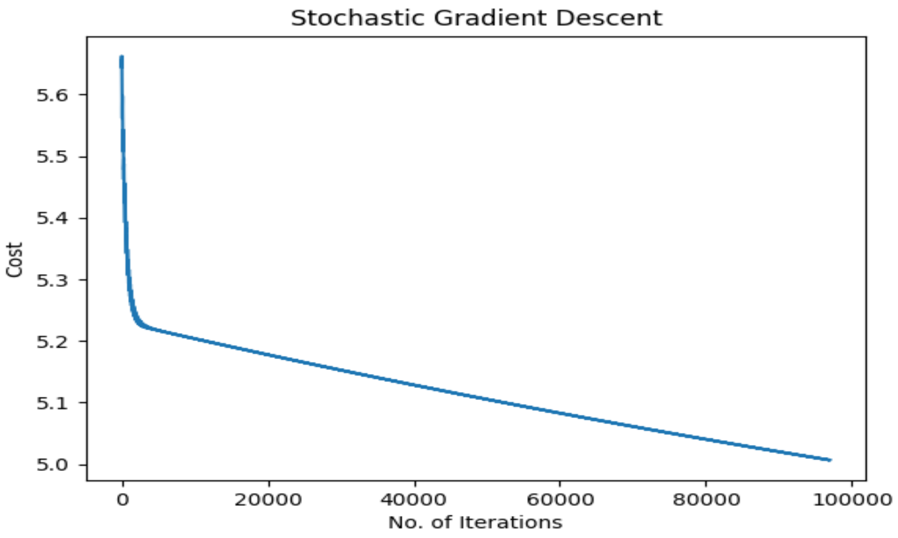
Intercept value for Batch Gradient Descent : -3.737067442532012

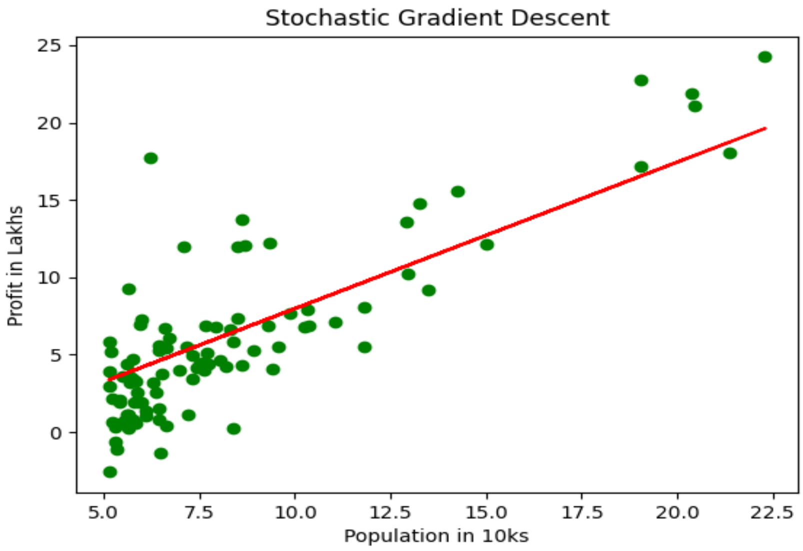
Slope value for Batch Gradient Descent : 1.1752902181280942

Final Cost : 4.479799367820934

Time taken for Batch Gradient Descent : 8.315285682678223

The graphs for Stochastic Gradient Descent are as follows:





Theta values for Stochastic Gradient Descent : [-1.48013207 0.94617247]

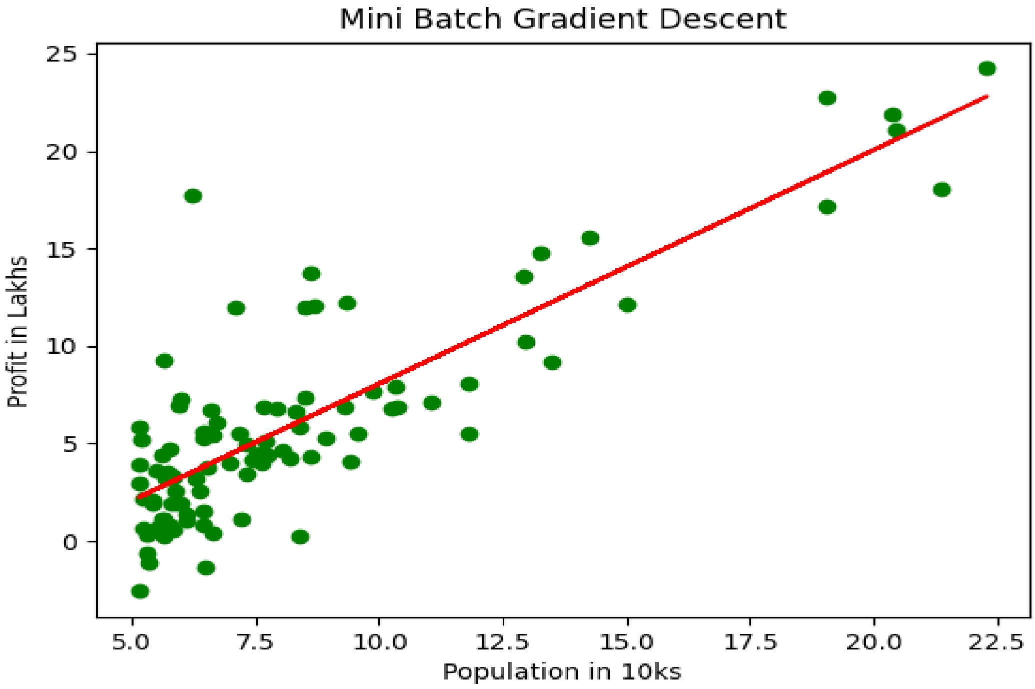
Intercept value for Stochastic Gradient Descent : -1.4801320735237868

Slope value for Stochastic Gradient Descent : 0.9461724708477199

Final Cost 5.006968576684462

Time taken for Stochastic Descent 54.9911675453186

The graphs for Mini-Batch Gradient Descent are as follows:



Theta values for Mini-Batch Gradient Descent : [-3.91489228 1.19800872]

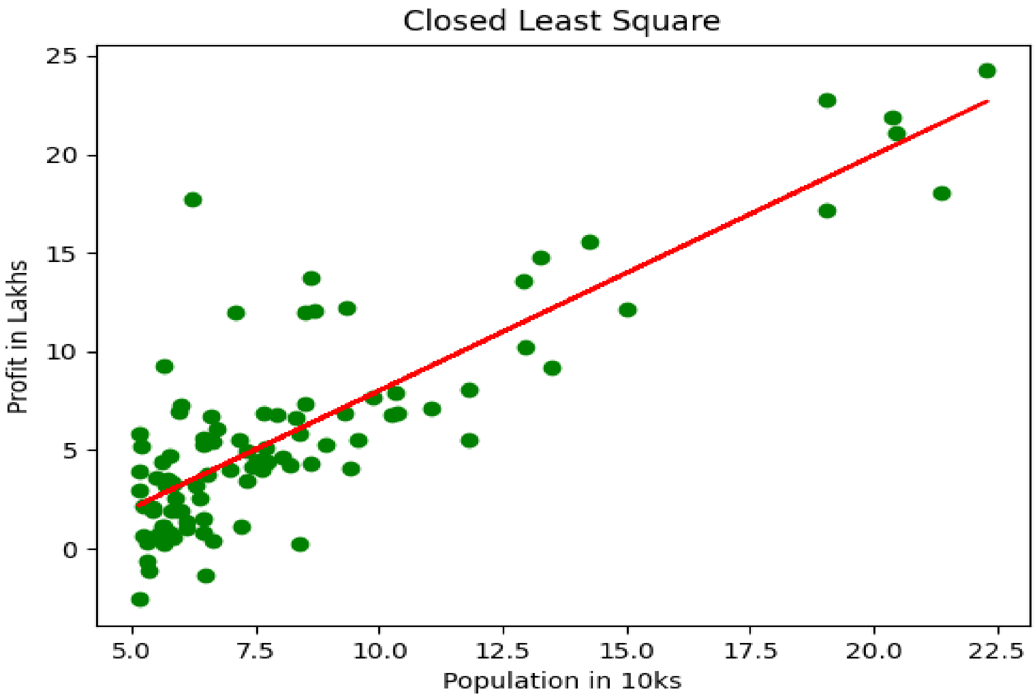
Intercept value for Stochastic Gradient Descent : -3.9148922802298363

Slope value for Stochastic Gradient Descent : 1.1980087225296736

Final Cost 15.864351743928287

Time taken for Mini batch Descent 42.78075456619263

The graphs for least square closed are as follows:



Theta values for Least Square Closed Form : [-3.91508424 1.19303364]

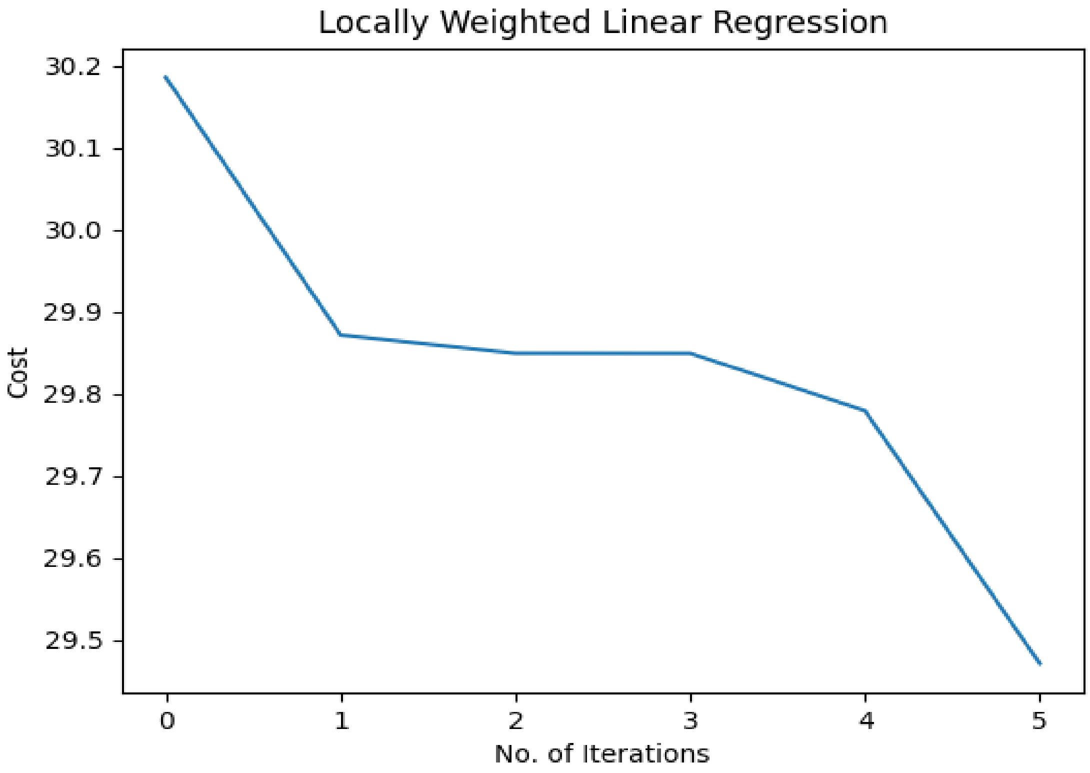
Intercept value for Least Square Closed Form : -3.91508424273081

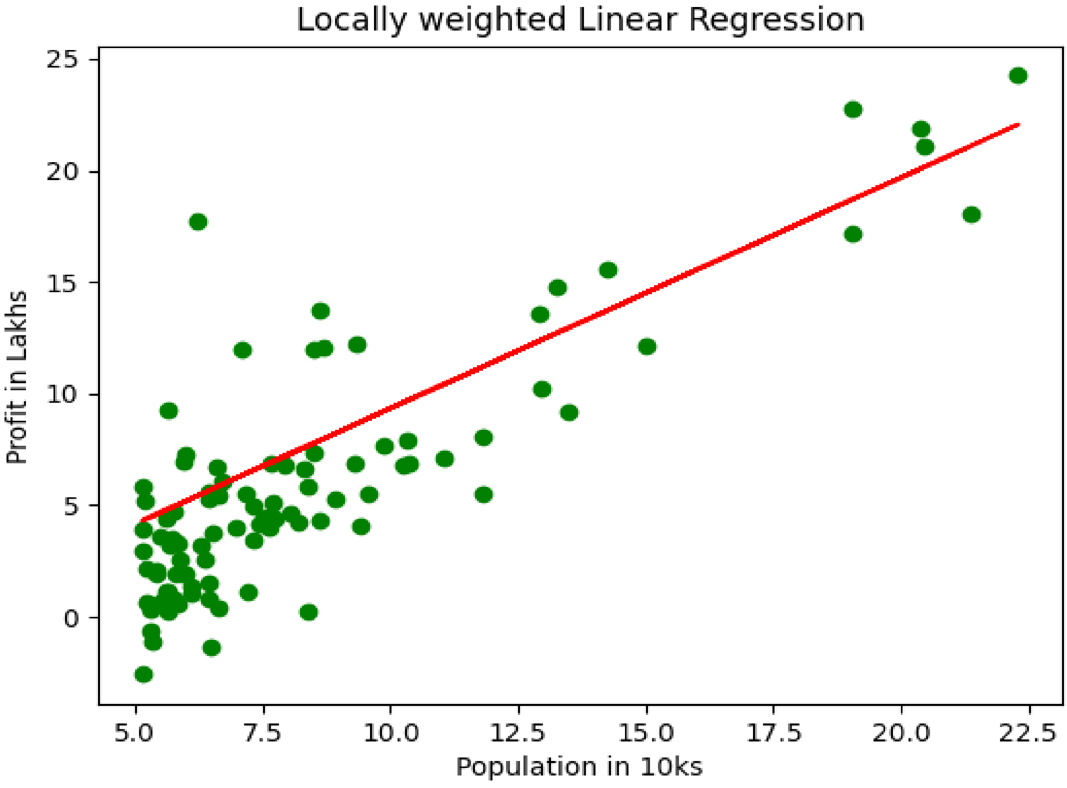
Slope value for Least Square Closed Form : 1.1930336441895941

Final Cost 4.476971375975179

Time taken for Least Square Closed 0.007084608078002930

The graphs for locally weighted linear regression are:


Locally Weighted Linear Regression
Cost
30.2
30.1
30.0
29.9
29.8
29.7
29.6
29.5
0
1
2
3
4
5
No. of Iterations


Locally weighted Linear Regression
Profit in Lakhs
25
20
15
10
5
0
5.0
7.5
10.0
12.5
15.0
17.5
20.0
22.5
Population in 10ks

② (b) Locally Weighted Linear Regression

Population in 10ks →	6.2101	5.6277	8.6186	7.1032
Profit in Lakhs →	17.692	9.2302	13.762	11.954

$x_i = 7.576$; $\tau = 0.5$

$\omega = \exp \frac{(-(x_i - x)^2)}{(2\tau^2)} \Rightarrow$

$$\Rightarrow \quad x = \begin{bmatrix} 17.692 \\ 9.2302 \\ 13.762 \\ 11.954 \end{bmatrix} \qquad \omega = \begin{bmatrix} 0.023 \\ 0.0005 \\ 0.11 \\ 0.039 \end{bmatrix}$$

$$J(\theta) = \frac{1}{2} \Sigma\, \omega_i \left(\theta^T (x^i - y)^2 \right)$$

Iteration 1 : $Y_{pred} = \begin{bmatrix} x \\ \ \end{bmatrix} \times [0] + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

$= [0, 0, 0, 0)$

$\Rightarrow \theta_0 = \theta_0 + \Sigma\, d\, \omega (y_{pred}{}^{i} - \theta_0 - \theta_1 x^i) = 0.04$

$\theta_1 = \theta_1 + \Sigma\, d\, \omega\, (y_{pred}{}^{i} - \theta_0 - \theta_1 x_i . x_i) = 0.26$

$\theta = [\ 0.04, 0.262]$

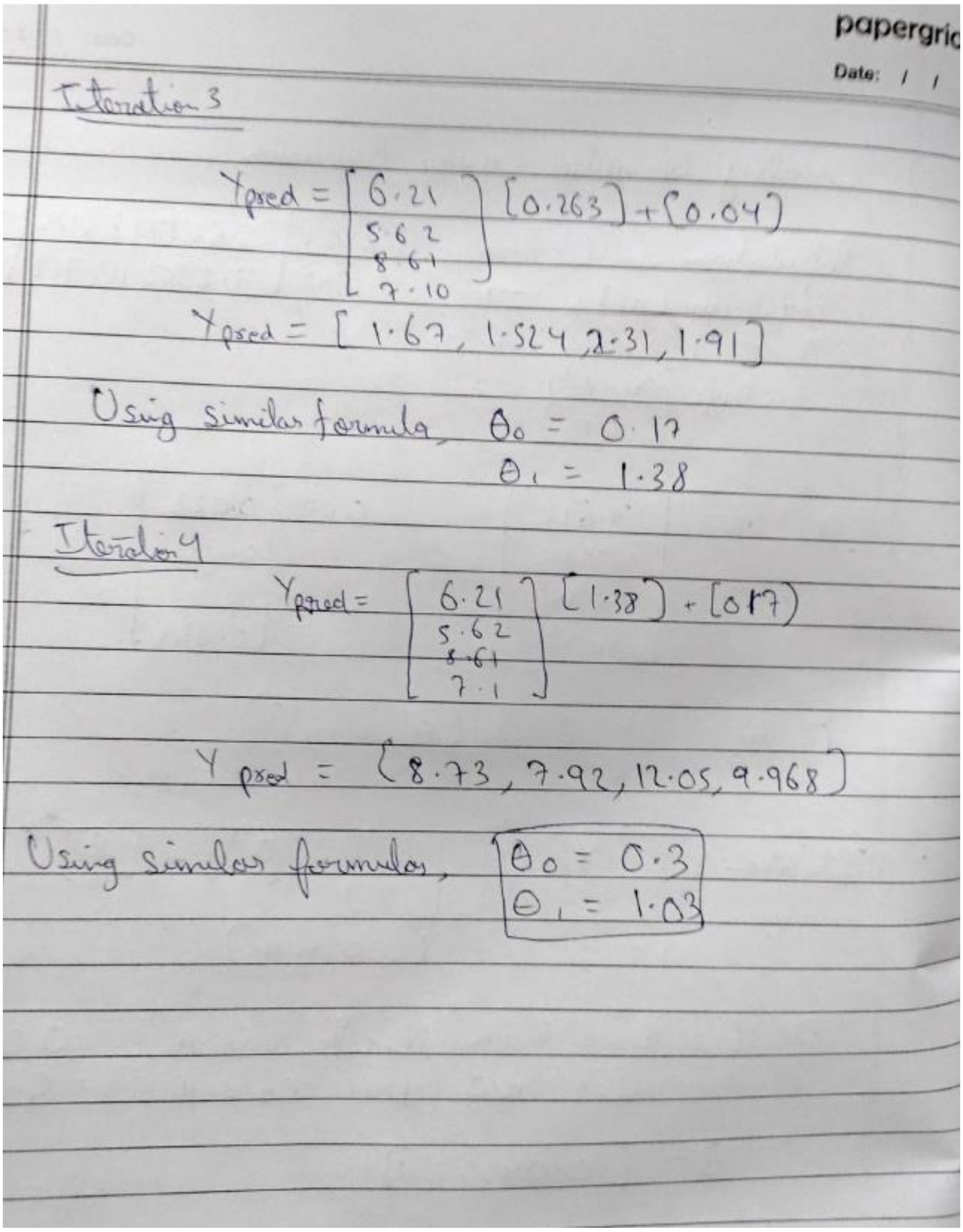
Iteration 2 : $Y_{pred} = \begin{bmatrix} 6.21 \\ 5.62 \\ 8.61 \\ 7.1 \end{bmatrix} [0.262] + [0.04)$

$Y_{pred} = [1.61, 1.46, 2.23, 1.84] + [0.04]$

$= [1.65, 1.5, 2.27, 1.88]$

$\theta_0 = \theta_0 + \Sigma\, d\, \omega^2 (y_{pred} - \theta_0 - \theta_1 x) = 0.04$

$\theta_1 = \theta_1 + \Sigma\, d\, \omega^2 (y_{pred} - \theta_0 - \theta_1 x) = 0.263$



Date: / /

Iteration 3

$$Y_{pred} = \begin{bmatrix} 6.21 \\ 5.62 \\ 8.61 \\ 7.10 \end{bmatrix} [0.263] + [0.04]$$

$$Y_{pred} = [1.67, 1.524, 2.31, 1.91]$$

Using similar formula, $\theta_0 = 0.17$

$\theta_1 = 1.38$

Iteration 4

$$Y_{pred} = \begin{bmatrix} 6.21 \\ 5.62 \\ 8.61 \\ 7.1 \end{bmatrix} [1.38] + [0.17]$$

$$Y_{pred} = [8.73, 7.92, 12.05, 9.968]$$

Using similar formulas, $\boxed{\begin{matrix} \theta_0 = 0.3 \\ \theta_1 = 1.03 \end{matrix}}$

Theta values for Locally Weighted Linear Regression : [-1.   1.03386816]

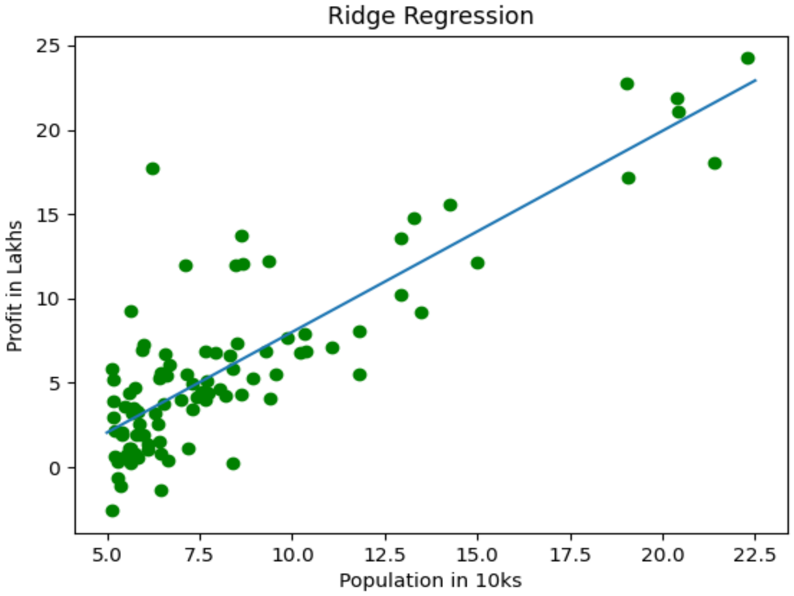
Intercept value for Locally Weighted Linear Regression : -1.0

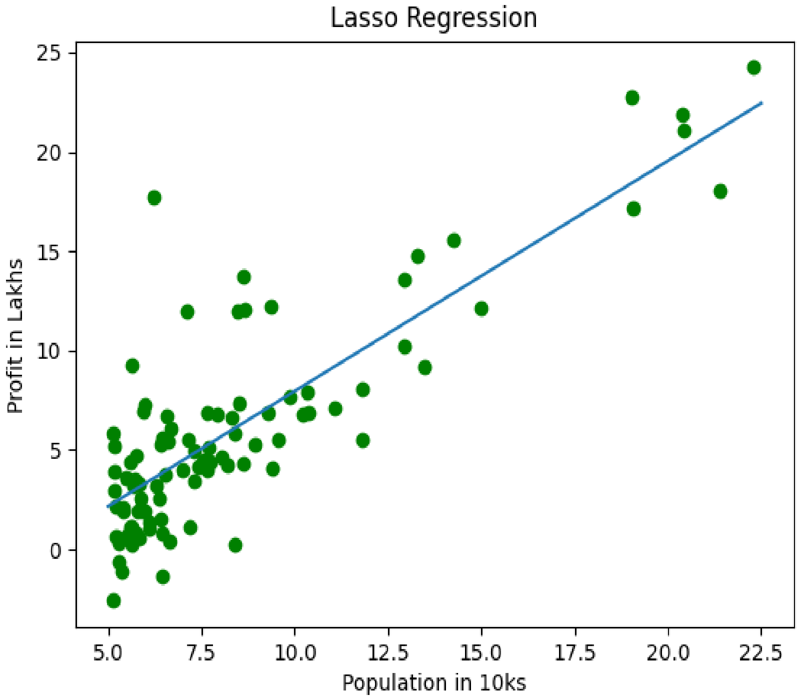
Slope value for Locally Weighted Linear Regression : 1.0338681555502818

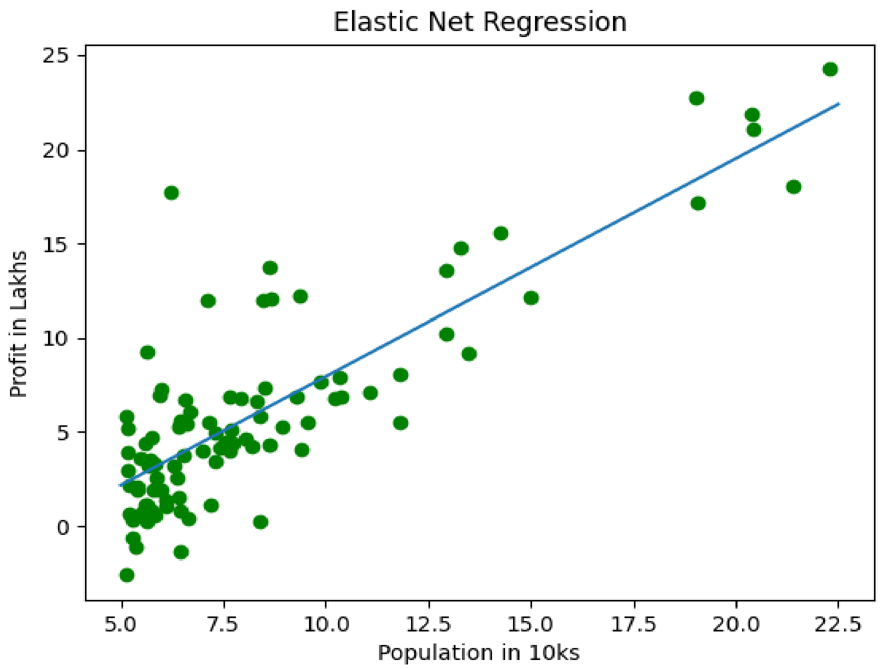
Final Cost 29.471397288114474

Time taken for Locally Weighted Linear Regression 0.01680135726928711

The graphs for Ridge, Lasso and Elastic Regression are as follows:


Ridge Regression
Profit in Lakhs
Population in 10ks
25
20
15
10
5
0
5.0
7.5
10.0
12.5
15.0
17.5
20.0
22.5

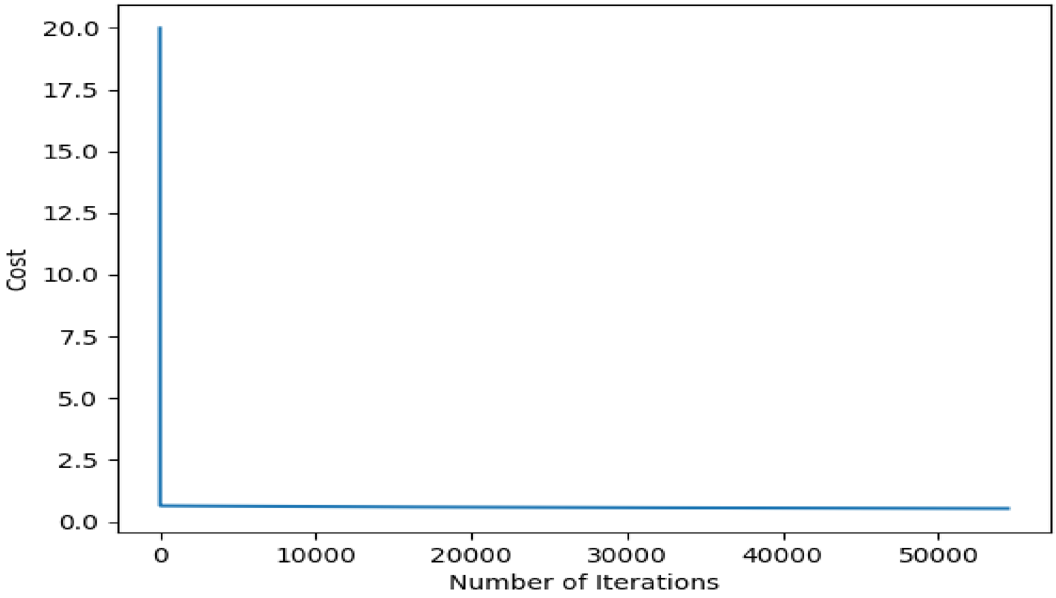

Lasso Regression
Profit in Lakhs
Population in 10ks
25
20
15
10
5
0
5.0
7.5
10.0
12.5
15.0
17.5
20.0
22.5



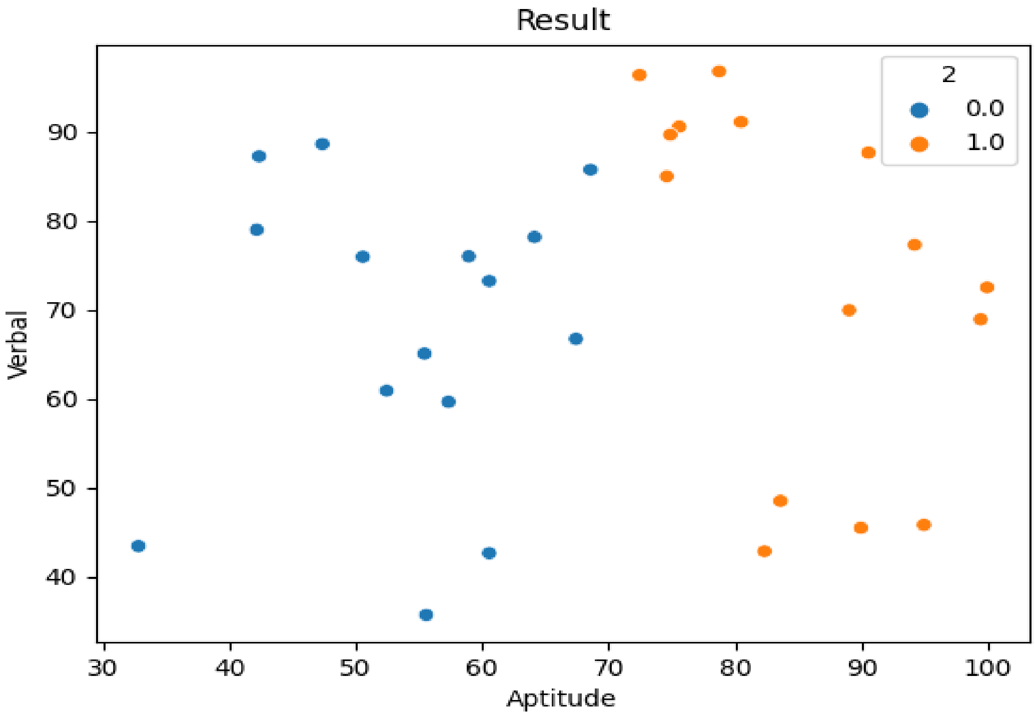
```
Theta values for Ridge Regression : [-3.911658346584465 array([0.        ,
1.19261888])]

Intercept value for Ridge Regression : -3.911658346584465

Slope value for Ridge Regression : 1.1926188767440928

Final Cost 4.476972650870759

Time taken for Ridge Regression 0.024686813354492188


Theta values for Lasso Regression : [-3.636443737640752 array([0.        ,
1.15929911])]

Intercept value for Lasso Regression : -3.636443737640752

Slope value for Lasso Regression : 1.1592991100495345

Final Cost 4.485405009510193

Time taken for Lasso Regression 0.006447792053222656


Theta values for Elastic Net Regression : [-3.6146182812011443 array([0.        ,
1.15665674])]

Intercept value for Elastic Net Regression : -3.6146182812011443

Slope value for Elastic Net Regression : 1.1566567389945928

Final Cost 4.48677793819175
```

Time taken for Elastic Net Regression 0.0061910152435302734

3. The result of logistic regression is uploaded as output1.txt. The code is uploaded as question3.py. I used the value 0.7 as threshold in logistic regression.



Logistic Regression cost vs number of iterations



5. The code is uploaded as question5.py.

   After visualizing the tree, we observe that the root node of that tree was - Outlook. The tree is printed in the output.