Homework 3
Applied Machine Learning
Fall 2017
CSCI-P 556/INFO-I 526

Instructor: Hasan Kurban

October 27, 2017

All the work done here is solely mine. - Manoj Joshi

Directions

Please follow the syllabus guidelines in turning in your homework. I am providing the LaTeX of this document too. This homework is due Friday Oct 27, 2017 11:59p.m. **OBSERVE THE TIME**. Absolutely no homework will be accepted after that time. Bring a hard-copy to Tuesday's class on the 1st. If you do not bring a hard-copy with the statement of your own work, the homework will not be accepted. All the work should be your own. Within a week, AIs can contact students to examine code; students must meet within three days. The session will last no longer than 5 minutes. If the code does not work, the grade for the program may be reduced. Lastly, source code cannot be modified post due date.

Linear and Logistic Regression

This part is provided to help you implement linear and logistic regression.

Notations

- Δ: data set
- m: number of training examples, n: number of features, x's: input variables, y's: output variable.
- $(x^{(i)}, y^{(i)})$: i^{th} training example
- $x_j^{(i)}$: value of feature j in i^{th} training example
- $x_0 = 1$ (the first feature (x_0) is a vector of 1's) – you should add $x_0 = 1$ to data before answering the questions.
- α: learning rate

Linear Regression

Parameters: $\theta = (\theta_0, \ldots, \theta_n)$

Hypothesis/Model: $h_\theta(x) = \theta(x) = \theta^T x = \theta_0 x_0 + \ldots + \theta_n x_n$

Cost Function: $J(\theta_0, \ldots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

$$\Rightarrow \frac{\partial}{\partial \theta_j} J(\theta_0, \ldots, \theta_n) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})\ x_j^{(i)}$$

Logistic Regression

Parameters: $\theta = (\theta_0, \ldots, \theta_n)$

Hypothesis/Model: $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$

Cost Function: $J(\theta_0, \ldots, \theta_n) = \frac{1}{m}\big[-y^{(i)}\ log(h_\theta(x^{(i)})) - (1-y^{(i)})\ log(1-h_\theta(x^{(i)}))\big]$

$$\Rightarrow \frac{\partial}{\partial\theta_j} J(\theta_0, \ldots, \theta_n) = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})\ x_j^{(i)}$$

Linear Regression and Logistic Regression via Gradient Descent

Repeat until convergence$\{$

$$\theta_j \leftarrow \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m} h_\theta(x^{(i)}) - y^{(i)})\ x_j^{(i)}$$

$\}$ (simultaneously update θ_j for all j)

Problem 1 [100 points]

Implement gradient descent algorithm for linear regression and answer the following questions. In this question, you are asked to use your gradient descent implementation to fit linear models to Auto data set which can be found in the "ISLR" package.

```
> require("ISLR")
> Auto
```

Initialization of the parameters

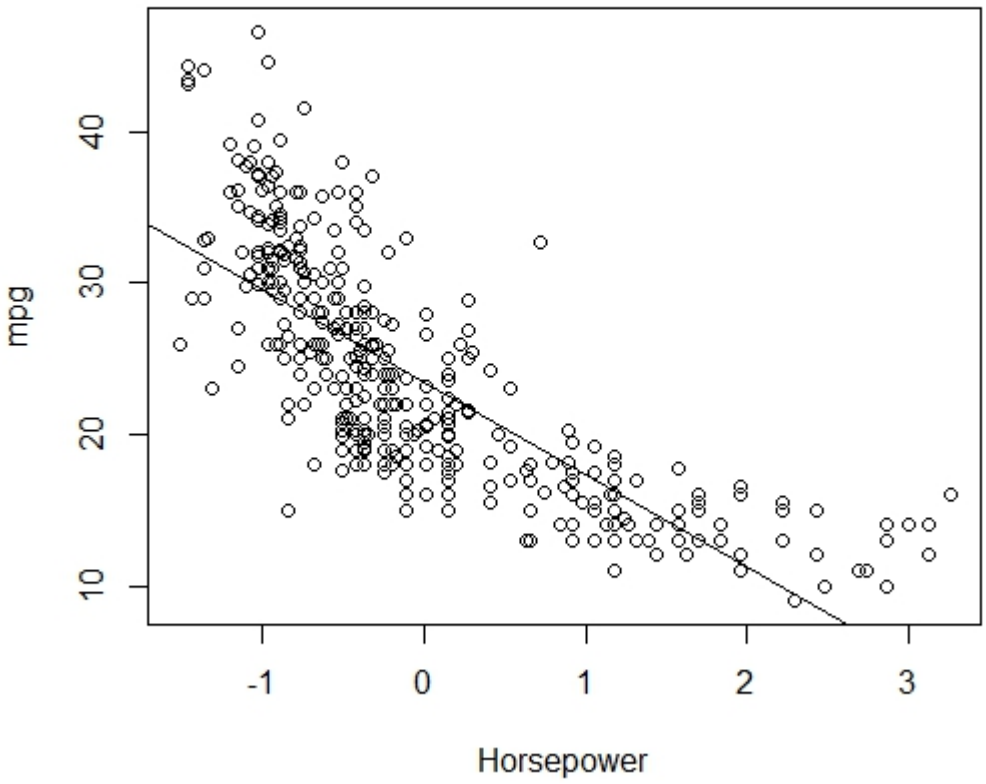
- Add $x_0 = 1$ to data (all ones feature)
- Set the learning rate α to 0.01 and iteration number to 1000. You may need to use different α and iteration number values if you observe they are not sufficient.
- Initialize θ's as 0's – $(\theta_0, \ldots, \theta_n) = (0, \ldots, 0)$

Simple Linear Regression [45 points]

1.1 Perform a simple linear regression with "mpg" as the response and "horsepower" as the predictor. What are the parameters $\theta = (\theta_0, \theta_1)$? Is the relationship between horsepower and mpg positive or negative? [20 pt]

After running it for 100 iterations with $\alpha = 1.5$, the value of cost function was 11.97183. $(\theta_0 = 23.445918), (\theta_1 = -6.075627)$. Horsepower and mpg are **negatively co-related** which is evident from the negative value of θ_1.

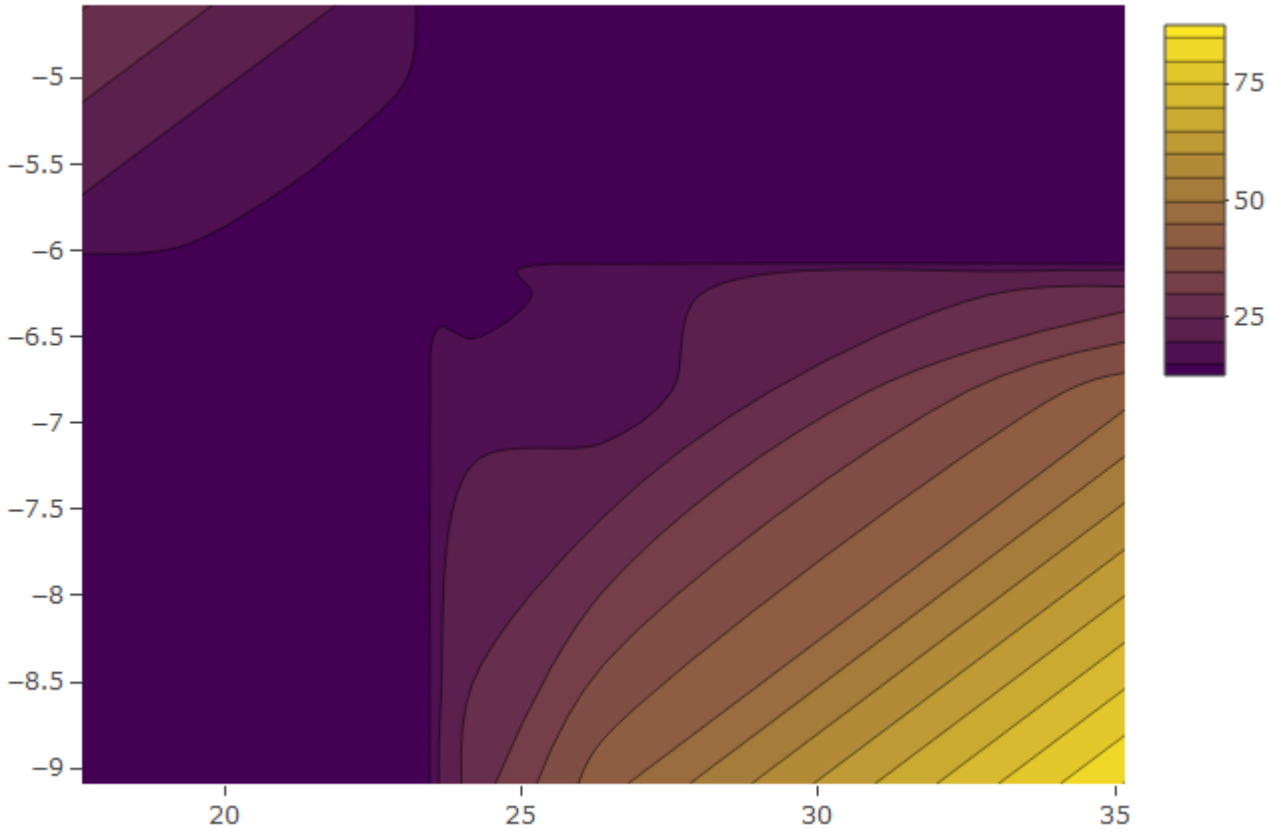
1.2 Plot the output variable and the input variable. Display the least squares regression line. [5 pt]



1.3 Use the model obtained in Q.1.1 to make predictions. What is the "mpg" value for "horsepower = 220"? [5 pt]

The value of mpg for horsepower = 220 is **5.43892**.

1.4 In a contour plot, show how $J(\theta)$ varies with changes in θ_0 and θ_1. Does $J(\theta)$ have a global minimum? [10 pt]]



From the graph we see that there is a global minimum in the center.

1.5 The closed-form solution to linear regression is $\theta = (\Delta^T \Delta)^{-1} \Delta^T y$. Report the coefficients using this formula. [5 pt]

The coefficients are $\theta_0 = 23.44592, \theta_1 = -6.075627$.

Multivariate Linear Regression [55 points]

1.6 First, perform feature scaling (mean normalization) over the Auto data set to make gradient descent converge faster. Then, train a multivariate linear regression with "mpg" as the response and all other variables except name as the predictors. Report the parameters (θ's). What does the coefficient for the "year" variable suggest? [30 pt]

The parameters in order are $(\theta_0, \ldots, \theta_8) =$
$(23.4459184, -0.8415931, 2.0819598, -0.6524692,$
$-5.4990690, 0.2222978, 2.7656487, 1.1487821)$.
The coefficient of year is "2.7656487". This indicates that "year" has a positive effect on "mpg". Every increase in year contributes by a factor of 2.76 to the "mpg". This kind of makes sense in the real world because the automobile industry tries its best year after year to improve the "mpg" factor of a car to reduce fuel cost.

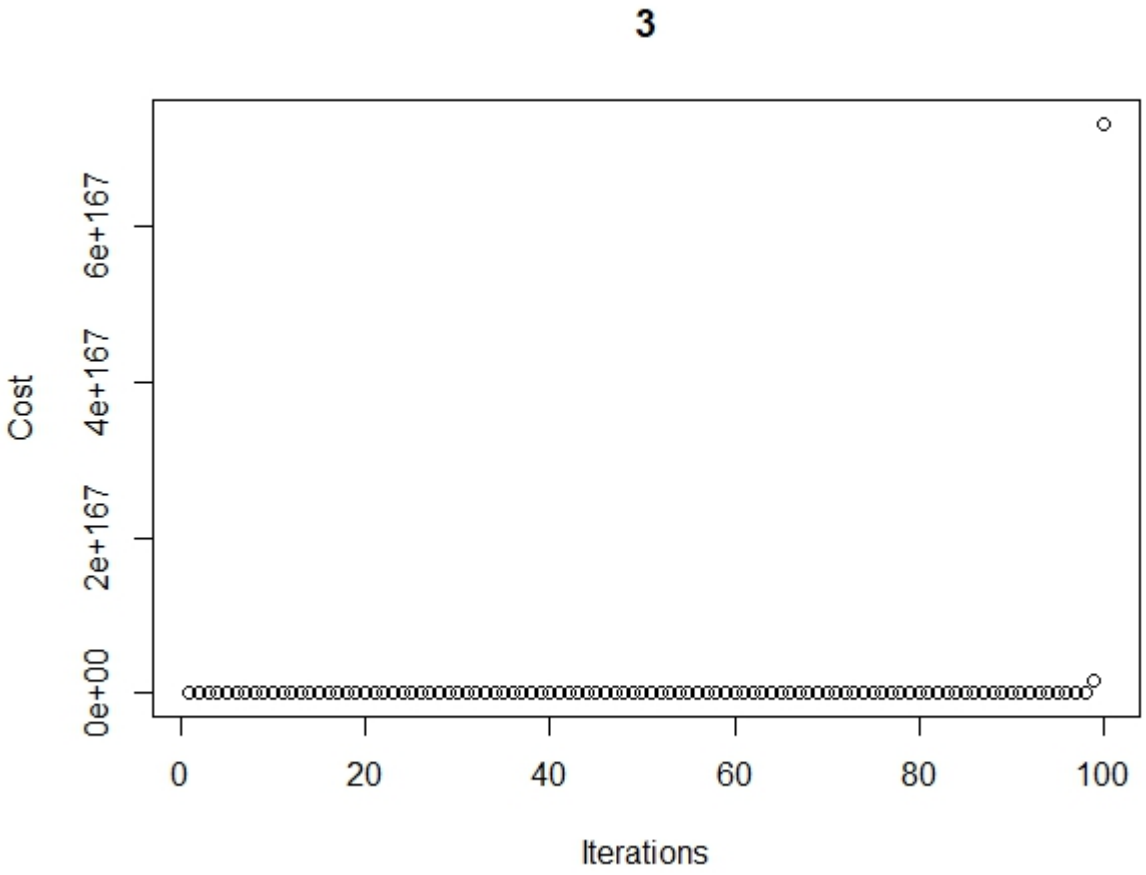
1.7 Use the model obtained in Q.1.6 to make predictions. What is the "mpg" value for $(x_1, \ldots, x_7) =$

$(4, 300, 200, 3500, 11, 70, 2)$? [5 pt]

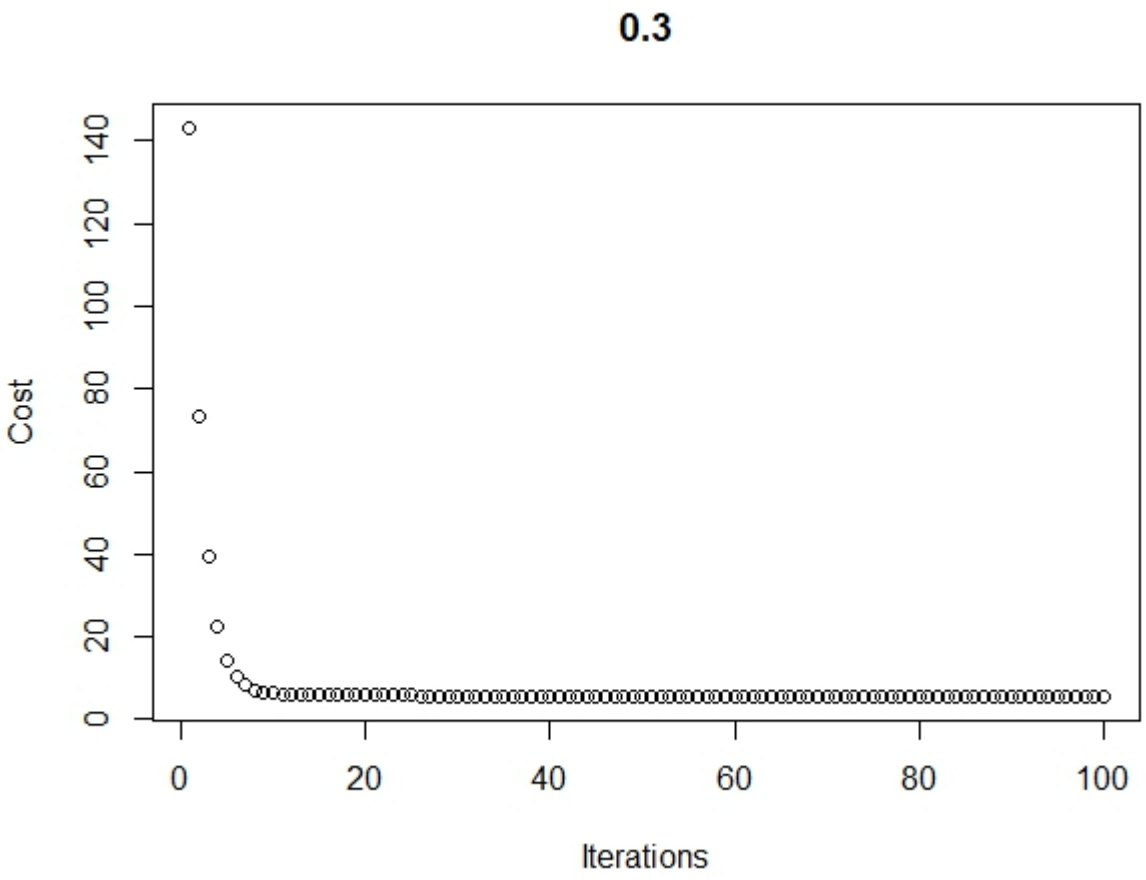
The "mpg" value is "17.05505".

1.8 In this question, you are asked to test different learning rates. Run your gradient descent for 100 iterations at the chosen learning rates ($\alpha_1 = 3, \alpha_2 = 0.3, \alpha_3 = 0.03, \alpha_4 = 0.00003$). For each learning rate, make a plot that shows how $J(\theta)$ changes at each iteration. Discuss the plots? i.e., which one looks better? does it converge? [15pt]

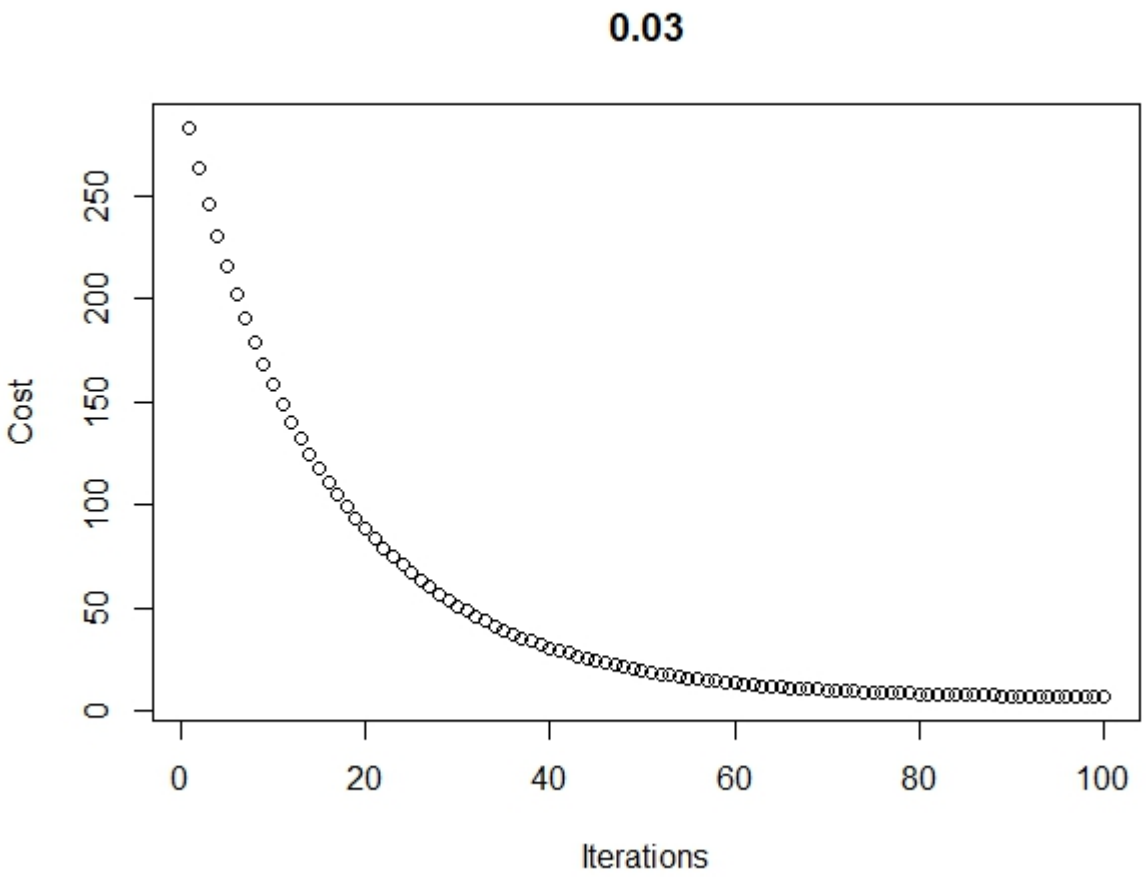
For $\alpha_1 = 3$, find the plot below.



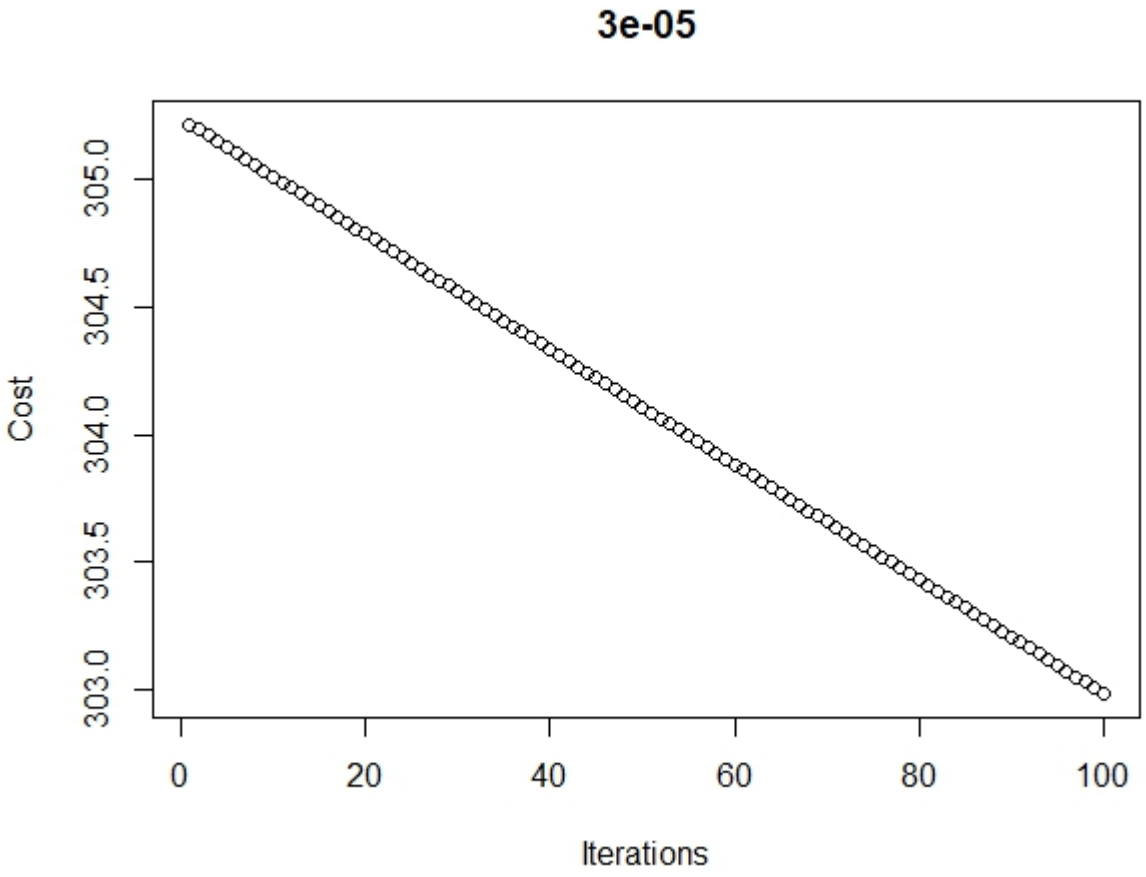
From the plot we can observe that the algorithm does not converge. Most of the values look like zero but they are not. Since the cost for 100th iteration is very high, the other values look like zero. This increase in cost function is because of the high learning rate.
For $\alpha_2 = 0.3$, find the plot below.



From the plot we can observe that the algorithm does converge. The convergence is also pretty fast. Looks like $\alpha_2 = 0.3$ is a very good learning rate.
For $\alpha_2 = 0.03$, find the plot below.



From the plot we can observe that the algorithm does converge. The convergence is slow compared to previous case. Looks like $\alpha_2 = 0.03$ is a slow learning rate.
For $\alpha_2 = 0.00003$, find the plot below.



From the plot we can observe that the algorithm does not converge in 100 iterations. The learning rate is very very slow.

1.9 Calculate the coefficients using the normal equations. [5pt]

The coefficients in order are $(\theta_0, \ldots, \theta_8) =$
$(23.4459184, -0.8415931, 2.0819598, -0.6524692,$
$-5.4990690, 0.2222978, 2.7656487, 1.1487821)$.

Problem 2 [35 points]

From textbook, Chapter 3 exercises 13, 14 and 15 (Pages 124-126).

13.a

R code is below:

```
set.seed(1)
x=rnorm(100)
```

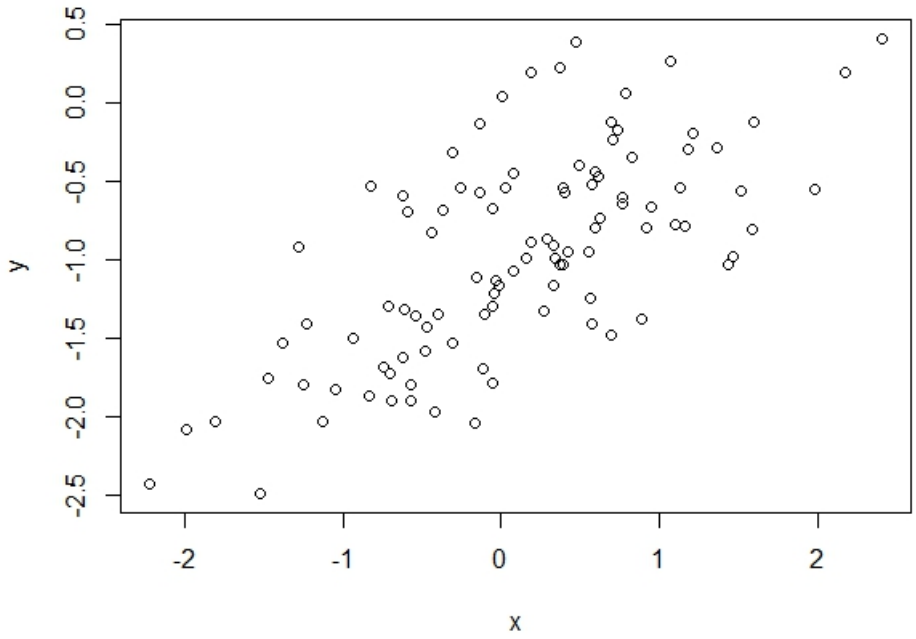
13.b

eps=rnorm(100,0,0.5)

13.c

The length of 'y' is 100.
The coefficients are $\theta_0 = -1, \theta_1 = 0.5$

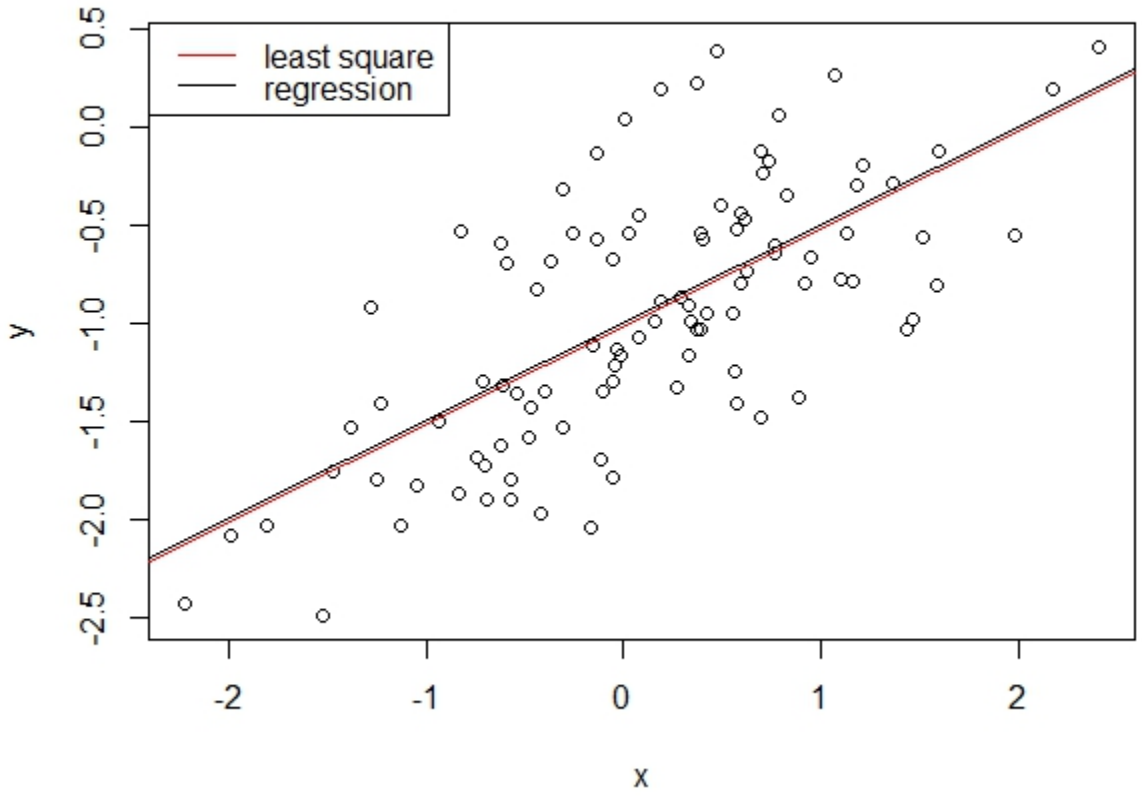
13.d



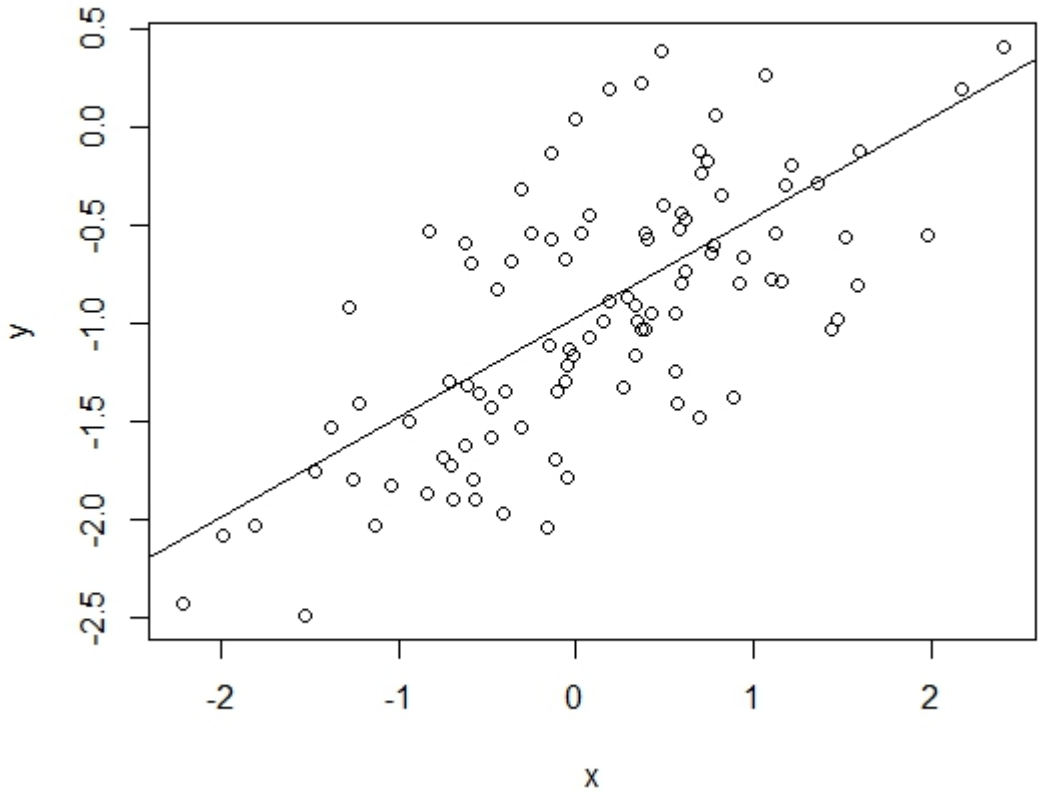
x and y are almost linearly co-related but there are some outliers.

13.e
model =lm(y~x)
parameters=model$coefficients
The parameters are -1.0188463 and 0.4994698 which are very close to the actual parameters -1 and 0.5

13.f
The plot is shown below:



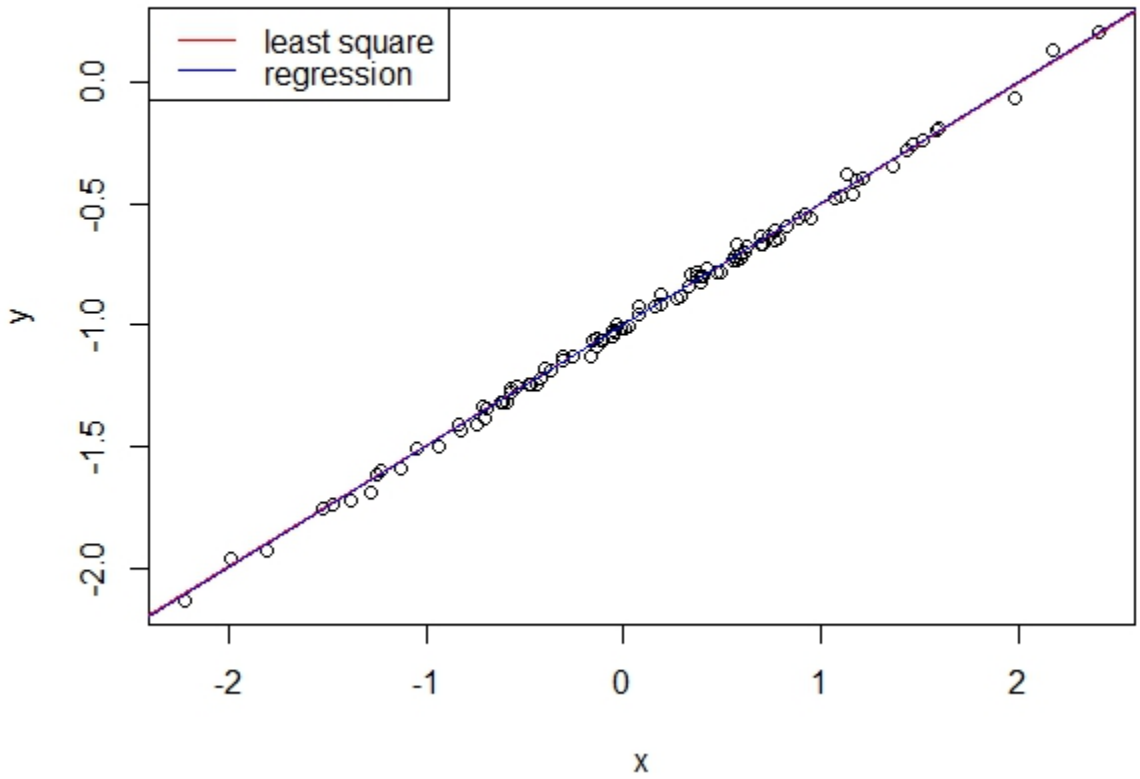
13.g



The P-value of the quadratic term is 0.164 which is greater than 0.05. Hence, there is no strong evidence that quadratic term improves the model.

13.h

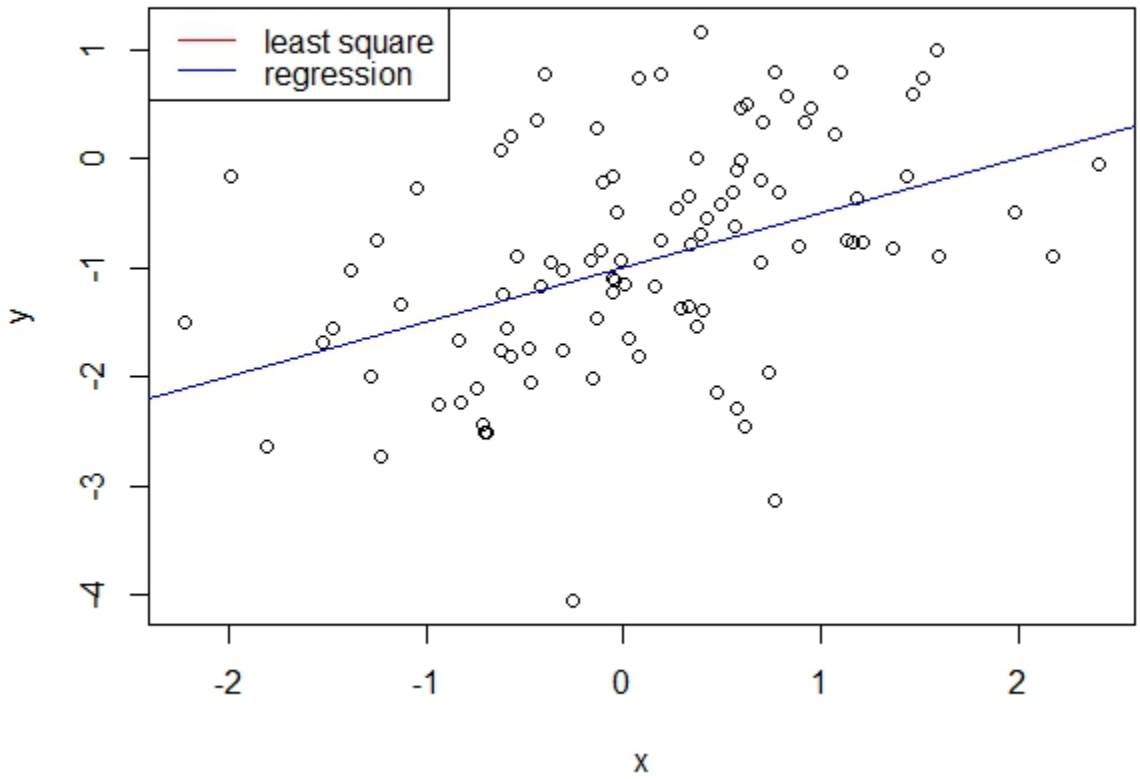
By reducing the variance of the normal distribution to 0.02, I got the below model



We can see that the model matches the actual coefficients. Hence the lines overlap.

13.i

By increasing the variance of the normal distribution to 0.9, I got the below model



Even though both the lines are very close to each other, the relationship is no longer linear. The data points are very scattered.

13.j

For the original model, find the confidence intervals below:

```
> confint(model_original)
                 2.5 %     97.5 %
(Intercept) -1.1150804 -0.9226122
x            0.3925794  0.6063602
```

For the model with noise, find the confidence intervals below:

```
> confint(model_noise)
                 2.5 %     97.5 %
(Intercept) -1.0036083 -0.9952970
x            0.4958075  0.5050391
>
```

For the model with less noise, find the confidence intervals below:

```
> confint(model_less_noise)
                 2.5 %     97.5 %
(Intercept) -1.1623715 -0.7883647
x            0.3113392  0.7267611
```

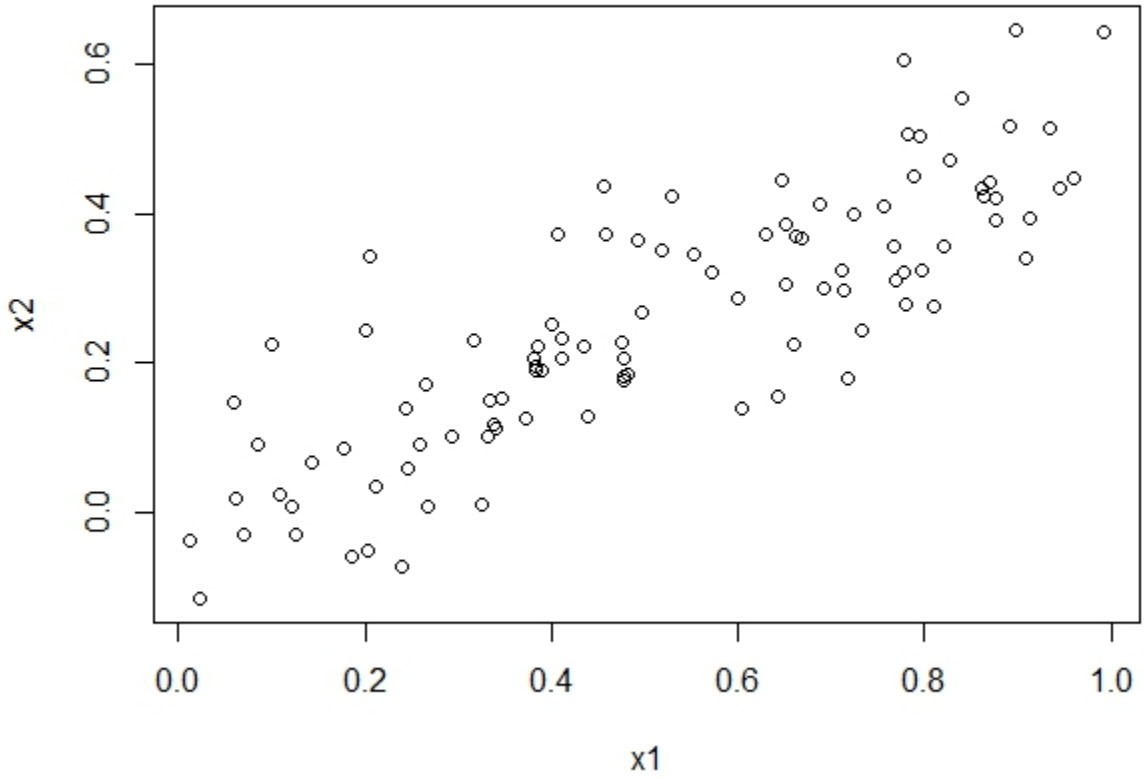
From the above results, we see that the confidence interval widens if we induce more noise in the data. The confidence interval is least for the second set above since the standard deviation was very small.

14.a

The coefficients are $\theta_0 = 2, \theta_1 = 2, \theta_2 = 0.3$

14.b

The correlation between x1 and x2 is : **0.8351212**. They seem to have good correlation. The scatter plot is below.



14.c

```
> summary(model)

Call:
lm(formula = y ~ x1 + x2)

Residuals:
    Min      1Q  Median      3Q     Max
-2.8311 -0.7273 -0.0537  0.6338  2.3359

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.1305     0.2319   9.188 7.61e-15 ***
x1            1.4396     0.7212   1.996   0.0487 *
x2            1.0097     1.1337   0.891   0.3754
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.056 on 97 degrees of freedom
Multiple R-squared:  0.2088,    Adjusted R-squared:  0.1925
F-statistic:  12.8 on 2 and 97 DF,  p-value: 1.164e-05
```

Only the predicted $\beta_0 = 2.13$ is close to the actual $\beta_0 = 2$ whereas the predicted $\beta_1 = 1.44, \beta_2 = 1.01$ are very different than the actual $\beta_1 = 2, \beta_2 = 0.3$.
Since the P-value $0.0487 < 0.05$, we may reject the null hypothesis for β_0.
Since the P-value $0.3754 > 0.05$, we cant reject the null hypothesis for $beta_1$.

14.d

```
> summary(model)

Call:
lm(formula = y ~ x1)

Residuals:
     Min       1Q   Median       3Q      Max
-2.89495 -0.66874 -0.07785  0.59221  2.45560

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.1124     0.2307   9.155 8.27e-15 ***
x1            1.9759     0.3963   4.986 2.66e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.055 on 98 degrees of freedom
Multiple R-squared:  0.2024,    Adjusted R-squared:  0.1942
F-statistic: 24.86 on 1 and 98 DF,  p-value: 2.661e-06
```

The p-value is $2.66e-06$ which is less than 0.05. Hence we reject the null hypothesis.

14.e

```
> summary(model)

Call:
lm(formula = y ~ x2)

Residuals:
     Min       1Q   Median       3Q      Max
-2.62687 -0.75156 -0.03598  0.72383  2.44890

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.3899     0.1949   12.26  < 2e-16 ***
x2            2.8996     0.6330    4.58 1.37e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.072 on 98 degrees of freedom
Multiple R-squared:  0.1763,    Adjusted R-squared:  0.1679
F-statistic: 20.98 on 1 and 98 DF,  p-value: 1.366e-05
```

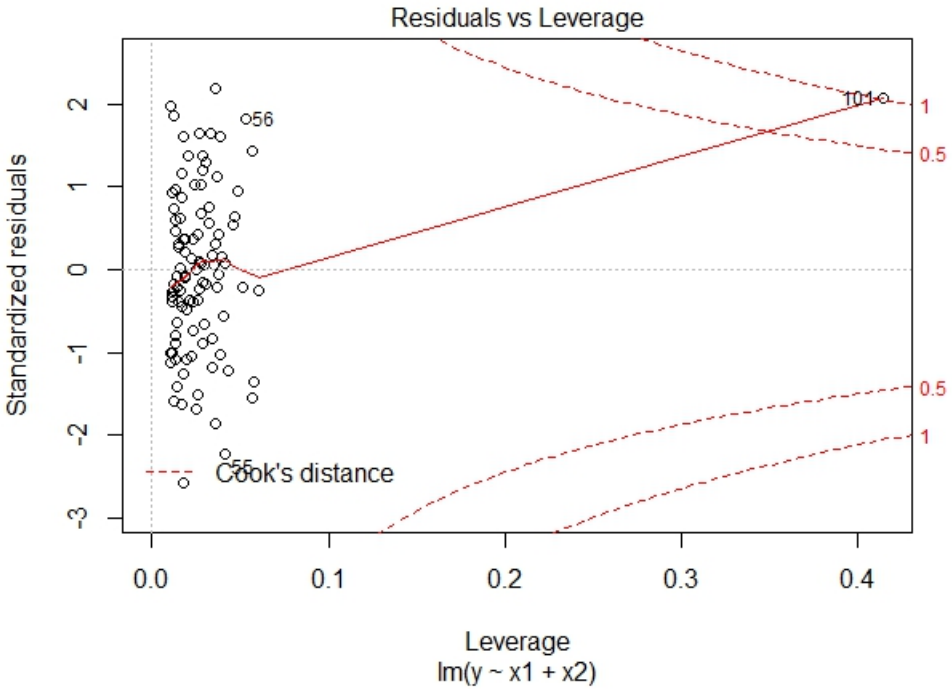
The p-value is $1.37e-05$ which is less than 0.05. Hence we reject the null hypothesis.

14.f

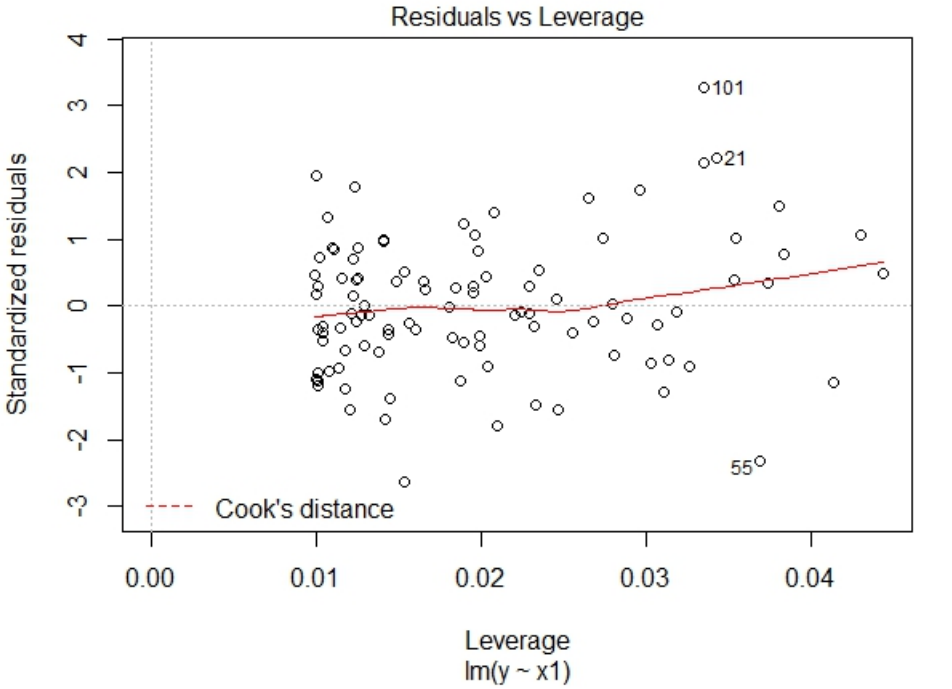
No, the results do not contradict each other. Due to high correlation between x1 and x2, it is difficult to figure out how each of them affects the response variable individually.

14.g

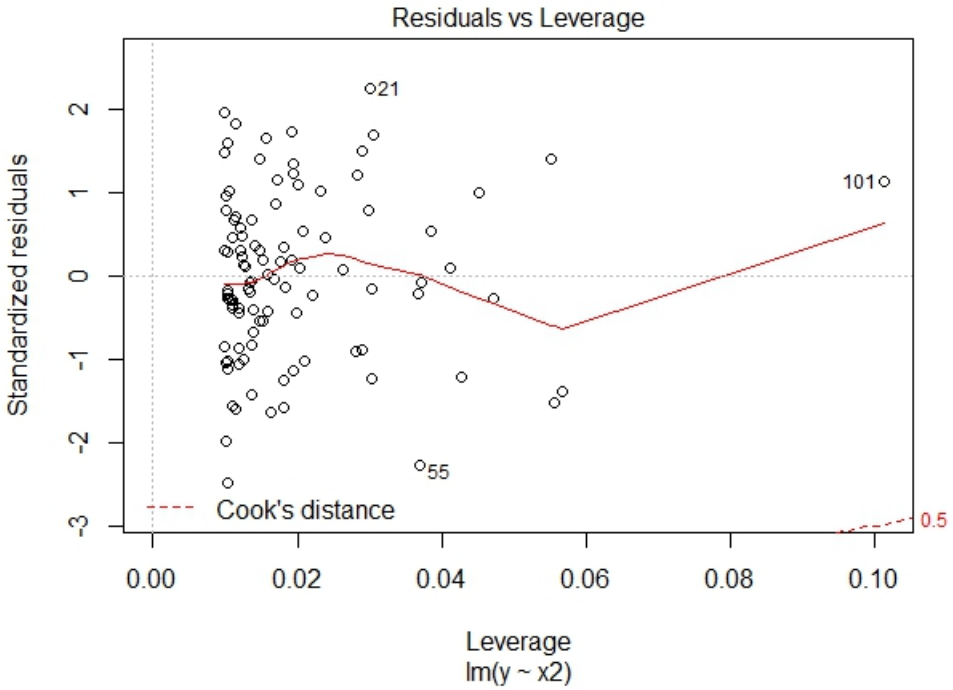
Find the plot below of y = x1+x2



From the above plot, we can see that the **101st point is a high leverage point**.
Find the plot below of y = x1



From the above plot, we can see that the **101st point is an outlier**.
Find the plot below of y = x2



From the above plot, we can see that the **101st point is a high leverage point**.

15.a

```
> model = lm(crim~chas,data=Boston)
> summary(model)

Call:
lm(formula = crim ~ chas, data = Boston)

Residuals:
   Min     1Q Median     3Q    Max
-3.738 -3.661 -3.435  0.018 85.232

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.7444     0.3961   9.453   <2e-16 ***
chas         -1.8928     1.5061  -1.257    0.209
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.597 on 504 degrees of freedom
Multiple R-squared:  0.003124,  Adjusted R-squared:  0.001146
F-statistic: 1.579 on 1 and 504 DF,  p-value: 0.2094
```

All predictors have P-value less than 0.05 except the predictor "chas". This says that each of the predictors are significant for prediction except "chas".

15.b

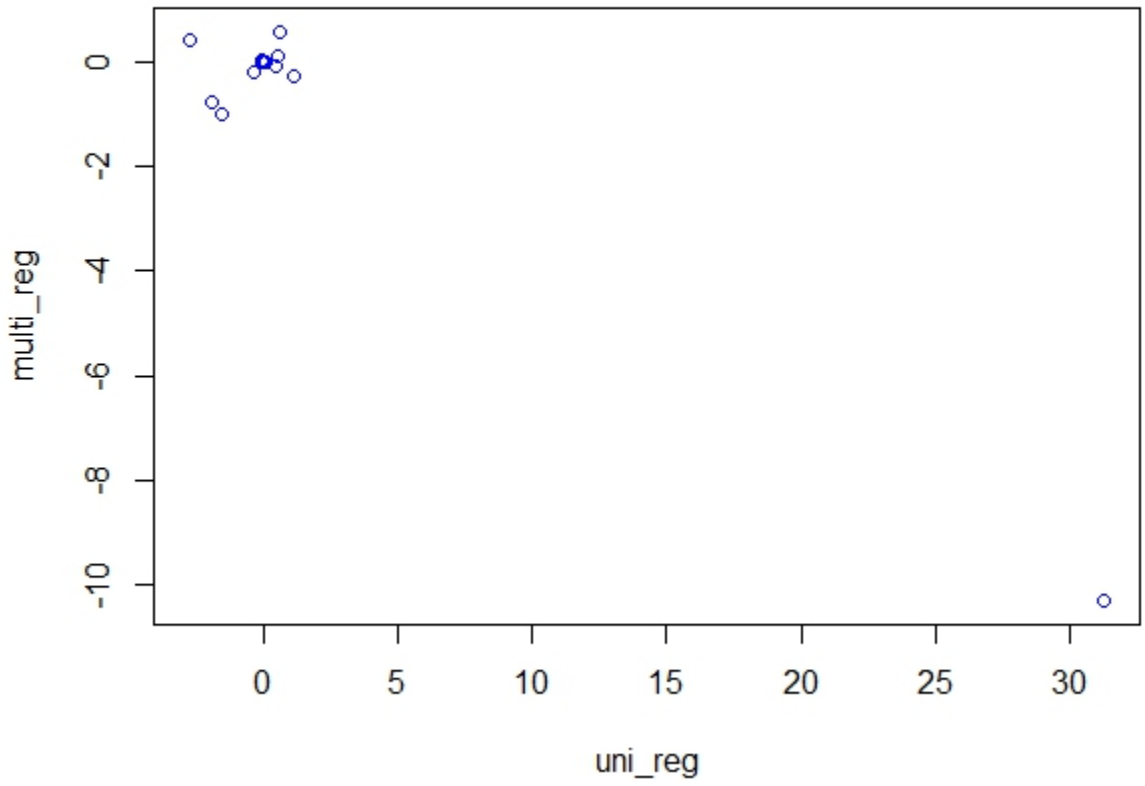
Find the summary of the model below,

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  17.033228   7.234903   2.354 0.018949 *
zn            0.044855   0.018734   2.394 0.017025 *
indus        -0.063855   0.083407  -0.766 0.444294
chas         -0.749134   1.180147  -0.635 0.525867
nox         -10.313535   5.275536  -1.955 0.051152 .
rm            0.430131   0.612830   0.702 0.483089
age           0.001452   0.017925   0.081 0.935488
dis          -0.987176   0.281817  -3.503 0.000502 ***
rad           0.588209   0.088049   6.680 6.46e-11 ***
tax          -0.003780   0.005156  -0.733 0.463793
ptratio      -0.271081   0.186450  -1.454 0.146611
black        -0.007538   0.003673  -2.052 0.040702 *
lstat         0.126211   0.075725   1.667 0.096208 .
medv         -0.198887   0.060516  -3.287 0.001087 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

From the summary, we can reject the null hypothesis for ”zn”, ”dis”,”rad”,”black”,”medv” since their p-values are less than 0.05

15.c

Find the plot below



15.d

```
> model = lm(crim~chas,data=Boston)
> summary(model)

Call:
lm(formula = crim ~ chas, data = Boston)

Residuals:
   Min     1Q Median     3Q    Max
-3.738 -3.661 -3.435  0.018 85.232

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.7444     0.3961   9.453   <2e-16 ***
chas         -1.8928     1.5061  -1.257    0.209
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 8.597 on 504 degrees of freedom
Multiple R-squared:  0.003124,  Adjusted R-squared:  0.001146
F-statistic: 1.579 on 1 and 504 DF,  p-value: 0.2094
```

By using summary of each model, I got the below results

For "zn", "rm", "rad", "tax" and "lstat", **the cubic co-efficients suggest that they are not significant**. For "indus", "nox, "age", "dis", "ptratio" and "medv", **both the quadratic and cubic co-efficients have significant p-values**. Lastly, for "black" predictor, **neither the quadratic or cubic p-values are significant**.

Problem 3 [65 points]

Implement gradient descent algorithm for logistic regression and answer the following questions. In this question, you are asked to use your gradient descent implementation to train logistic regression models over Auto data set.

Initialization of the parameters

- Add $x_0 = 1$ to data (all ones variable)
- Set the learning rate α to 0.01 and iteration number to 1000. You may need to use different α and iteration number values if you observe they are not sufficient.
- Initialize θ's as 0's – $(\theta_0, \ldots, \theta_n) = (0, \ldots, 0)$

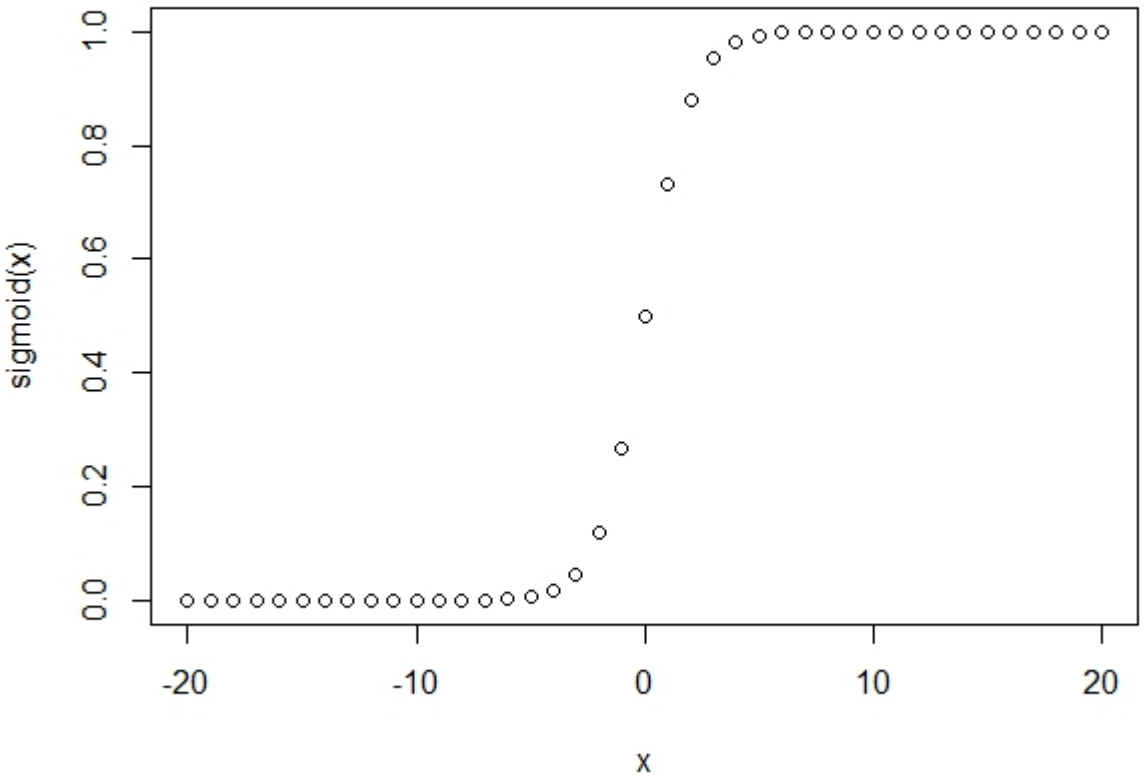
Data Preprocessing

- Create a binary variable, "mpg01", that contains a 1 if mpg contains a value above its median, and a 0 if mpg contains a value below its median
- Create a new data set called new.Auto by extracting the features mpg01, cylinders, displacement, horsepower, weight from Auto data set. Keep the order of the features as shown here in the new.Auto data set
- Do feature scaling (mean normalization) over the input variables of the new.Auto data set to make gradient descent converge faster.
- Use the new.Auto data set to answer the following questions

Logistic Regression

3.1 Implement the sigmoid function and make a plot of it by testing different inputs. $\left(g(z) = \frac{1}{1+e^{-z}}\right)$ [5 pt]

Below is graph of sigmoid function for values between $[-20, 20]$



3.2 Perform logistic regression on the new.Auto data set in order to predict "mpg01" using the input variables: cylinders, displacement, horsepower, weight. Report the parameters (θ's). [30 pt]

Below parameters were obtained after running logistic regression for 800 iterations with a learning rate of 3.
The parameters in order are $(\theta_0, \ldots, \theta_4) =$
$(-0.99302408, -0.02313124, -1.35660261, -1.62137416, -1.65348742)$

3.3 What is the error of the model over new.Auto data set? [10 pt]

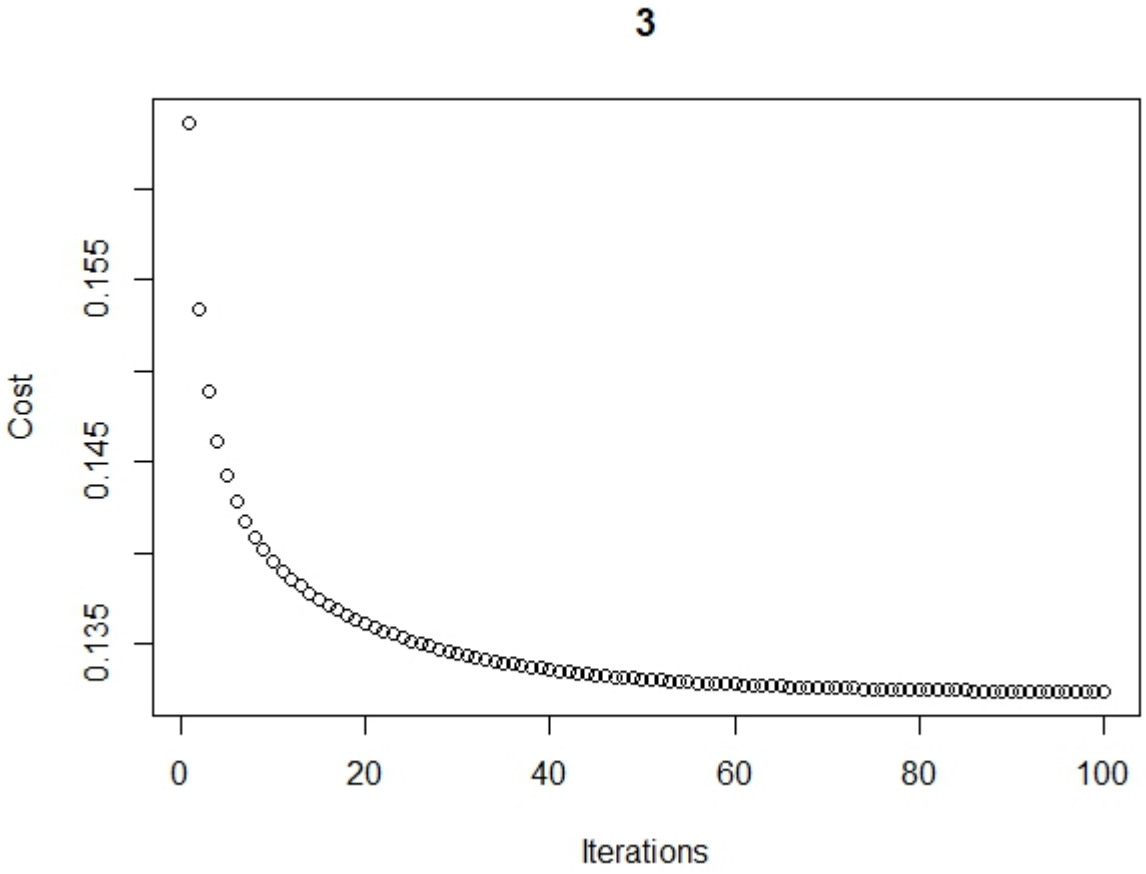
After running it for 800 iterations with $\alpha = 3$, the cost function was 0.1321863. To see how good the model performs, I use the following mis-classification methodology:
I count the number of wrong predictions the model made on the actual answers. The lesser the wrong predictions, the better the model is. The predictions from the model are as follows:
If the model predicts a value greater than or equal to 0.5 then its 1 else its 0.
With this methodology, i got **"zero(0)"** mis-classification rate.

3.4 Use the model obtained in Q.3.1 to make predictions. What is the "mpg01" value for $(x_1, \ldots, x_4) = (8, 340, 200, 3500)$ (first, scale the data point with the parameters obtained earlier while normalizing the features) [5 pt]

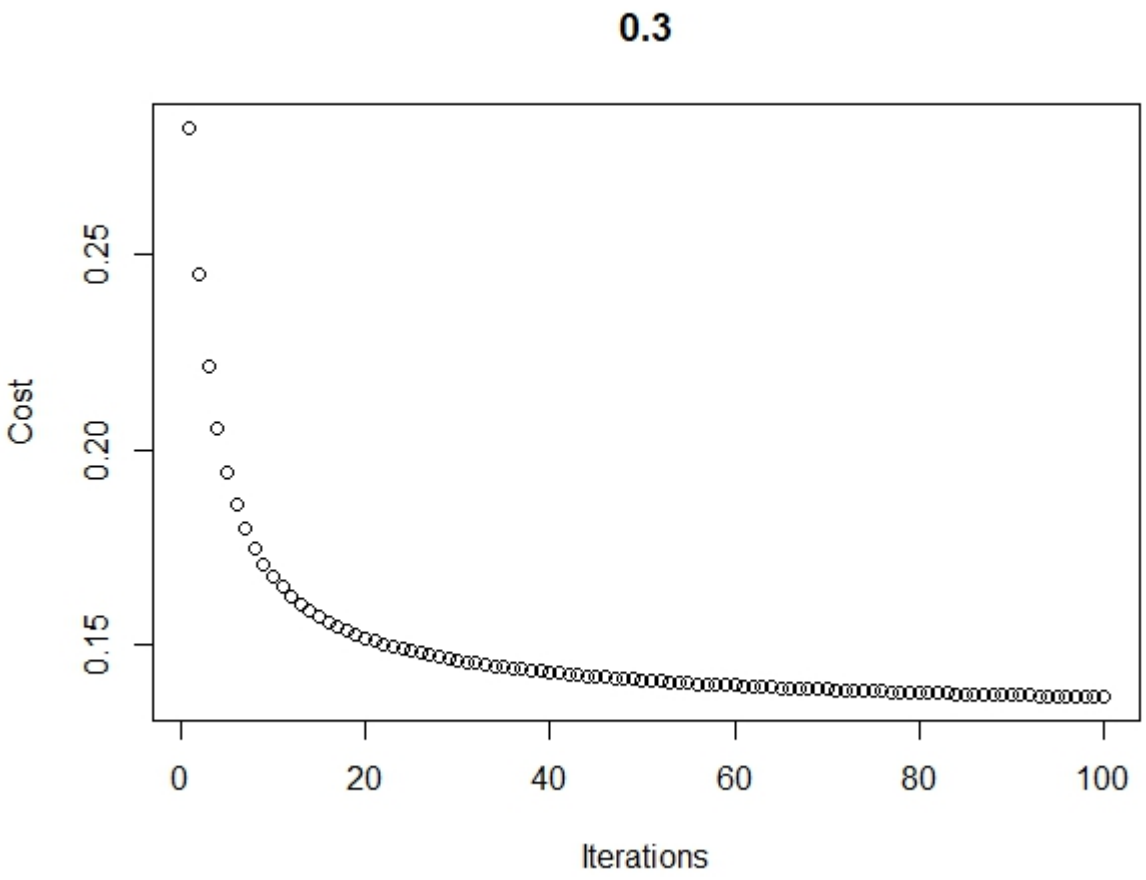
The model predicted the mpg01 values as "0".

3.5 In this question, you are asked to test different learning rates. Run your gradient descent for 100 iterations at the chosen learning rates ($\alpha_1 = 3, \alpha_2 = 0.3, \alpha_3 = 0.03, \alpha_4 = 0.00003$). For each learning rate, make a plot that shows how $J(\theta)$ changes at each iteration. Discuss the plots? i.e., which one looks better (faster)? does it converge? [15pt]

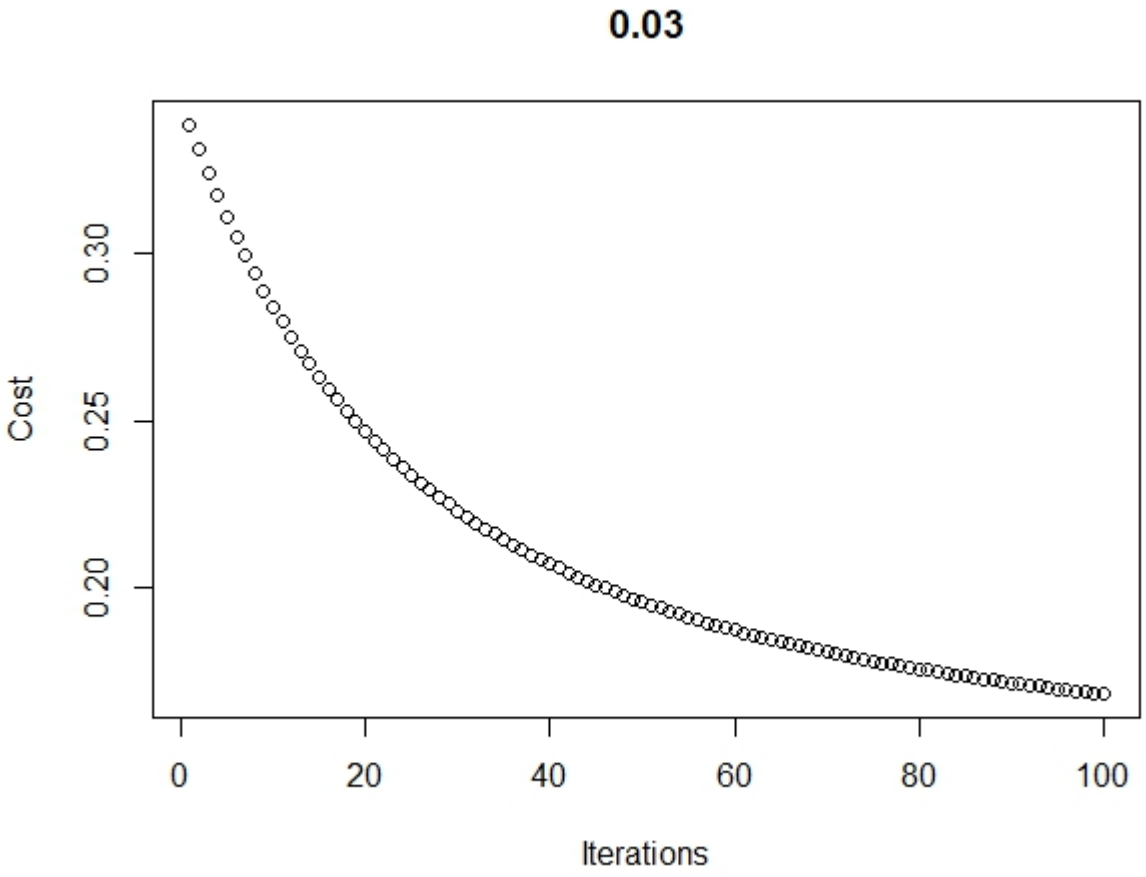
For $\alpha_1 = 3$, find the plot below.



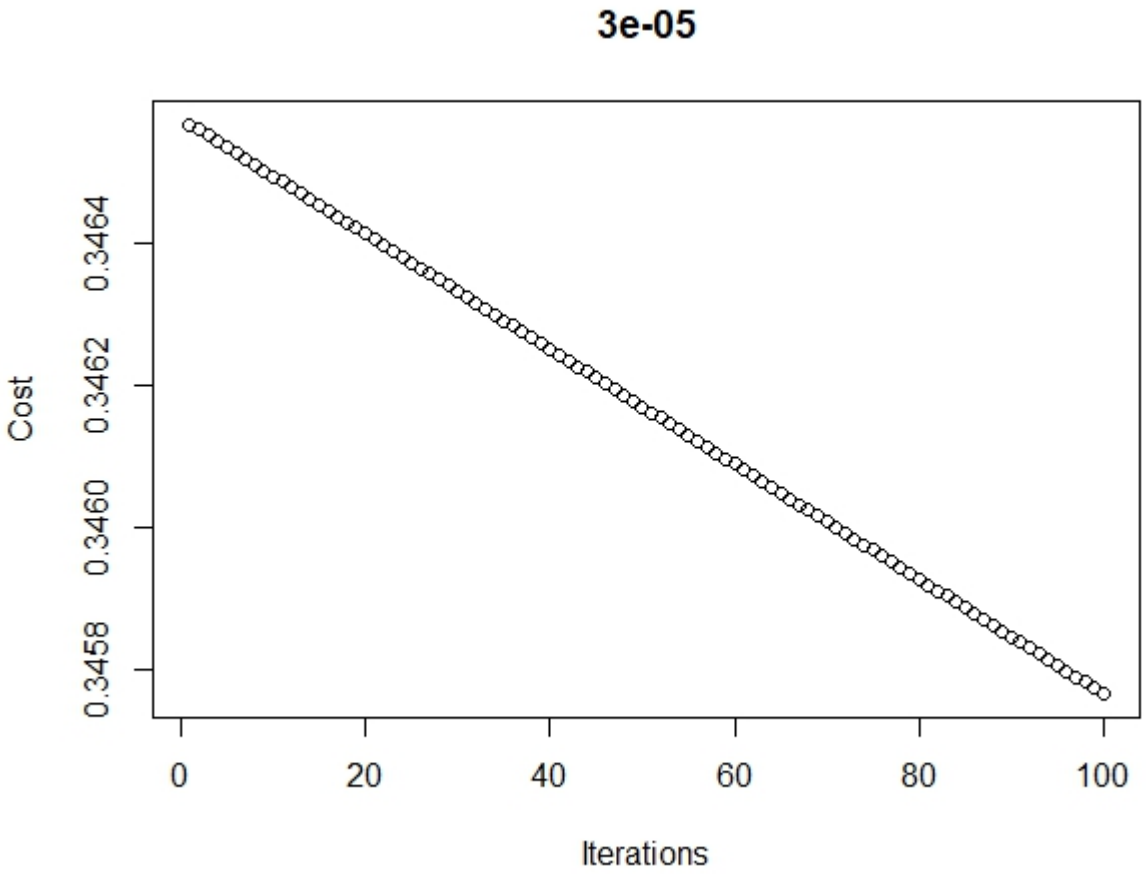
From the plot we can observe that the algorithm does "almost" converge but not completely. There is room for improvement. If we run it for more iterations, it will converge completely. But overall, its a good fit with this learning rate.
For $\alpha_2 = 0.3$, find the plot below.



From the plot we can observe that the algorithm does not converge completely. We need to run it for many iterations to converge. Basically, the learning rate is pretty less. Hence the slower convergence. For $\alpha_2 = 0.03$, find the plot below.



From the plot we can observe that the algorithm does not converge. The learning rate is very less. For it to converge, we need to run a lot many iterations.
For $\alpha_2 = 0.00003$, find the plot below.



From the plot we can observe that the algorithm does not converge in 100 iterations. The learning rate is very very slow.

What to Turn-in – Submission Instructions

- Zip the files requested below for your submission. The zipped folder should be named as "usename-section number", i.e, hakurban-P556
  - The *tex and *pdf of the written answers to this document
  - *`R`files for:
    - Gradient descent implementation for problems 1.1 and 1.6 – file name: "*linearRegression.R*".
    - Normal equation implementation for problems 1.5 and 1.9 – file name: "*normalSolution.R*".
    - Gradient descent implementation for problem 3.2 – file name: "*logisticRegression.R*".
    - new.Auto data set – file name: "*newAuto.R*".
  - A README file that explains how to run your code and other files in the folder