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

Chuck Jia

October 28, 2017

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]

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]

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]

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]

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]

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]

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]

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

Simple Linear Regression

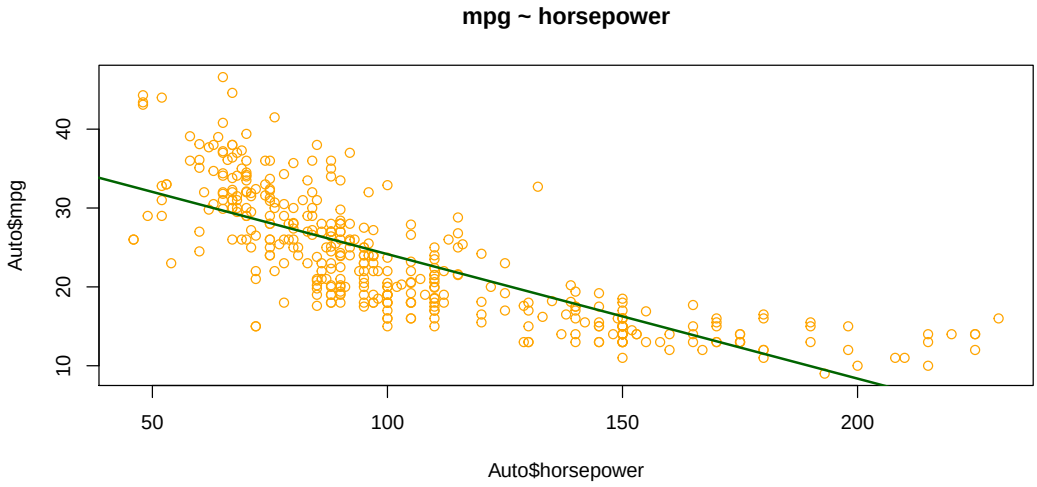
(1.1) The linear regression is performed by commands in `Prob1-Q1.R`. To run the file, execute all commands in the file directly, or execute the command in the first part of the file `Prob1-Part1-SimpleLR.R`.

The result of the regression is

$$\hat{\theta} = (\hat{\theta}_0, \hat{\theta}_1)^T = (39.93586, -0.1578447)^T$$

From this, we can see that the relationship between horsepower and mpg is negative, because the slope $\hat{\theta}_1 = -0.1578447$ is negative.

(1.2)

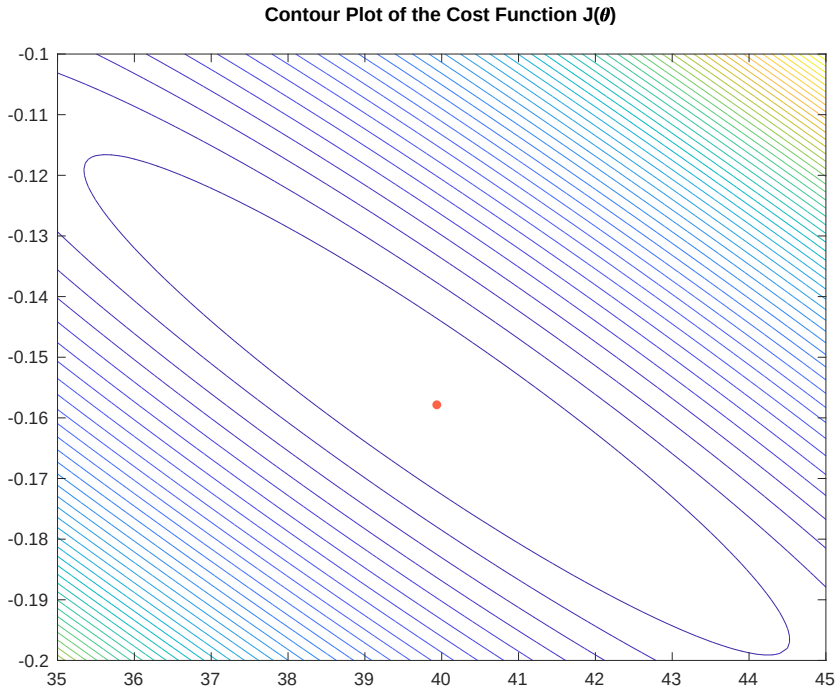


In the plot, the green line represents the least squares regression line.

(1.3) Using the model obtained in Q.1.1, when horsepower = 220,

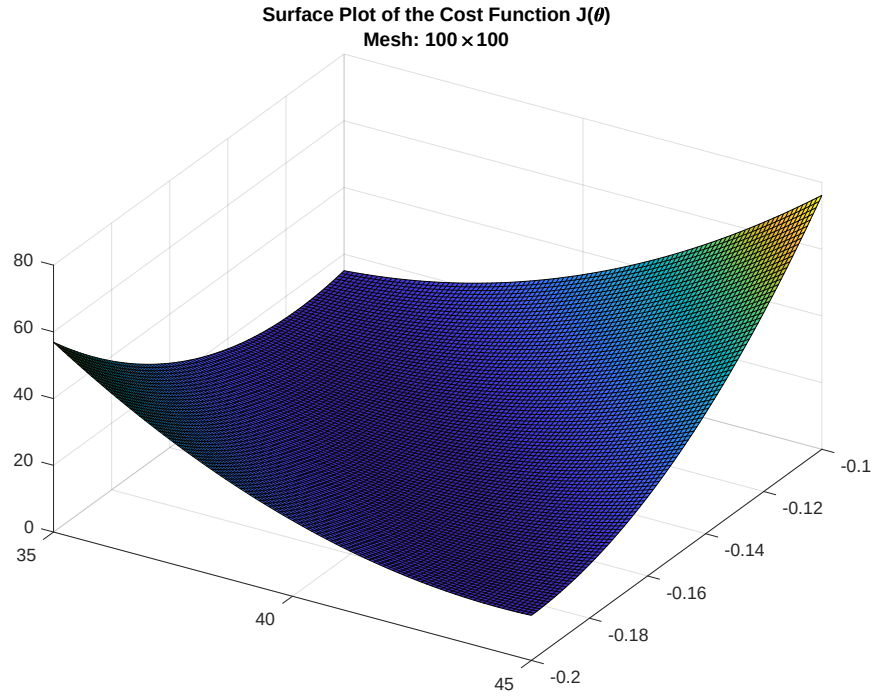
$$\hat{\text{mpg}} = \hat{\theta}_0 + \hat{\theta}_1 \times 220 = 39.93586 - 0.1578447 \times 220 = 5.210026$$

(1.4) The contour plot is shown below



In the plot, the orange dot in the middle represents the solution from the gradient descent method. The transition of level curve color from yellow to purple represents the descending of the cost function $J(\theta)$.

For convenience of analysis, we also provide a surface plot for the cost function.



(1.5) The result calculated by the closed form formula is

$$\hat{\theta} = (\hat{\theta}_0, \hat{\theta}_1)^T = (39.93586, -0.1578447)^T$$

Multivariate Linear Regression

(1.6) The parameters in the result of the gradient descent solver of the linear regression is

$$\hat{\theta} = (\hat{\theta}_0, \hat{\theta}_1, \hat{\theta}_2, \cdots, \hat{\theta}_7)^T = \begin{pmatrix} 5.741574 \times 10^{-16} \\ -0.1078237 \\ 0.2667399 \\ -0.08359488 \\ -0.704555 \\ 0.02848128 \\ 0.3543428 \\ 0.1471847 \end{pmatrix}$$

In `R`, the result with the corresponding labels is

```
                    [,1]
Intercept     5.741574e-16
cylinders    -1.078237e-01
displacement  2.667399e-01
horsepower   -8.359488e-02
weight       -7.045550e-01
```

```
acceleration  2.848128e-02
year          3.543428e-01
origin        1.471847e-01
```

The coefficient for "year" θ_{year} is 0.3543428. It represents the "slope" along the feature "year" direction of the regression affine hyperplane. It shows how fast the mpg value would increase when the year value increases. In other words, it is the correlation between mpg and year in the regression model. It suggests that in the regression model we derived, the scaled mpg value would increase by 0.3543428, if the scaled feature "year" increases its value by 1.

Moreover, if we scale back and use the original units, then this coefficient would suggest that if we increase the value of the feature "year" by 1, the mpg would increase its value in the original unit by

$$\hat{\theta}_{\text{year}}\sigma_{\text{mpg}}\sigma_{\text{year}}^{-1} = 0.7507725$$

where σ_{mpg} and σ_{year} are the sample standard deviations of features "mpg" and "year" without scaling.

(1.7) The scaled values for x is

$$\begin{aligned}&(x_1^{\text{scaled}}, x_2^{\text{scaled}}, \ldots, x_7^{\text{scaled}})\\ &= (-0.8629108, 1.0090211, 2.4818845, 0.6150391, -1.6460856, -1.6232409, 0.5257105)\end{aligned}$$

This is performed by subtracting each feature element by the original feature sample mean, and then dividing each feature element by the original feature sample standard deviation.

Using the formula, the predicted value of mpg is

$$\hat{\text{mpg}}^{\text{scaled}} = (1, x_1^{\text{scaled}}, \cdots, x_7^{\text{scaled}})\hat{\theta} = (1, x_1^{\text{scaled}}, \cdots, x_7^{\text{scaled}})(\hat{\theta}_0, \hat{\theta}_1, \hat{\theta}_2, \cdots, \hat{\theta}_7)^T = -0.8233031$$

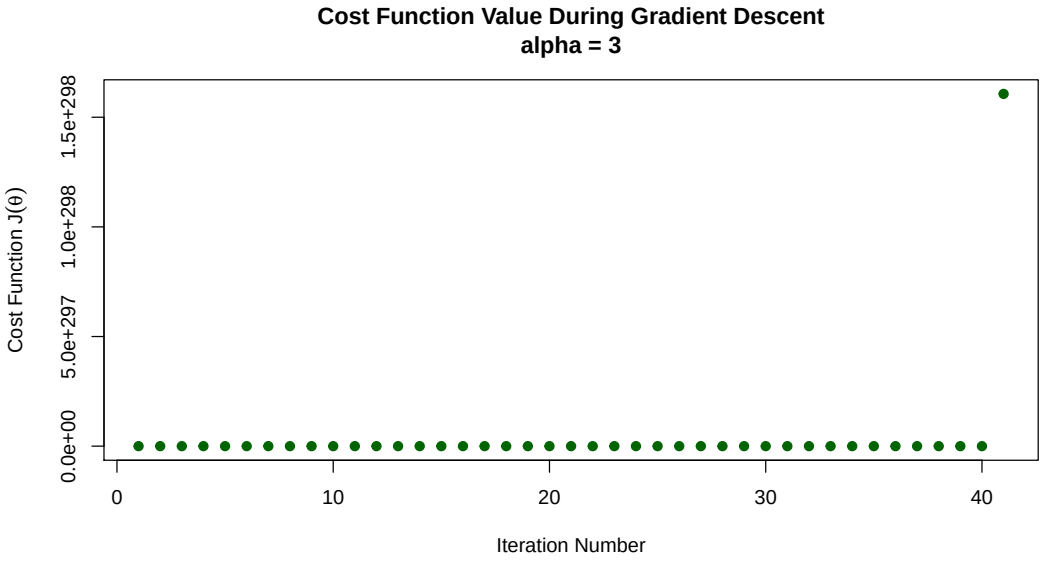
Now scale back to the original unit, we have

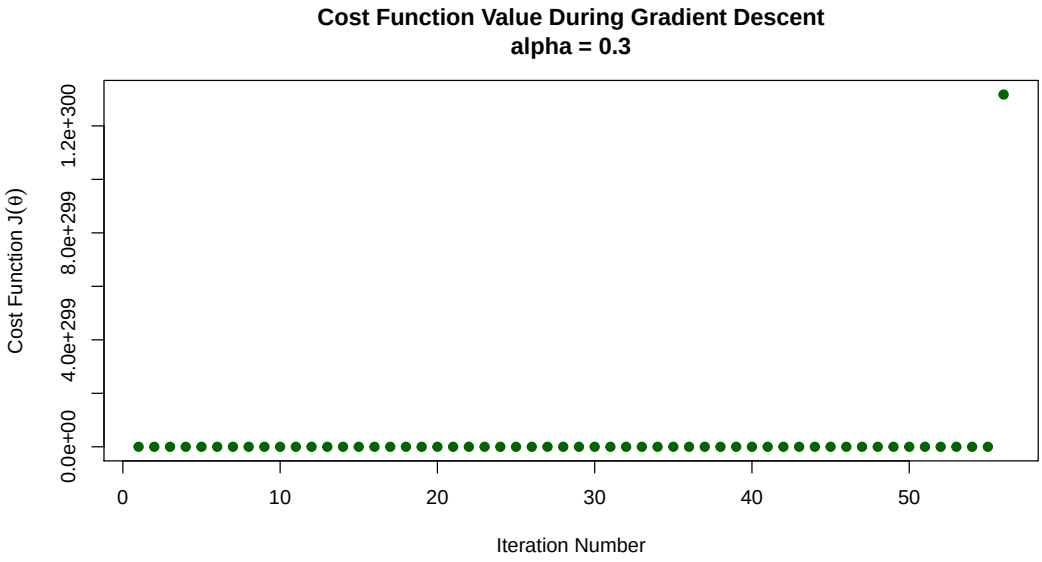
$$\hat{\text{mpg}} = \hat{\text{mpg}}^{\text{scaled}}\sigma_{\text{mpg}} + \mu_{\text{mpg}} = 17.02003$$

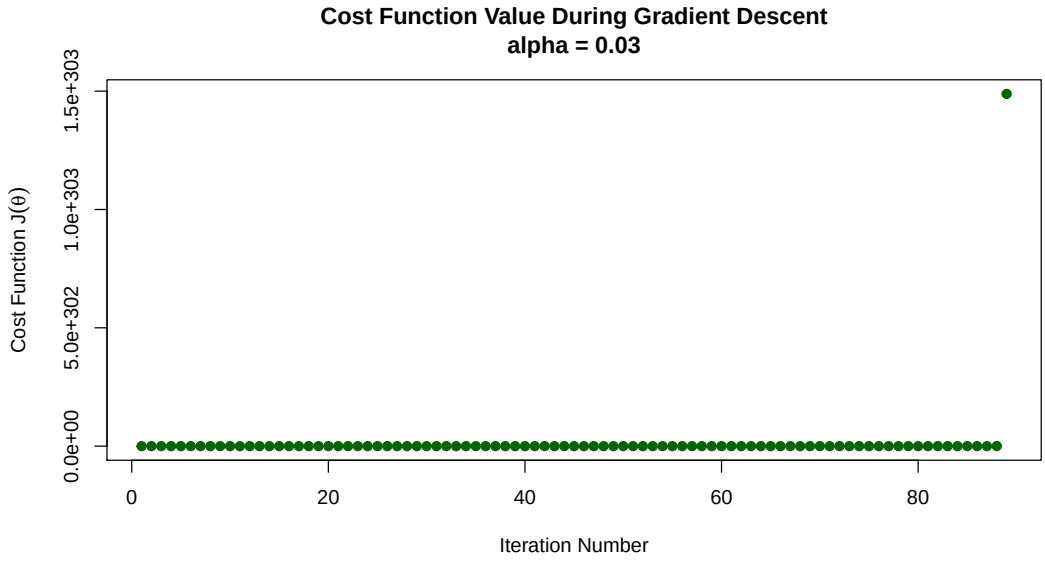
σ_{mpg} and μ_{mpg} are the sample standard deviation and sample mean of feature "mpg" in its original unit without scaling.

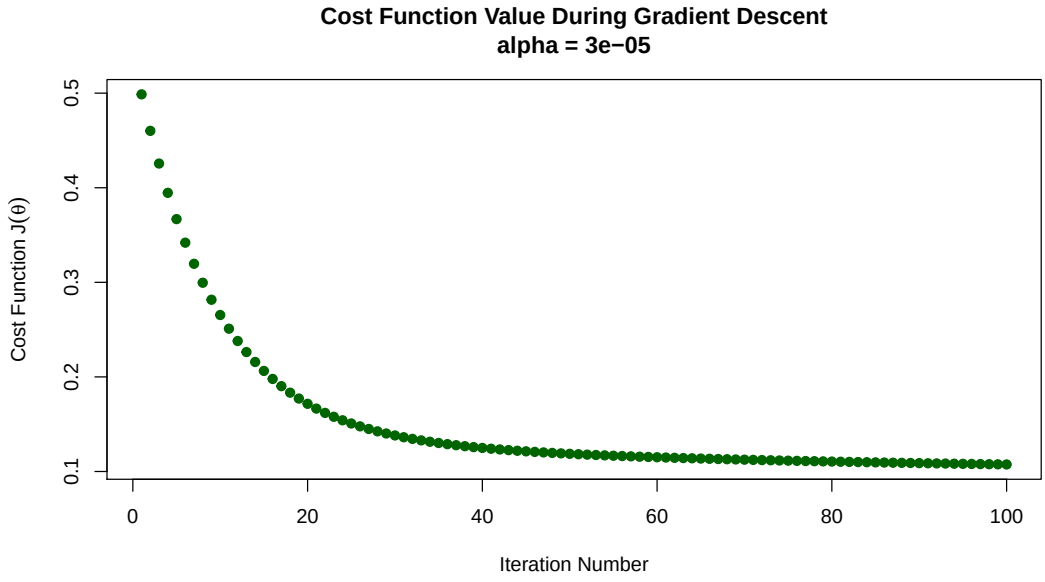
The `R` code for the calculation of this question can be found in the Problem 1.7 part of the file `Prob1-Part2-MultiLR.R`.

(1.8)


Cost Function Value During Gradient Descent
alpha = 3
Cost Function J(θ)
0.0e+00
5.0e+297
1.0e+298
1.5e+298
0
10
20
30
40
Iteration Number


Cost Function Value During Gradient Descent
alpha = 0.3
Cost Function J(θ)
0.0e+00
4.0e+299
8.0e+299
1.2e+300
0
10
20
30
40
50
Iteration Number

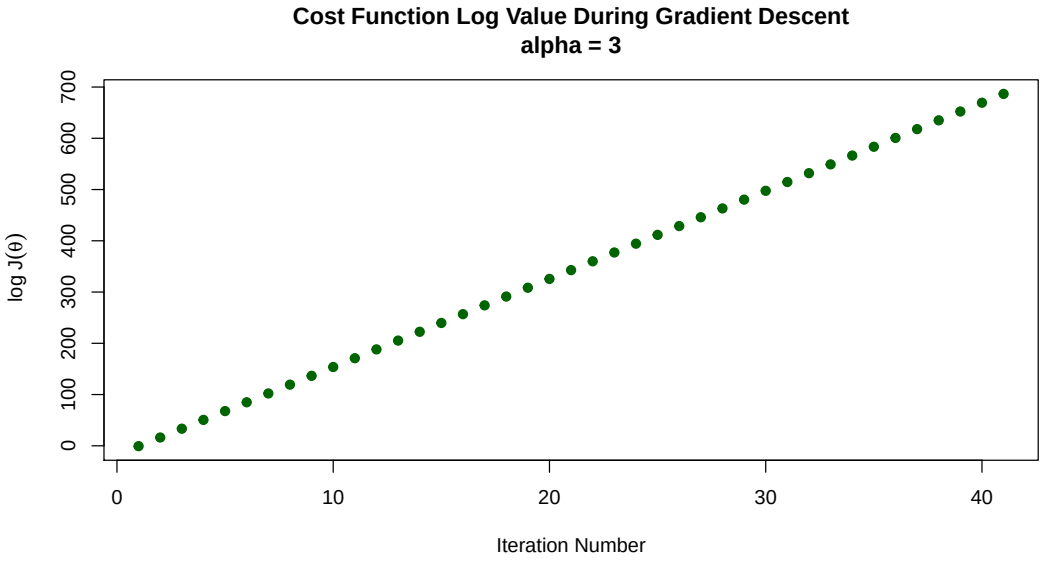


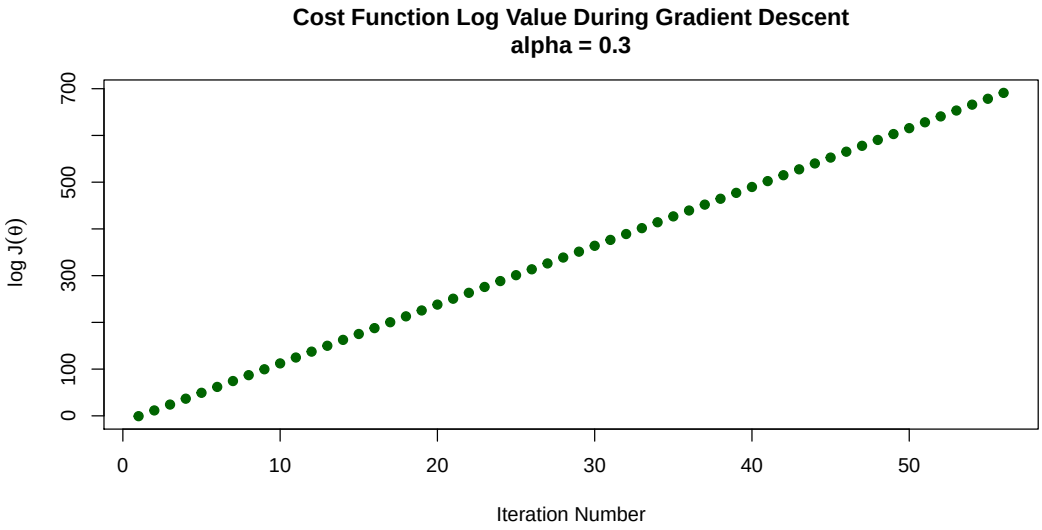


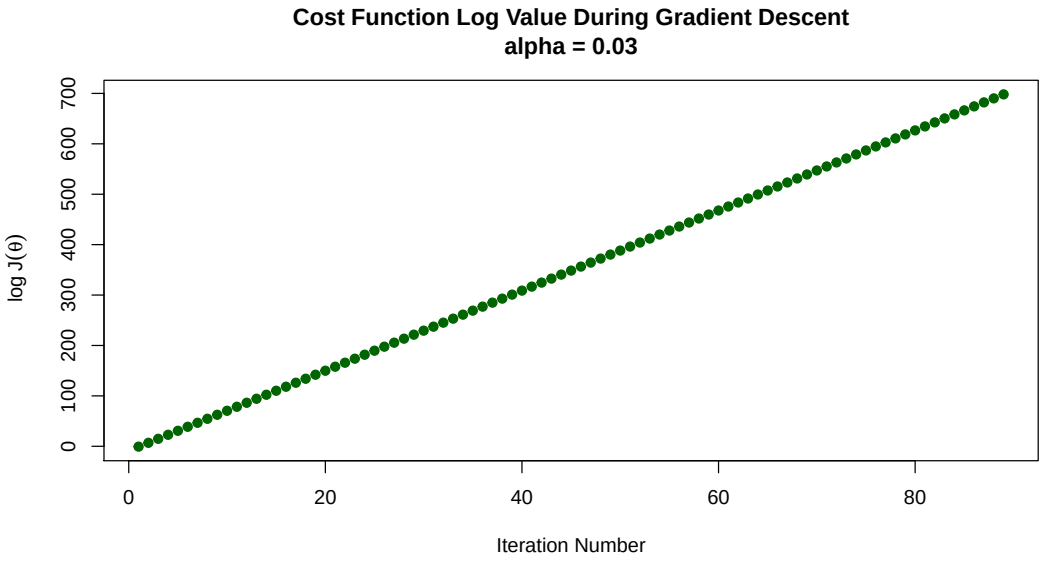
Analysis

As we can see from the plots, when we use $\alpha = 3, 0.3$, or 0.03, the method does not converge and the cost function blows up. When $\alpha = 0.00003$, from the fourth graph, we can see that the method converges. Therefore, the plot with $\alpha = 0.00003$ looks much better than others.

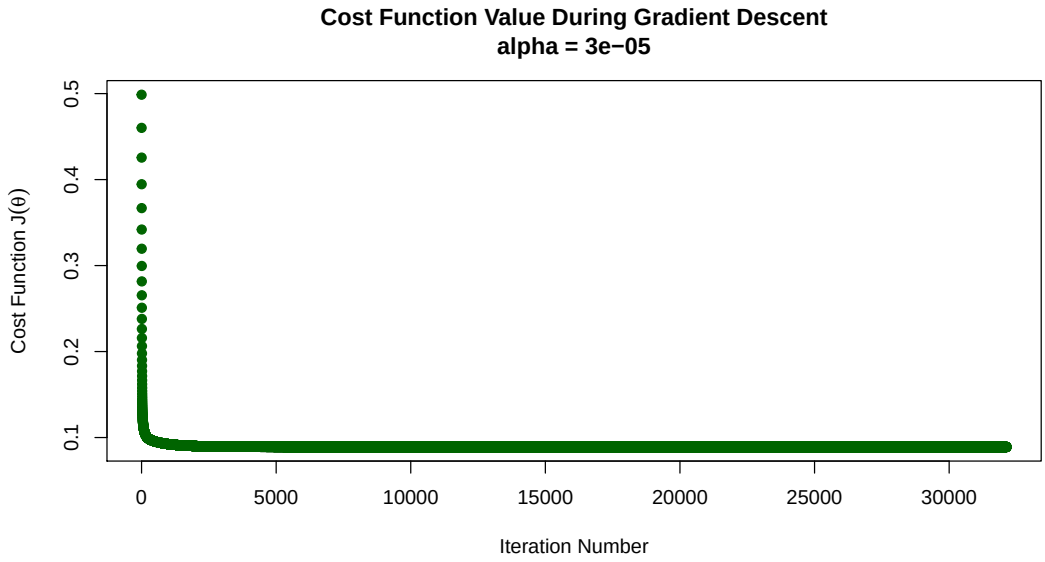
Note that in the first 3 graphs, it seems that all cost function values, except the last one, are the same. This is not really the case. The cost function in fact grows at exponential rate, therefore masks the true growth rate in the plot. To show more clearly how fast the growth is, we provide below the log plots for the first 3 cases.


Cost Function Log Value During Gradient Descent
alpha = 3
log J(θ)
0
100
200
300
400
500
600
700
0
10
20
30
40
Iteration Number


Cost Function Log Value During Gradient Descent
alpha = 0.3
log J(θ)
0
100
300
500
700
0
10
20
30
40
50
Iteration Number


Cost Function Log Value During Gradient Descent
alpha = 0.03
log J(θ)
0
100
200
300
400
500
600
700
0
20
40
60
80
Iteration Number

Also, here we give a plot for $\alpha = 0.00003$ in the long run, until actual convergence.



(1.9) By using the closed form formula,

$$\theta = (X^T X)^{-1} X^T y = \begin{pmatrix} -7.372575 \times 10^{-16} \\ -0.1078273 \\ 0.2667467 \\ -0.08359632 \\ -0.7045565 \\ 0.02848143 \\ 0.3543429 \\ 0.1471853 \end{pmatrix}$$

where $X \in \mathbb{R}^{m \times n}, y \in \mathbb{R}^m$ with

$$X = \begin{pmatrix} x^{(1)} \\ x^{(2)} \\ \cdots \\ x^{(m)} \end{pmatrix}$$

being the matrix of sample values of the input variables, and y being the vector of sample values of the response variable.

The result in `R` is given below, with the labels on the coefficients

```
Intercept    -7.372575e-17
cylinders    -1.078273e-01
displacement  2.667467e-01
horsepower   -8.359623e-02
weight       -7.045565e-01
acceleration  2.848143e-02
year          3.543429e-01
origin        1.471853e-01
```

The actual calculation using the `R` language is performed in the "Problem 1.9" part of the file `Prob1-Part2-MultiLR.R`.

Problem 2 [35 points]

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

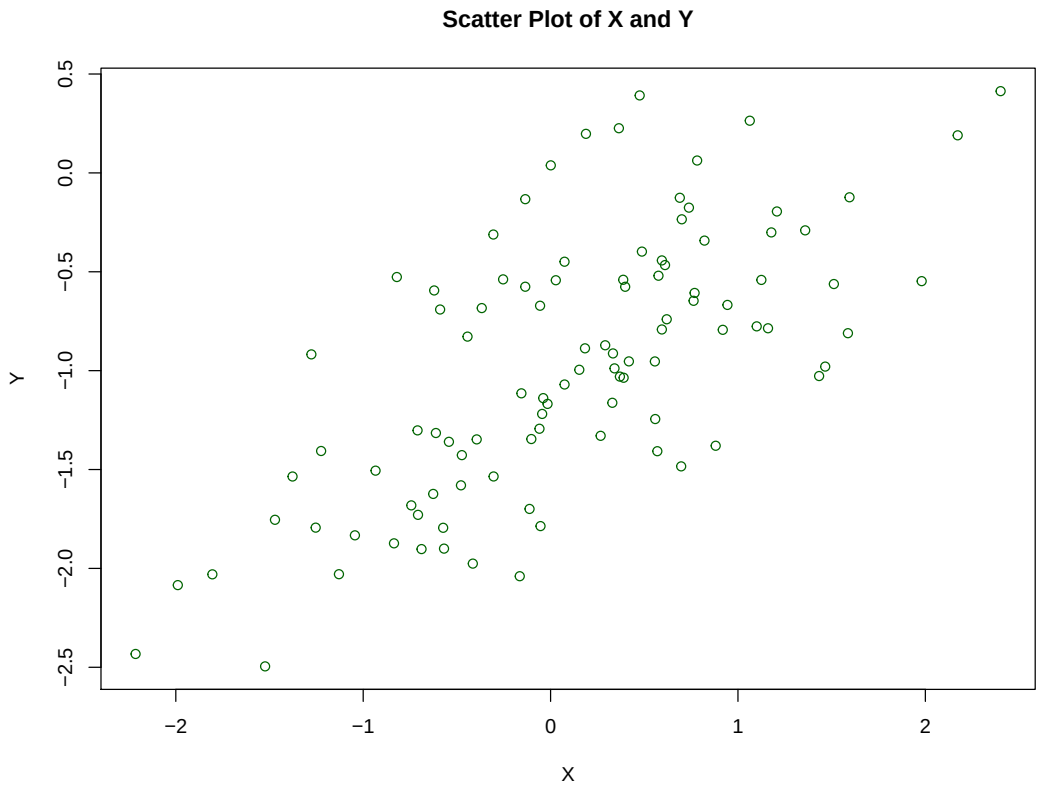
Exercise 13

Original Noise

(a-c) The random data generation is performed in the `R` file `Prob2-Ex13-Part1.R`.
The length of the vector `y` is 100. In the linear model, the coefficients are

$$\beta_0 = -1, \qquad \beta_1 = 0.5$$

(d)



The scatter plot shows that the `x` and `y` roughly has a linear correlation, but the relationship is very noisy as well.

(e) The linear fitting is performed in the `R` file `Prob2-Ex13-Part1.R`. The coefficients in the results are

$$\hat{\beta}_0 = -1.0188463, \qquad \hat{\beta}_1 = 0.4994698$$

That is, our linear model can be described as

$$\texttt{y} \sim \hat{\beta}_0 + \hat{\beta}_1 \texttt{x} = -1.0188463 + 0.4994698\texttt{x}$$

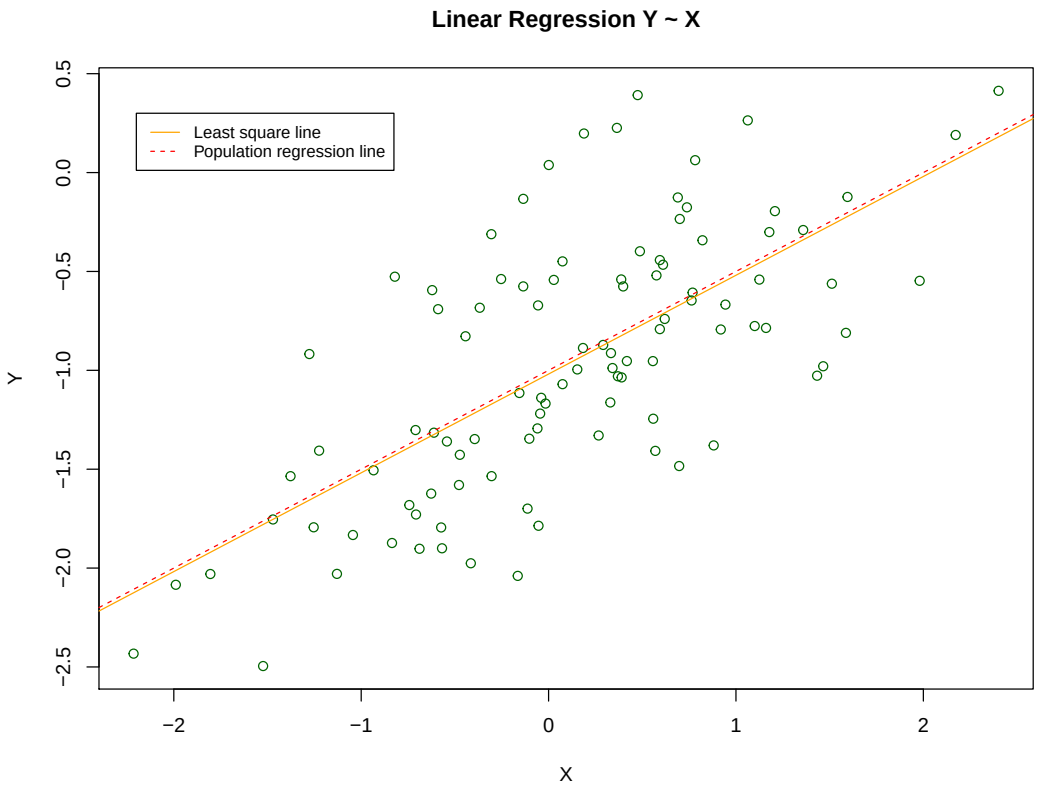
This model is very close to the original population model without noise $Y = -1 + 0.5X$.

Comparing with the original β_0 and β_1, we calculate in `R`

$$|\hat{\beta}_0 - \beta_0| = 0.01884631 \quad |\hat{\beta}_1 - \beta_1| = 0.0005301931$$

We can see that our estimated coefficients $\hat{\beta}_0$, $\hat{\beta}_1$ are close to the population values β_0 and β_1. The difference is caused by the random noise ϵ. Also we can see that $\hat{\beta}_1$ is very close to β_1. This is expected, as our random noise ϵ is symmetrical, i.e. with mean 0.

(f)



In the plot, the orange line represents the least square line. The red dotted line represents the population regression line. The two lines are very close. Thus we chose dotted line for the population line to differentiate it from the other line.

(g) Using commands in the `R` file `Prob2-Ex13-Part1.R`, we have the following result for the polynomial fitting

```
(Intercept)           x      I(x^2)
 -0.9716425   0.5085804  -0.0594606
```

or

$$\mathrm{y} \sim \hat{\beta}_0 + \hat{\beta}_1 \mathrm{x} + \hat{\beta}_2 \mathrm{x}^2 = -0.9716425 + 0.5085804\mathrm{x} - 0.0594606\mathrm{x}^2$$

In order to compare the result with the linear regression, we calculate the residual sum of squares (RSS) for both cases.

$$\text{RSS}_{\text{linear}} = 22.7089, \qquad \text{RSS}_{\text{quadratic}} = 22.25728$$

From the RSS point of view, the quadratic fitting performs better than the linear fitting, but the differences is not significant.

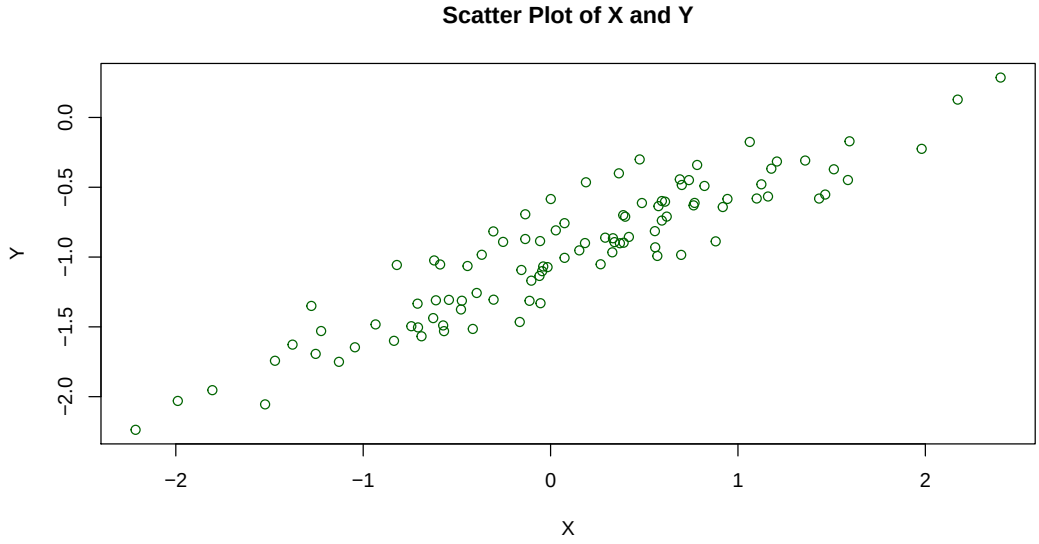
Less Noise

(h) We use the noise $\epsilon \sim N(0, 0.2^2)$.

(a-c) The length of the vector `y` is 100. In the linear model, the coefficients are still

$$\beta_0 = -1, \qquad \beta_1 = 0.5$$

(d)



The scatter plot shows that the `x` and `y` roughly has a linear correlation. The relationship is noisy, but the noise is smaller in scale compare to the original data set.

(e) The coefficients in the results are

$$\hat{\beta}_0 = -1.0075385, \qquad \hat{\beta}_1 = 0.4997879$$

That is, our linear model can be described as

$$\texttt{y} \sim \hat{\beta}_0 + \hat{\beta}_1 \texttt{x} = -1.0075385 + 0.4997879\texttt{x}$$

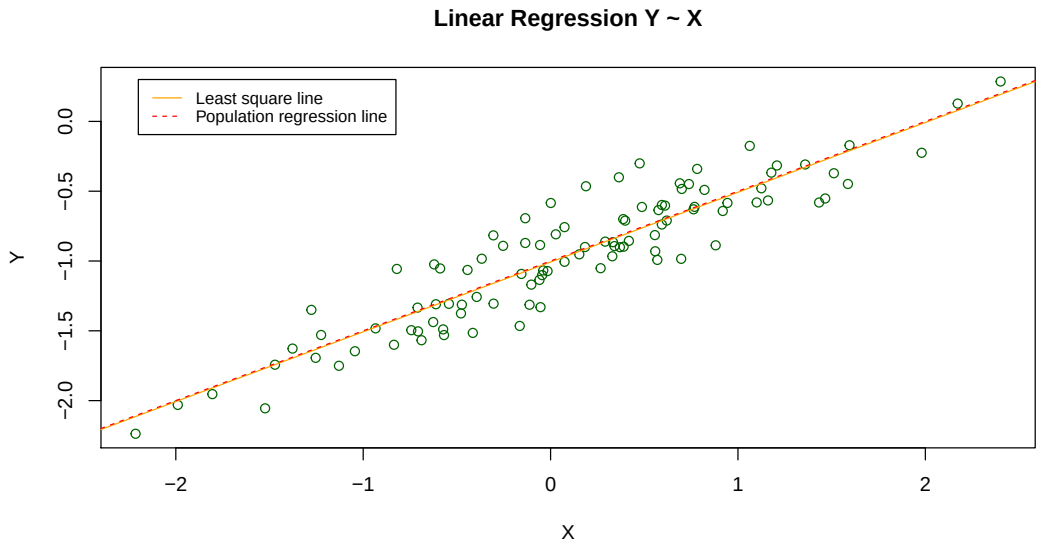
This model is very close to the original population model without noise $Y = -1 + 0.5X$.

Comparing with the original β_0 and β_1, we calculate in `R`

$$|\hat{\beta}_0 - \beta_0| = 0.007538523 \quad |\hat{\beta}_1 - \beta_1| = 0.0002120772$$

We can see that our estimated coefficients $\hat{\beta}_0$, $\hat{\beta}_1$ are close to the population values β_0 and β_1. The difference is caused by the random noise ϵ. Also we can see that $\hat{\beta}_1$ is very close to β_1. This is expected, as our random noise ϵ is symmetrical, i.e. with mean 0.

(f)



In the plot, the orange line represents the least square line. The red dotted line represents the population regression line. The two lines are very close. Thus we chose dotted line for the population line to differentiate it from the other line.

(g) Using commands in the `R` file `Prob2-Ex13-Part1.R`, we have the following result for the polynomial fitting

```
(Intercept)           x      I(x^2)
-0.98865700  0.50343217 -0.02378424
```

or

$$\texttt{y} \sim \hat{\beta}_0 + \hat{\beta}_1 \texttt{x} + \hat{\beta}_2 \texttt{x}^2 = -0.98865700 + 0.50343217\texttt{x} - 0.02378424\texttt{x}^2$$

In order to compare the result with the linear regression, we calculate the residual sum of squares (RSS) for both cases.

$$\text{RSS}_{\text{linear}} = 3.633424, \qquad \text{RSS}_{\text{quadratic}} = 3.561164$$

From the RSS point of view, the quadratic fitting performs better than the linear fitting, but the difference is not significant.

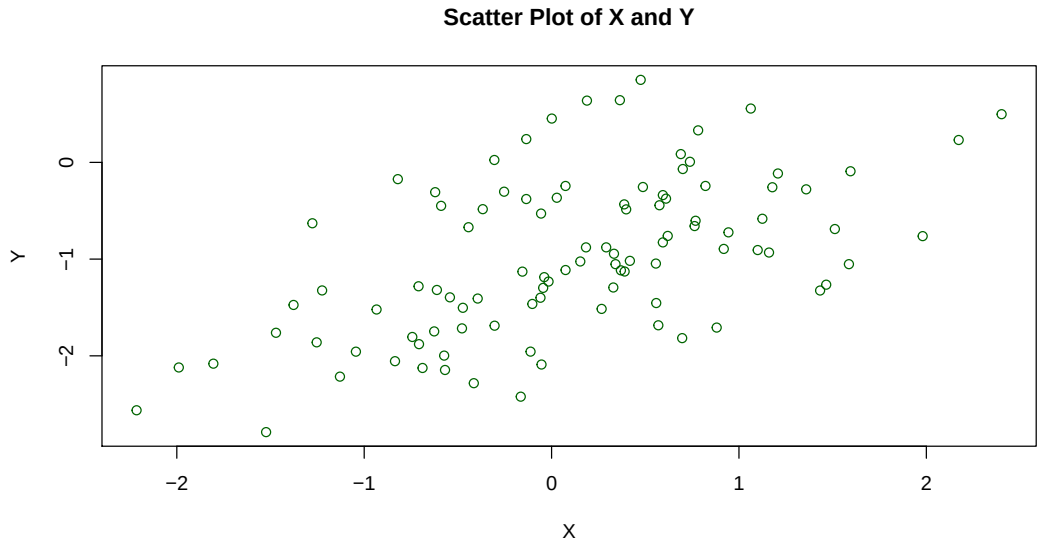
More Noise

(i) We use the noise $\epsilon \sim N(0, 0.7^2)$.

(a-c) The length of the vector `y` is 100. In the linear model, the coefficients are still

$$\beta_0 = -1, \qquad \beta_1 = 0.5$$

(d)



The scatter plot shows that the `x` and `y` roughly has a linear correlation. The relationship is very noisy, and the noise is larger in scale compare to the previous 2 data sets.

(e) The coefficients in the results are

$$\hat{\beta}_0 = -1.0263848, \qquad \hat{\beta}_1 = 0.4992577$$

That is, our linear model can be described as

$$\texttt{y} \sim \hat{\beta}_0 + \hat{\beta}_1 \texttt{x} = -1.0263848 + 0.4992577\texttt{x}$$

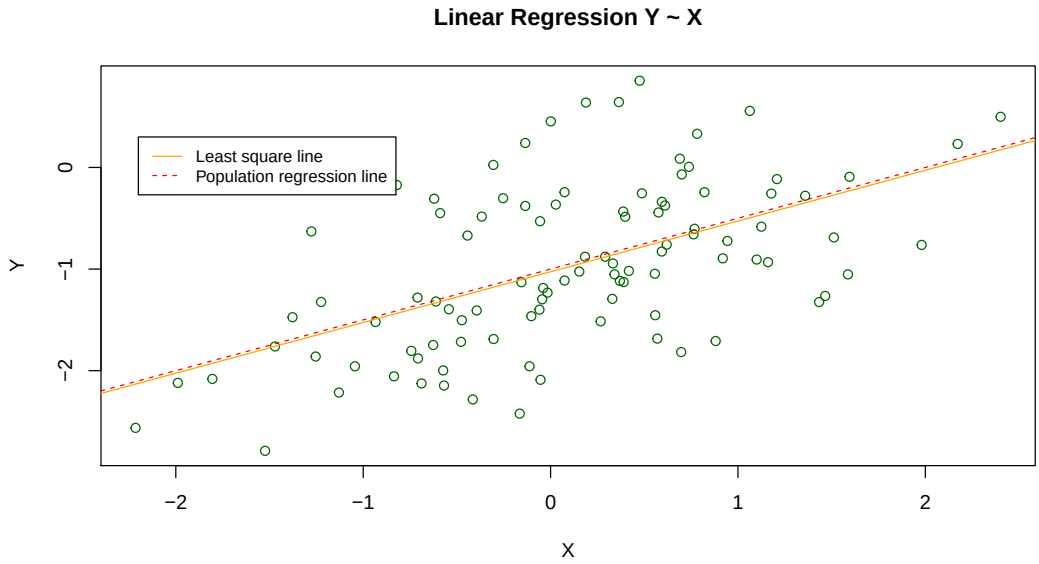
This model is very close to the original population model without noise $Y = -1 + 0.5X$.

Comparing with the original β_0 and β_1, we calculate in `R`

$$|\hat{\beta}_0 - \beta_0| = 0.02638483 \quad |\hat{\beta}_1 - \beta_1| = 0.0007422703$$

We can see that our estimated coefficients $\hat{\beta}_0$, $\hat{\beta}_1$ are close to the population values β_0 and β_1. The difference is caused by the random noise ϵ. Also we can see that $\hat{\beta}_1$ is very close to β_1. This is expected, as our random noise ϵ is symmetrical, i.e. with mean 0.

(f)



In the plot, the orange line represents the least square line. The red dotted line represents the population regression line. The two lines are very close. Thus we chose dotted line for the population line to differentiate it from the other line.

(g) Using commands in the `R` file `Prob2-Ex13-Part1.R`, we have the following result for the polynomial fitting

```
(Intercept)           x      I(x^2)
-0.96029949  0.51201261 -0.08324484
```

or

$$\mathtt{y} \sim \hat{\beta}_0 + \hat{\beta}_1 \mathtt{x} + \hat{\beta}_2 \mathtt{x}^2 = -0.96029949 + 0.51201261\mathtt{x} - 0.08324484\mathtt{x}^2$$

In order to compare the result with the linear regression, we calculate the residual sum of squares (RSS) for both cases.

$$\mathrm{RSS}_{\mathrm{linear}} = 44.50945, \qquad \mathrm{RSS}_{\mathrm{quadratic}} = 43.62426$$

From the RSS point of view, the quadratic fitting performs better than the linear fitting, but the difference is not significant. In this case, the improvement on RSS is better than the previous cases, because our data set is more noisy. This improvement on RSS does not really mean the quadratic model performs better. In fact, in this case, the quadratic fitting captures more noise than the underlying pattern.

Confidence Intervals

(j) Here we give the 95% confidence intervals for the coefficients β_0 and β_1.

- **Less Noise**

$$I_{\beta_0} = [-1.0460321, -0.9690449], \quad I_{\beta_1} = [0.4570318, 0.5425441]$$

Interval lengths:

$$|I_{\beta_0}| = |-1.0460321 - (-0.9690449)| = 0.0769872,$$
$$|I_{\beta_1}| = |0.4570318 - 0.5425441| = 0.0855123$$

- **Original Noise**

$$I_{\beta_0} = [-1.1150804, -0.9226122], \quad I_{\beta_1} = [0.3925794, 0.6063602]$$

Interval lengths:

$$|I_{\beta_0}| = |-1.1150804 - (-0.9226122)| = 0.1924682,$$
$$|I_{\beta_1}| = |0.3925794 - 0.6063602| = 0.2137808$$

- **More Noise**

$$I_{\beta_0} = [-1.1611125, -0.8916571], \quad I_{\beta_1} = [0.3496112, 0.6489043]$$

Interval lengths:

$$|I_{\beta_0}| = |-1.1611125 - (-0.8916571)| = 0.2694554,$$
$$|I_{\beta_1}| = |0.3496112 - 0.6489043| = 0.2992931$$

Here, we compare the results by comparing the lengths of the confidence intervals. As we can see, the interval lengths increase as we go from less noise to more noise. This is consistent with the increase in the noise. When noise is heavier, our estimate would become less accurate, and thus result in larger confidence intervals.

Exercise 14

(a) The linear model is

$$Y = 2 + 2X_1 + 0.3X_2 + \epsilon$$

where $\epsilon \sim N(0, 0.1^2)$. The regression coefficients are corresponding to $X_0 = 1$, X_1 and X_2

$$\beta_0 = 2, \quad \beta_1 = 2, \quad \beta_2 = 0.3$$

(b) The correlation is

$$\text{cor}(\texttt{x1}, \texttt{x2}) = 0.8351212$$

(c) Using linear regression, we have the least square regression coefficients as

$$\hat{\beta}_0 = 2.130500, \quad \hat{\beta}_1 = 1.439555, \quad \hat{\beta}_2 = 1.009674$$

The least squares regression model can be described as

$$\texttt{y} \sim \hat{\beta}_0 + \hat{\beta}_1\texttt{x1} + \hat{\beta}_2\texttt{x2} = 2.130500 + 1.439555\texttt{x1} + 1.009674\texttt{x2}$$

The differences between all estimated coefficients and true coefficients are calculated below.

$$|\beta_0 - \hat{\beta}_0| = 0.1304996, \quad |\beta_1 - \hat{\beta}_1| = 0.5604446, \quad |\beta_2 - \hat{\beta}_2| = 0.7096742$$

From these numbers, we can see that β_0 is closest to the true value, while β_1 is quite faraway from the true value and β_2 is very faraway from the true value.

For the hypothesis tests:

- $H_0 : \beta_1 = 0$
  The t-value is 1.996 and this gives a p-value of 0.0487. If we use a significance level of 0.05, we would **reject** H_0 and accept the alternative hypothesis $H_1 : \beta_1 \neq 0$, i.e. `y` is linearly correlated with `x1` .

  However, the p-value is close to the significance level 0.05. This means that using any significance level lower than 0.0487, e.g. 0.01, would result in acceptance of H_0 and the conclusion that `y` is not linearly correlated with `x1` .

- $H_0 : \beta_2 = 0$
  The t-value is 0.891 and this gives a p-value of 0.3754. If we use a significance level of 0.05, we would **not reject** H_0, i.e. `y` is not linearly correlated with `x2` .

(d) Using linear regression, we have the least square regression coefficients as

$$\hat{\beta}_0 = 2.112394, \quad \hat{\beta}_1 = 1.975929$$

The least squares regression model can be described as

$$\texttt{y} \sim \hat{\beta}_0 + \hat{\beta}_1 \texttt{x1} = 2.112394 + 1.975929 \texttt{x1}$$

The differences between all estimated coefficients and true coefficients are calculated below.

$$|\beta_0 - \hat{\beta}_0| = 0.1123936, \quad |\beta_1 - \hat{\beta}_1| = 0.0240710$$

From these numbers, we can see that β_0 and β_1 are both close to the true value.

For the hypothesis test

$$H_0 : \beta_1 = 0$$

The t-value is 4.986 and this gives a p-value of 2.66×10^{-6}. If we use a significance level of 0.05, or any common levels, we would **reject** H_0 and accept the alternative hypothesis $H_1 : \beta_1 \neq 0$, i.e. `y` is linearly correlated with `x1` .

(e) Using linear regression, we have the least square regression coefficients as

$$\hat{\beta}_0 = 2.389949, \quad \hat{\beta}_1 = 2.899585$$

The least squares regression model can be described as

$$\texttt{y} \sim \hat{\beta}_0 + \hat{\beta}_1 \texttt{x2} = 2.389949 + 2.899585 \texttt{x2}$$

The differences between all estimated coefficients and true coefficients are calculated below.

$$|\beta_0 - \hat{\beta}_0| = 0.3899491, \quad |\beta_1 - \hat{\beta}_1| = 2.599585$$

From these numbers, we can see that β_0 is not faraway from the true value but β_1 is very different from the true value.

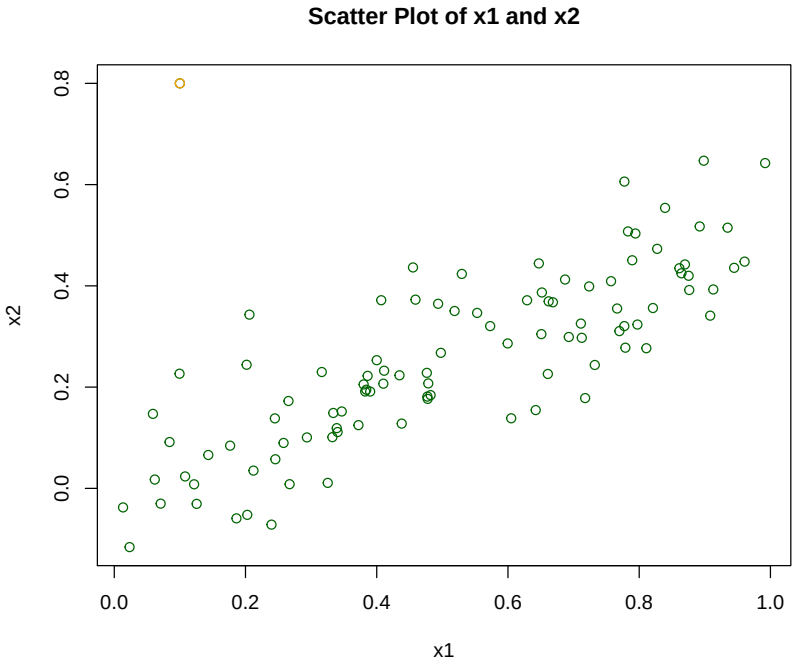
For the hypothesis test

$$H_0 : \beta_1 = 0$$

The t-value is 4.58 and this gives a p-value of 1.37×10^{-5}. If we use a significance level of 0.05, or any common levels, we would **reject** H_0 and accept the alternative hypothesis $H_1 : \beta_1 \neq 0$, i.e. `y` is linearly correlated with `x2` .

(f) The results obtained in (c)-(e) do not contradict each other. The reason that we conclude that `x2` is not significantly correlated with `y` in the multivariate linear regression tests and that `x2` is correlated with `y` with statistical significance in the univariate linear regression tests is that, `x1` and `x2` , or equivalently X_1 and X_2, are very correlated.

(g)



In the plot, the orange dot represents the added data point.

(c) Using linear regression, we have the least square regression coefficients as

$$\hat{\beta}_0 = 2.2266917, \quad \hat{\beta}_1 = 0.5394397, \quad \hat{\beta}_2 = 2.5145694$$

The least squares regression model can be described as

$$\texttt{y} \sim \hat{\beta}_0 + \hat{\beta}_1 \texttt{x1} + \hat{\beta}_2 \texttt{x2} = 2.2266917 + 0.5394397\texttt{x1} + 2.5145694\texttt{x2}$$

The differences between all estimated coefficients and true coefficients are calculated below.

$$|\beta_0 - \hat{\beta}_0| = 0.2266917, \quad |\beta_1 - \hat{\beta}_1| = 1.4605603, \quad |\beta_2 - \hat{\beta}_2| = 2.2145694$$

From these numbers, we can see that β_0 is closest to the true value, while β_1 and β_2 are both very faraway from the true value.

For the hypothesis tests:

- $H_0 : \beta_1 = 0$
  The p-value is 0.36458. If we use a significance level of 0.05, we would **not reject** H_0, i.e. `y` is not linearly correlated with `x1` .

- $H_0 : \beta_2 = 0$
  The p-value is 0.00614. If we use a significance level of 0.05 or 0.01, we would **reject** H_0, i.e. `y` is linearly correlated with `x2` .

(d) Using linear regression, we have the least square regression coefficients as

$$\hat{\beta}_0 = 2.256927, \quad \hat{\beta}_1 = 1.765695$$

The least squares regression model can be described as

$$\texttt{y} \sim \hat{\beta}_0 + \hat{\beta}_1 \texttt{x1} = 2.256927 + 1.765695 \texttt{x1}$$

The differences between all estimated coefficients and true coefficients are calculated below.

$$|\beta_0 - \hat{\beta}_0| = 0.2569274, \quad |\beta_1 - \hat{\beta}_1| = 0.2343045$$

From these numbers, we can see that β_0 and β_1 are both close to the true value.

For the hypothesis test

$$H_0 : \beta_1 = 0$$

The p-value is 4.29×10^{-5}. If we use a significance level of 0.05, or any common levels, we would **reject** H_0 and accept the alternative hypothesis $H_1 : \beta_1 \neq 0$, i.e. y is linearly correlated with x1 .

(e) Using linear regression, we have the least square regression coefficients as

$$\hat{\beta}_0 = 2.345107, \quad \hat{\beta}_1 = 3.119050$$

The least squares regression model can be described as

$$\texttt{y} \sim \hat{\beta}_0 + \hat{\beta}_1 \texttt{x2} = 2.345107 + 3.119050 \texttt{x2}$$

The differences between all estimated coefficients and true coefficients are calculated below.

$$|\beta_0 - \hat{\beta}_0| = 0.3451069, \quad |\beta_1 - \hat{\beta}_1| = 2.8190497$$

From these numbers, we can see that β_0 is not faraway from the true value but β_1 is very different from the true value.

For the hypothesis test

$$H_0 : \beta_1 = 0$$

The p-value is 1.25×10^{-6}. If we use a significance level of 0.05, or any common levels, we would **reject** H_0 and accept the alternative hypothesis $H_1 : \beta_1 \neq 0$, i.e. y is linearly correlated with x2 .

Summary

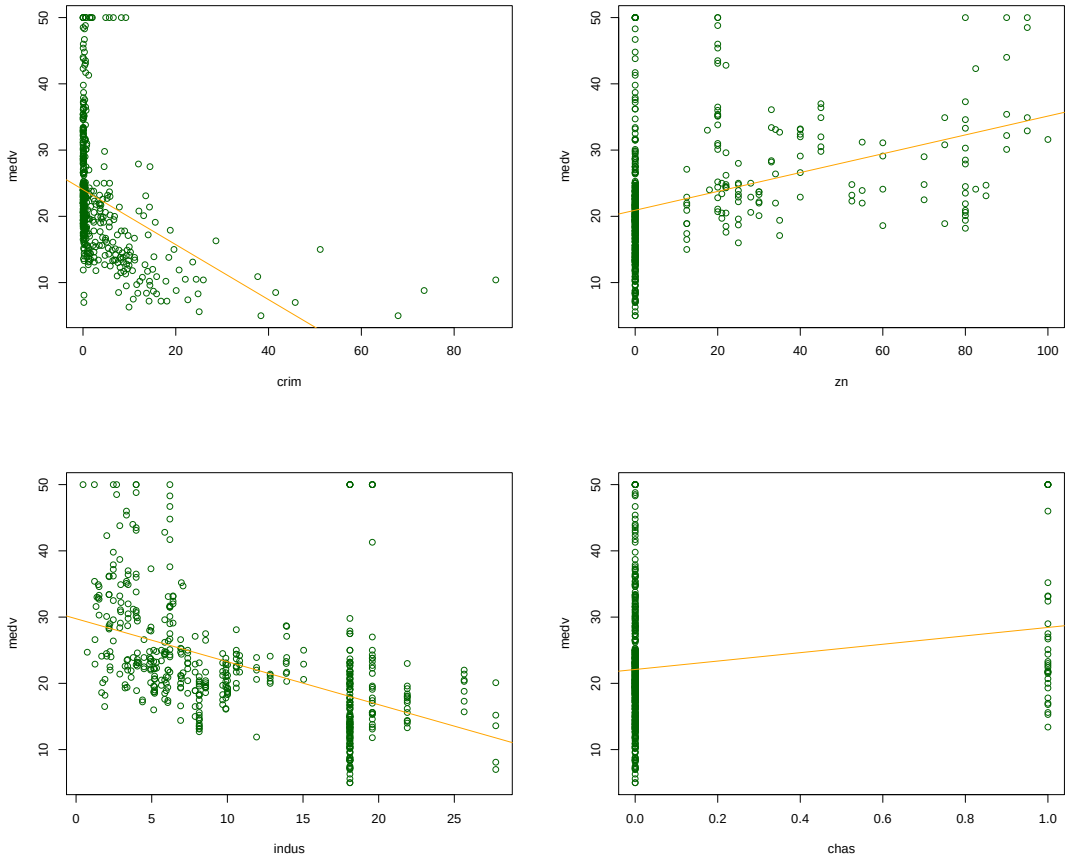
Adding the data point completely changed $\hat{\beta}_1$ and $\hat{\beta}_2$ and resulted in different hypothesis tests on them, i.e. accepting that x2 is linearly correlated with y but not x1. The change is not large however in the (d) and (e) models. The estimates and hypothesis tests gave similar results in both cases.

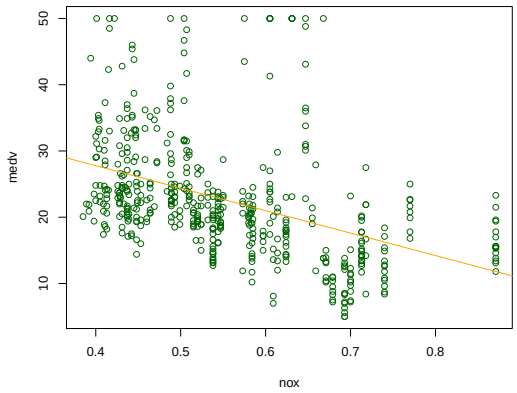
Exercise 15

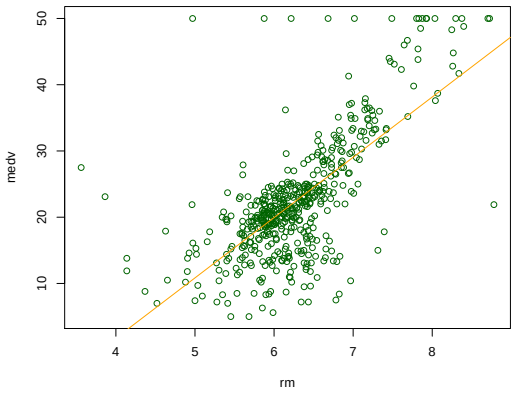
(a) Using R, we calculated all the least square regression coefficients.

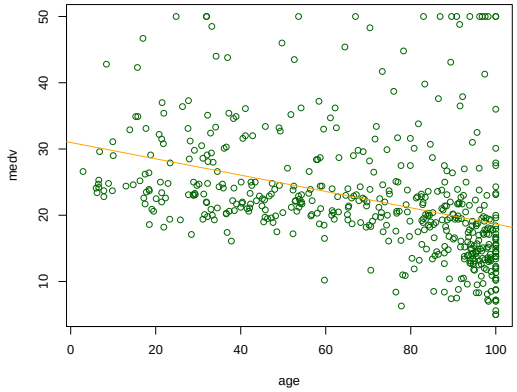
	beta_0	beta_1
crim	24.03311	-0.41519028
zn	20.91758	0.14213999
indus	29.75490	-0.64849005
chas	22.09384	6.34615711
nox	41.34587	-33.91605501
rm	-34.67062	9.10210898
age	30.97868	-0.12316272
dis	18.39009	1.09161302
rad	26.38213	-0.40309540
tax	32.97065	-0.02556810
ptratio	62.34463	-2.15717530
black	10.55103	0.03359306
lstat	34.55384	-0.95004935

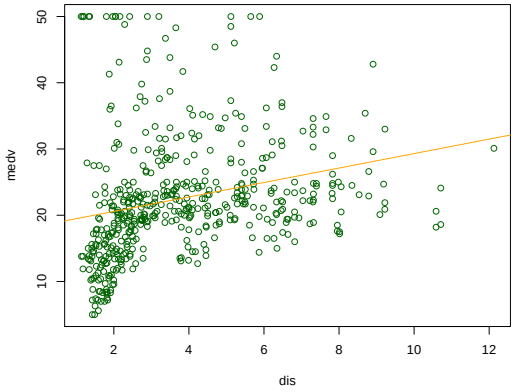
The following plots are scatter plots for each pair of the predictor and the response variable. The orange line represents the least square regression line.

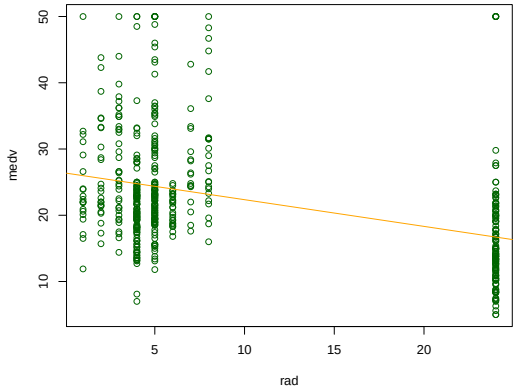


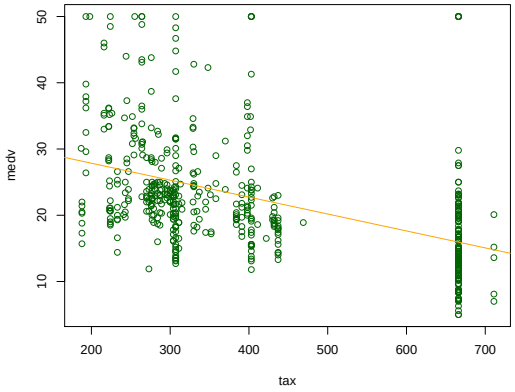

medv
nox

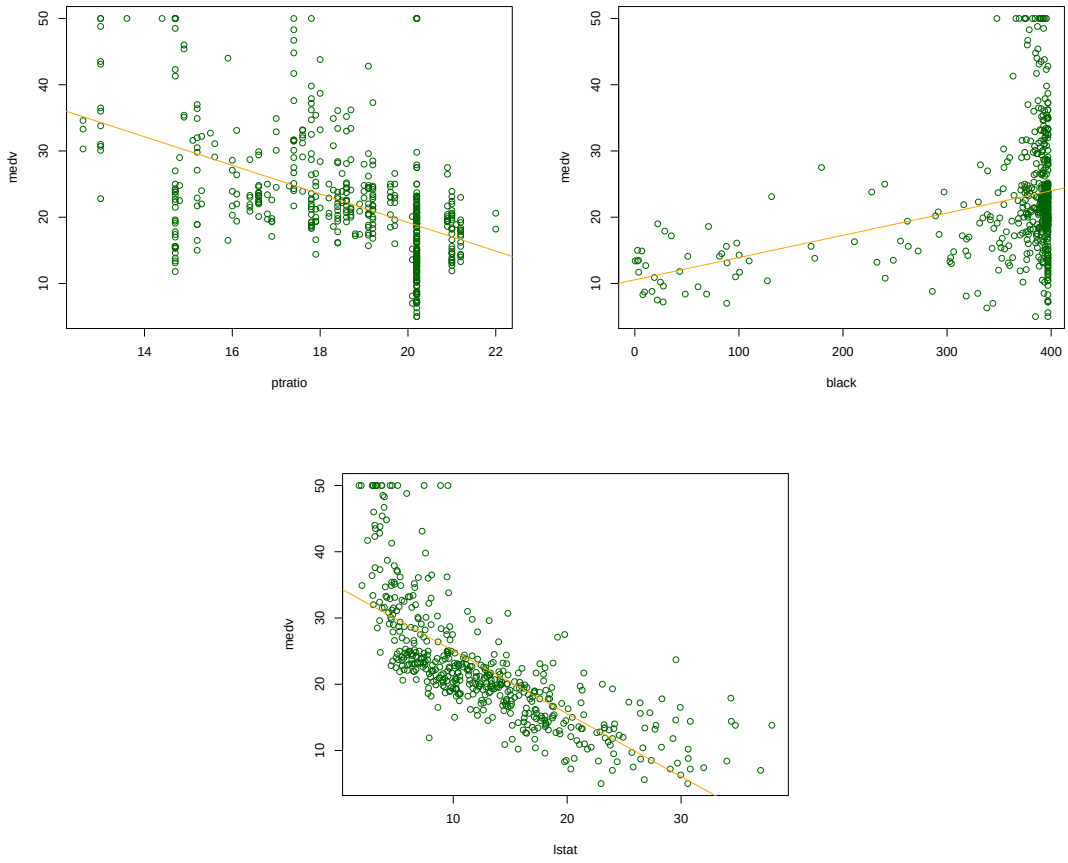

medv
rm


medv
age


medv
dis


medv
rad


medv
tax



From the graph, we can see that `nox`, `rm`, `age`, `dis`, `ptratio` and `lstat` seem to have statistically significant association with the response.

We also give the p-values below.

	beta_0	beta_1
crim	1.341723e-227	1.173987e-19
zn	9.489803e-195	5.713584e-17
indus	6.704987e-173	4.900260e-31
chas	7.002789e-208	7.390623e-05
nox	9.866245e-80	7.065042e-24
rm	6.950229e-34	2.487229e-74
age	6.814198e-119	1.569982e-18
dis	4.008955e-78	1.206612e-08
rad	3.282092e-186	5.465933e-19
tax	5.519383e-136	5.637734e-29
ptratio	9.077444e-69	1.609509e-34
black	3.491585e-11	1.318113e-14
lstat	3.743081e-236	5.081103e-88

From the p-values, we conclude that they all have statistically significant association with the response value.

Here, the p-values are very small. This might be the effect of large number of data points, which causes the t-distribution to have small tails. We should consider using lower significance levels or try other statistical methods to determine association.

(b)

Using `R`, we have the following result for the coefficients

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12
crim        -1.080e-01  3.286e-02  -3.287 0.001087
zn           4.642e-02  1.373e-02   3.382 0.000778
indus        2.056e-02  6.150e-02   0.334 0.738288
chas         2.687e+00  8.616e-01   3.118 0.001925
nox         -1.777e+01  3.820e+00  -4.651 4.25e-06
rm           3.810e+00  4.179e-01   9.116  < 2e-16
age          6.922e-04  1.321e-02   0.052 0.958229
dis         -1.476e+00  1.995e-01  -7.398 6.01e-13
rad          3.060e-01  6.635e-02   4.613 5.07e-06
tax         -1.233e-02  3.760e-03  -3.280 0.001112
ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12
black        9.312e-03  2.686e-03   3.467 0.000573
lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16
```

That is, the coefficients are

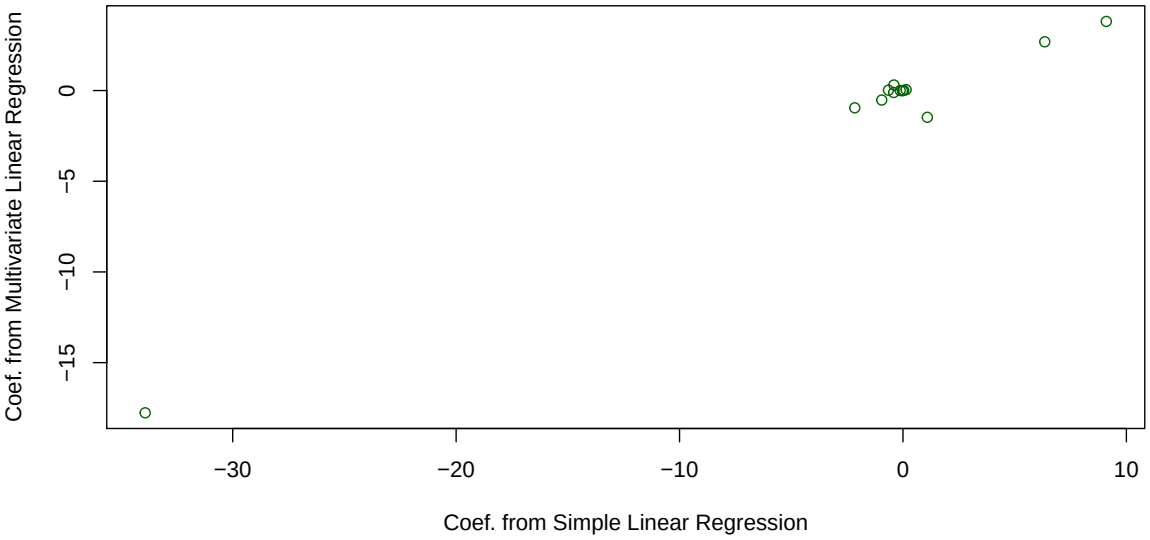
$$\beta = \begin{pmatrix} 36.45949 \\ -0.1080114 \\ 0.04642046 \\ 0.02055863 \\ 2.686734 \\ -17.76661 \\ 3.809865 \\ 0.0006922246 \\ -1.475567 \\ 0.3060495 \\ -0.01233459 \\ -0.9527472 \\ 0.009311683 \\ -0.5247584 \end{pmatrix}$$

We use significance level of 0.05. Then the following predictors have p-values less than the significance level 0.05:

`Intercept`, `crim`, `zn`, `chas`, `nox`, `rm`, `dis`, `rad`, `tax`, `ptratio`, `black`, `lstat`

That is, all the predictors except `indus` and `age`. For these predictors, we reject the null hypothesis H_0.

(c)



Compared to (a), we have more predictors statistically associated to the response in (b).

(d) Coefficients:

	beta_0	beta_1	beta_2	beta_3
crim	25.190479	-1.13640072	2.378483e-02	-1.488721e-04
zn	20.448597	0.64336518	-1.676457e-02	1.257022e-04
indus	37.080160	-2.80699408	1.404617e-01	-2.398890e-03
chas	22.093843	6.34615711	NA	NA
nox	-22.486390	315.09596351	-6.158267e+02	3.501860e+02
rm	241.310814	-109.39060642	1.649102e+01	-7.403937e-01
age	28.931102	-0.12241882	2.354563e-03	-2.317915e-05
dis	7.037894	8.59284424	-1.249528e+00	5.601936e-02
rad	30.251303	-3.79945390	6.163466e-01	-2.008638e-02
tax	52.216023	-0.16346964	3.029307e-04	-2.078684e-07
ptratio	312.286417	-48.69113602	2.839951e+00	-5.686478e-02
black	12.598120	-0.01703275	2.036076e-04	-2.224273e-07
lstat	48.649625	-3.86559278	1.487385e-01	-2.003868e-03

p-values:

	beta_0	beta_1	beta_2	beta_3
crim	3.638793e-224	2.235435e-14	5.184482e-04	2.541096e-02
zn	1.301845e-185	1.055518e-08	1.941865e-05	7.981118e-05
indus	4.912614e-77	5.713973e-08	7.807276e-04	1.802575e-02
chas	7.002789e-208	7.390623e-05	7.002789e-208	7.390623e-05
nox	5.596479e-01	1.069282e-01	5.522193e-02	4.100152e-02
rm	4.853140e-07	2.505427e-06	8.952478e-06	1.461903e-04

```
age      2.160623e-20 5.435765e-01  5.493772e-01 3.096903e-01
dis      1.598664e-02 3.767574e-05  2.569178e-03 2.146307e-02
rad      1.871399e-28 3.814616e-03  9.907537e-04 4.819196e-04
tax      2.062352e-04 1.496461e-01  2.920043e-01 3.530609e-01
ptratio  4.108411e-02 7.071842e-02  7.001702e-02 5.900888e-02
black    7.700903e-07 7.819282e-01  5.323182e-01 6.408515e-01
lstat   6.294703e-132 2.329688e-28  9.177622e-12 7.428441e-07
```

p-values from F-tests:

```
crim     1.448648e-26
zn       1.655884e-19
indus    4.415394e-35
chas     7.390623e-05
nox      2.561169e-23
rm       2.141469e-89
age      1.962401e-18
dis      4.735610e-12
rad      4.866503e-21
tax      9.824004e-28
ptratio  1.354349e-33
black    4.462765e-13
lstat   1.784655e-116
```

From the p-values, we can see strong evidence that non-linear association exists.

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]

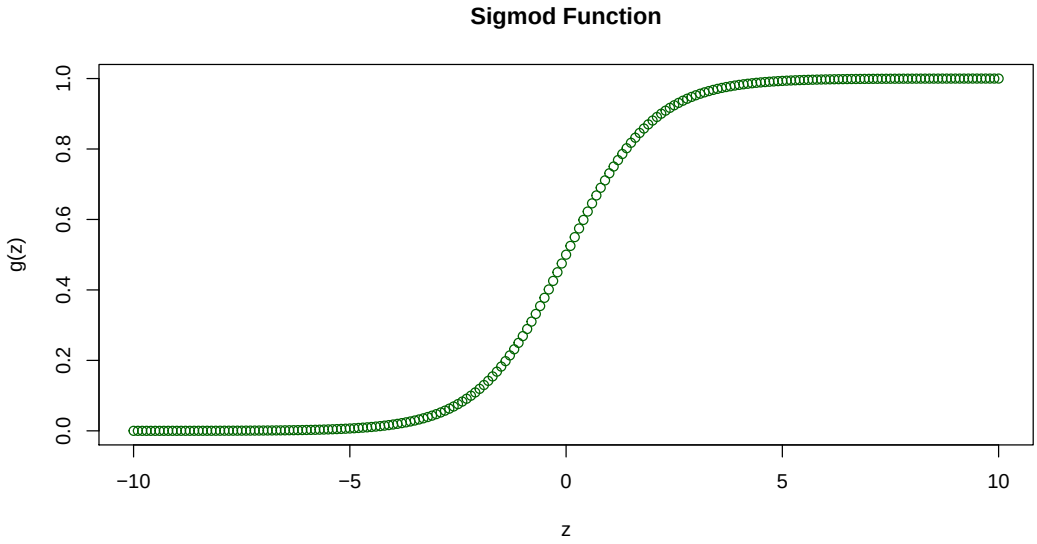
3.2 Perform logistic regression on the new.Auto data set in order to predict "mpg01" using the input variables: cylinders, displacement, horsepower, weight. Report the parameters (θ's). [30 pt]

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

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]

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]

(3.1) The code is in the file `Prob3-Commands.R`, part "Problem 3.1".



(3.2) The code is in the file `Prob3-Commands.R`, part "Problem 3.2", which calls the file `Prob3-Q2.R`, where the actual code lies. The result is

$$\hat{\theta} = (\hat{\theta}_0, \hat{\theta}_1, \cdots, \hat{\theta}_4) = (-0.993182, -0.02205626, -1.358486, -1.621712, -1.652734)$$

(3.3) We calculate the number of errors,

$$\text{number of errors} = \sum_{i=1}^{m} I(I(x^{(i)} \geq 0.5) = \text{mpg01}^{(i)}) = 40$$

where $I(S) = \begin{cases} 1, & \text{if } S \text{ is true} \\ 0, & \text{otherwise} \end{cases}$ is the indicator function. m is the number of data points. Therefore the error rate is

$$\text{error rate} = \frac{\text{number of errors}}{m} = 0.1020408$$

(3.4) The scaled values for x is

$$(x_1^{\text{scaled}}, x_2^{\text{scaled}}, x_3^{\text{scaled}}, x_4^{\text{scaled}}) = (1.4820530, 1.3912695, 2.4818845, 0.6150391)$$

This is performed by subtracting each feature element by the original feature sample mean, and then dividing each feature element by the original feature sample standard deviation. Using the formula, the predicted value of mpg01 canbe calculated by

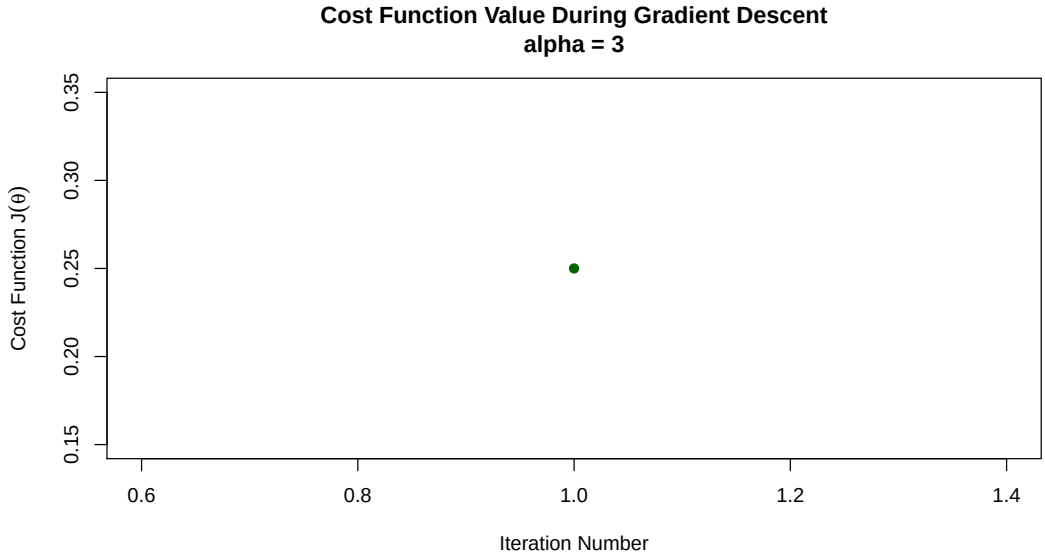
$$(1, x_1^{\text{scaled}}, \cdots, x_4^{\text{scaled}})\hat{\theta} = (1, x_1^{\text{scaled}}, \cdots, x_7^{\text{scaled}})(\hat{\theta}_0, \hat{\theta}_1, \hat{\theta}_2, \cdots, \hat{\theta}_4)^T = 0.0003499786$$

Since $0.0003499786 < 0.5$,

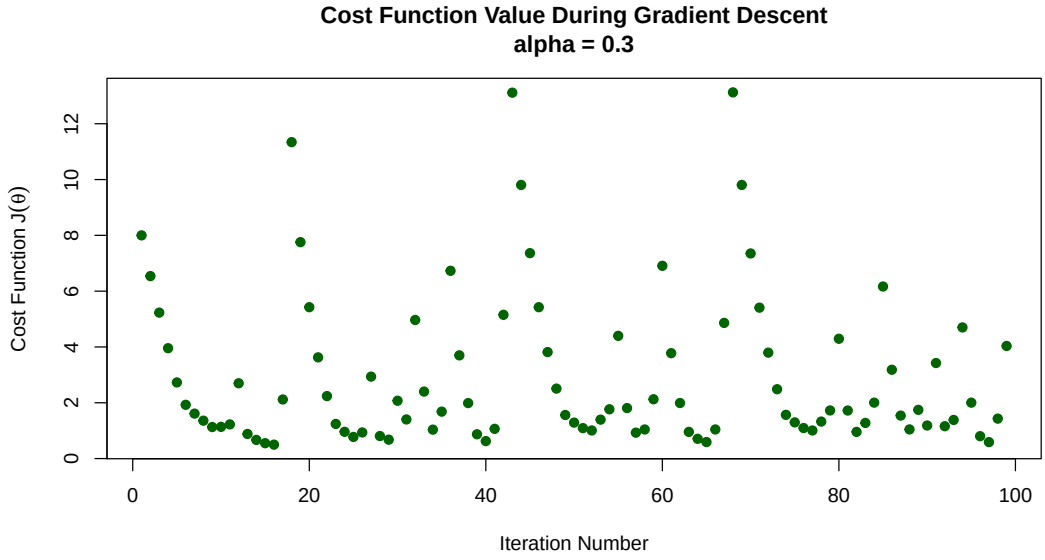
$$\hat{\text{mpg}}01 = 0$$

The `R` code for the calculation of this question can be found in the Problem 3.4 part of the file `Prob3-Commands.R`.

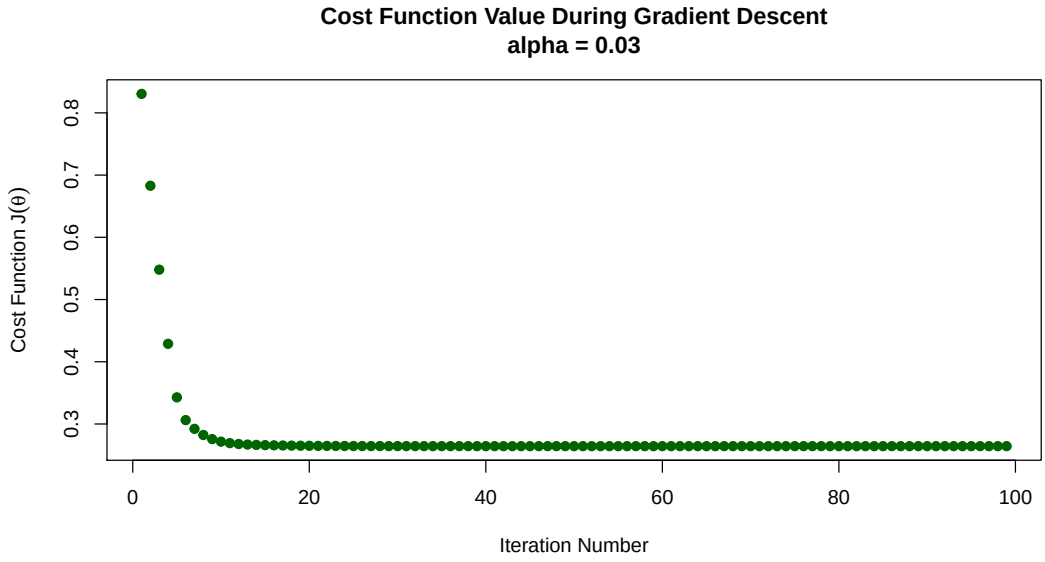
(3.5) With $\alpha_1 = 3$, the method quickly blew up, resulting in just 1 iteration.



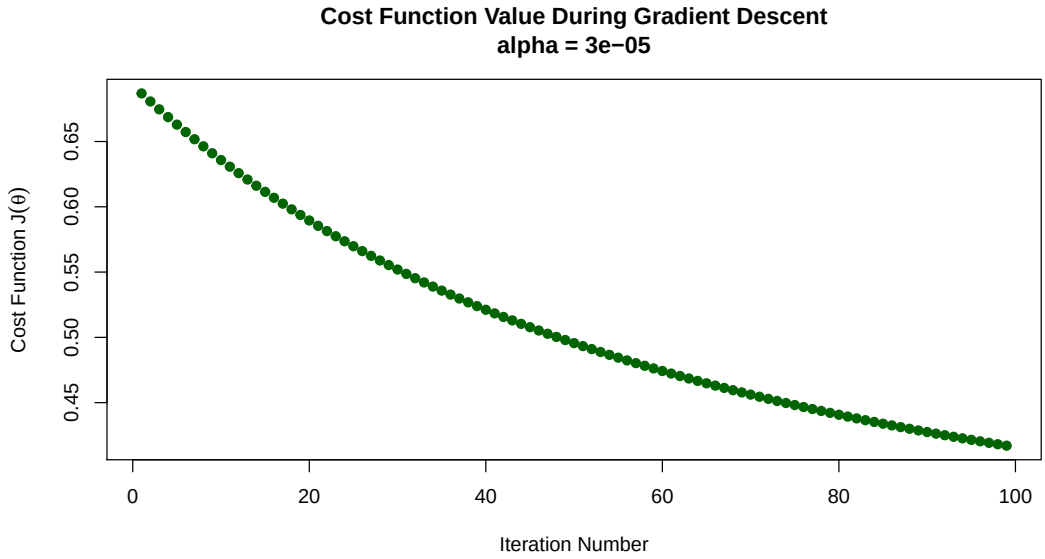
With $\alpha_2 = 0.3$, the method does not seem to converge as the cost function keeps fluctuating. My tolerance on convergence is set to be 1×10^{-20}.



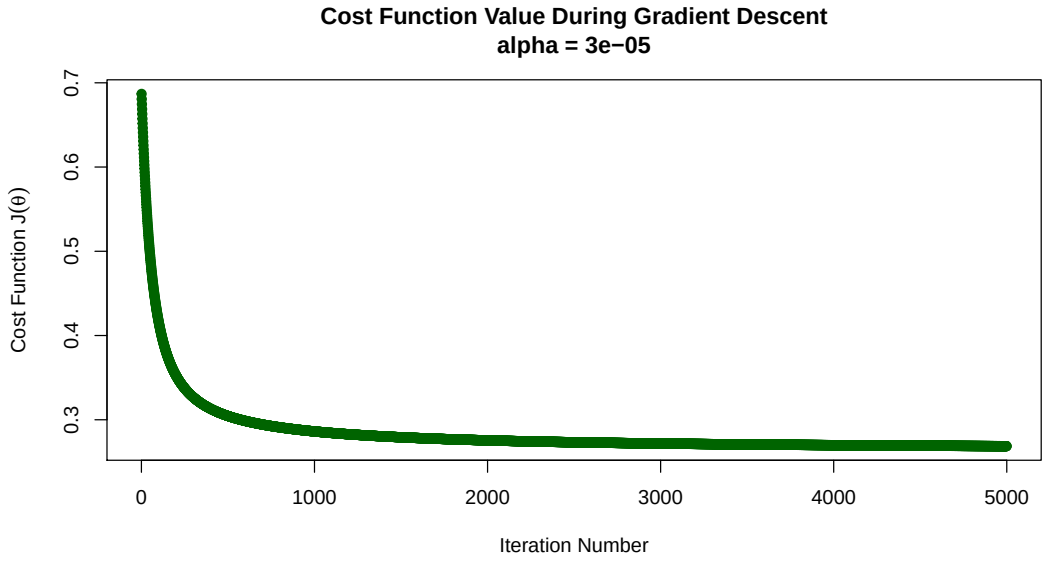
With $\alpha_3 = 0.03$, the method converges very nicely.



With $\alpha_4 = 0.00003$, the method converges, but very slowly.



To illustrate the slow convergence rate for α_4, we give a plot for longer run.



Again, we are using very small tolerance 1×10^{-20}, therefore, the method runs a long time for small α's.

In fact, we found that $\alpha = 0.001$ seems to be a very nice choice for learning rate, resulting in convergence at about 30000 iterations.

Summary of Analysis

As we can see from the plots, when we use $\alpha = 3$, or 0.3, the method does not converge and the cost function blows up or keeps fluctuating. When $\alpha = 0.03$, from the fourth graph, we can see that the method converges very nicely. With $\alpha = 0.03$, from the fourth graph, we can see that the method converges but very slowly.

Therefore, the plots with $\alpha = 0.03$ and 0.00003 look much better than others. And the method converges faster with $\alpha = 0.03$.

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