Multivariate Linear Regression

Wellington S. Villasin

11209534

Carl Michael O. Sy

11208929

Sidney Carlo P. Lopez

11222522

Jeremy B. Reccion

11210192

# Introduction

Multivariate Linear Regression helps in predicting values of a single or multiple responses. It has the same purpose with the linear regression yet in multiple independent variables. This is used to express the continuous of vector response with the combination of linear terms and the normal distribution. Most combinations are used with the combination of gradient descent and with common normal equations.

# Procedures

Procedure 2.0

1. Plot the raw data : (a) housing prices with respect to living area; (b) housing price with respect to the number of bedrooms before pre-processing.
2. Plot the preprocessed data: (a) housing prices with respect to living area; (b) housing prices with respect to the number of bedrooms after pre-processing.

Procedure 2.1

1. Observe the changes in the cost function happens as the learning rate changes. What happens when the learning rate is too small? Too large?
2. What is the best learning rate that you have found?

Procedure 2.2

1. Using the best learning rate that you found, run gradient descent until convergence to find
2. The final values of ϴ = \_\_\_\_\_\_\_\_\_\_\_\_
3. The predicted price of a house with 1650 square feet and 3 bedrooms.

(Don’t forget to scale your features when you make this prediction!)

Housing Price= \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Procedure 2.3

1. In your program, use the closed form solution to a least squares fit formula to calculate ϴ.
2. Remember that while you don’t need to scale your features, you still need to add an intercept term.
3. Once you have found from this method, use it to make a price prediction for a 1650 squarefoot house with 3 bedrooms.
4. Did you get the same price that you found through gradient descent?

Procedures 2.4

1. Provide your own data with at least 50 elements and at least 2 input features. Then , apply the procedures 2.1-2.3 performed in this experiment to your own data.

# Data and Results

## 1st Procedure

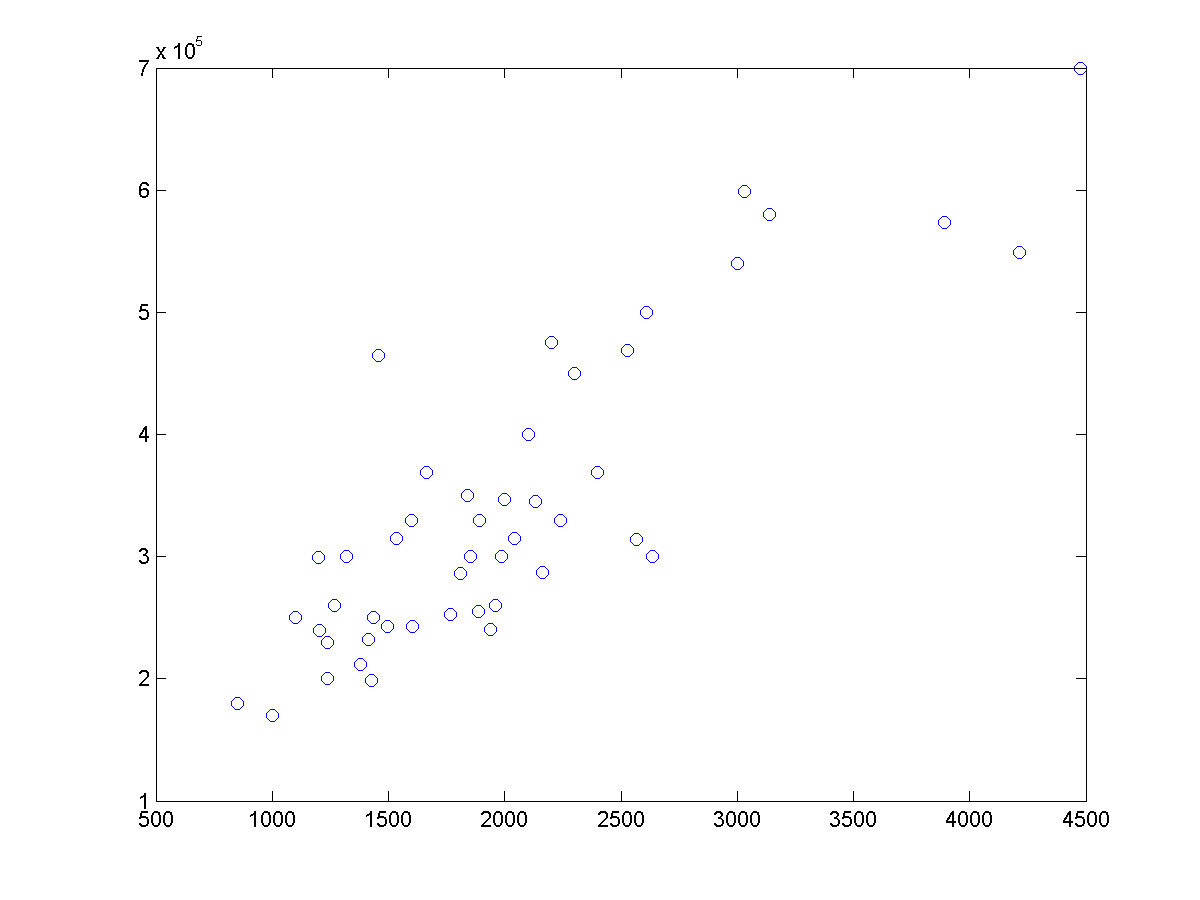
1. – Figure 1 & 2

Fig.1

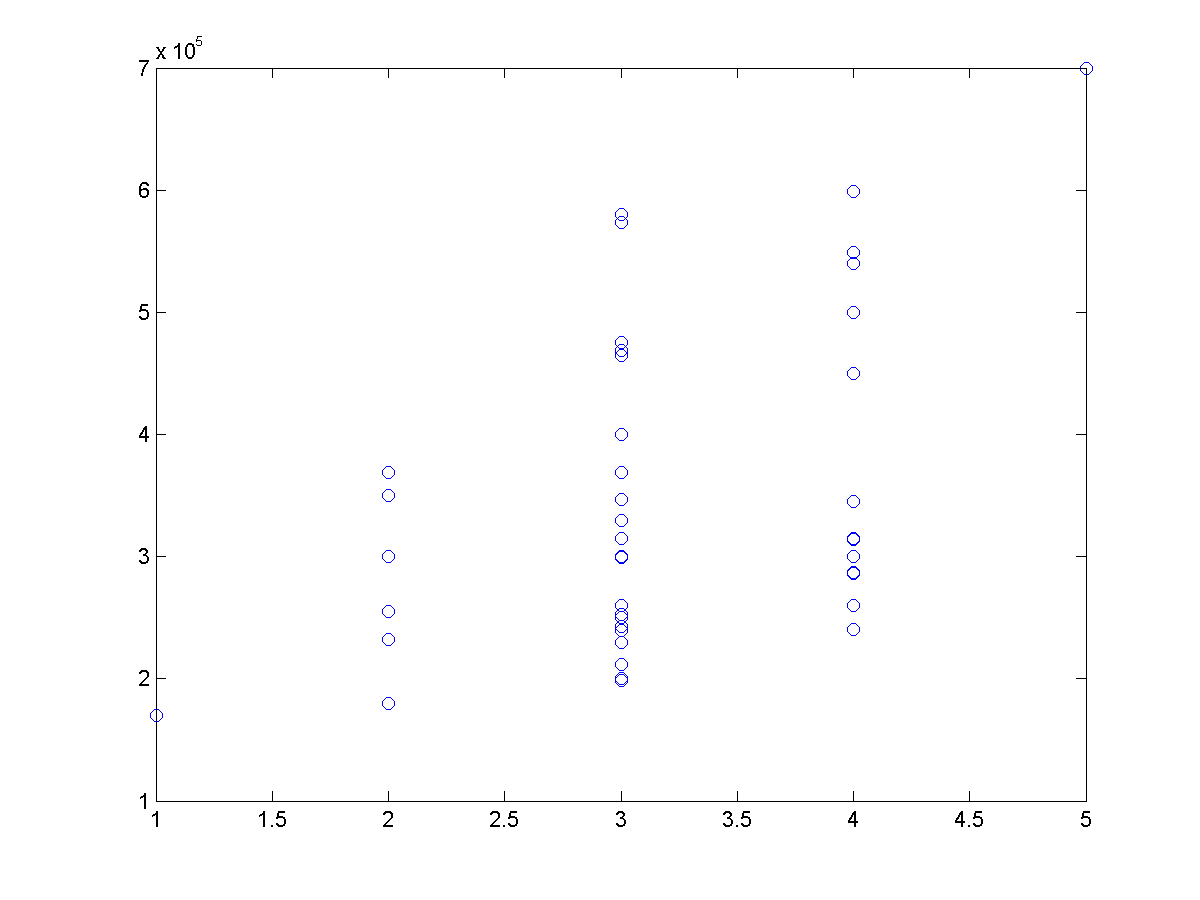
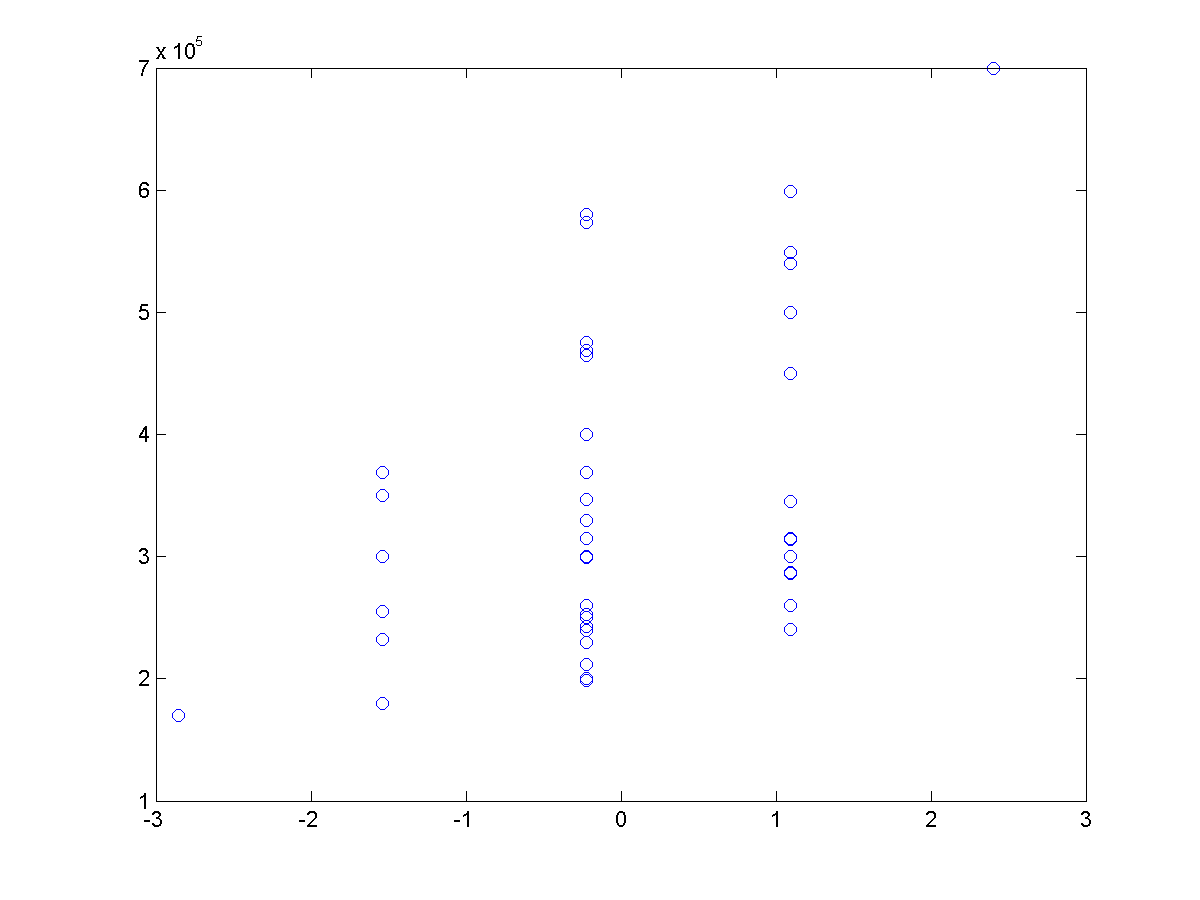
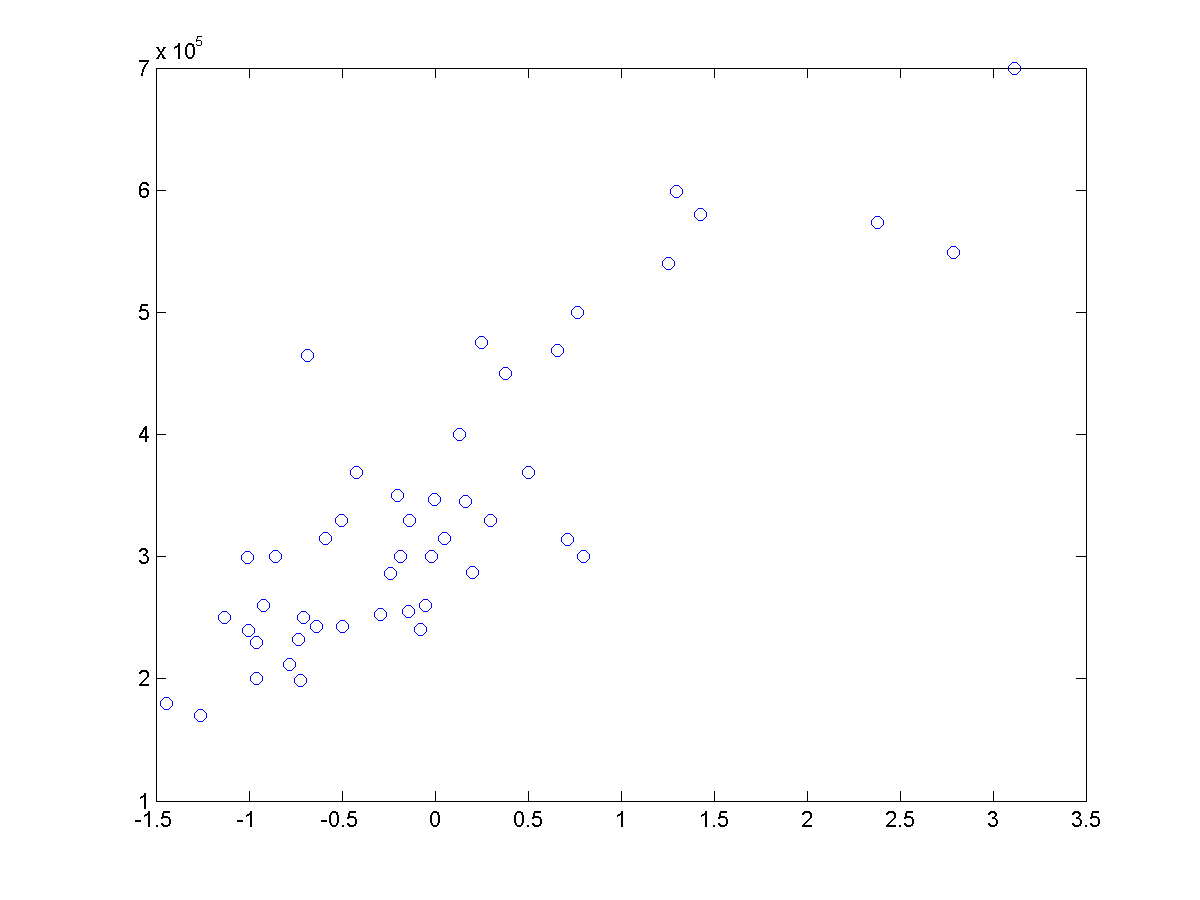


Fig.2

1. – Figure 3 & 4

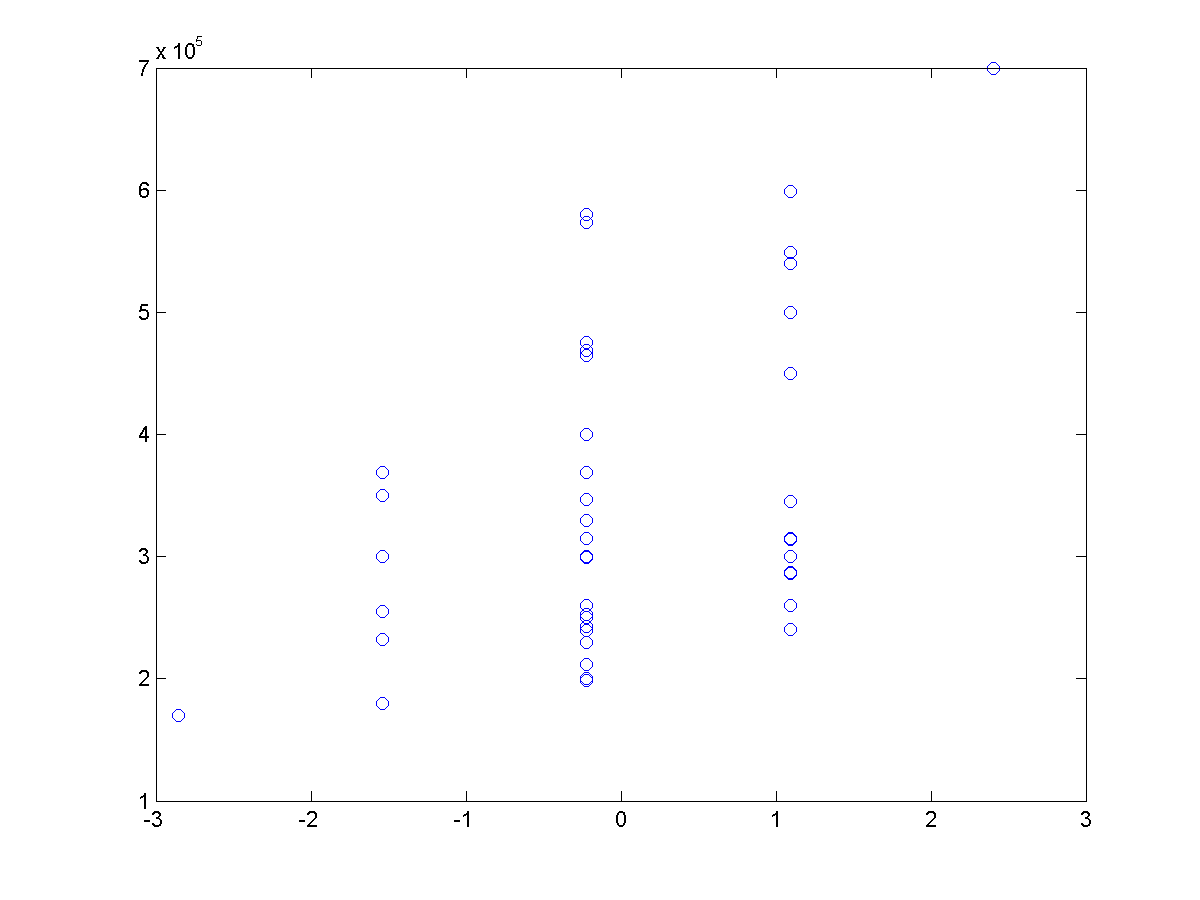
Fig.3

Fig.4

## Procedure 2.1

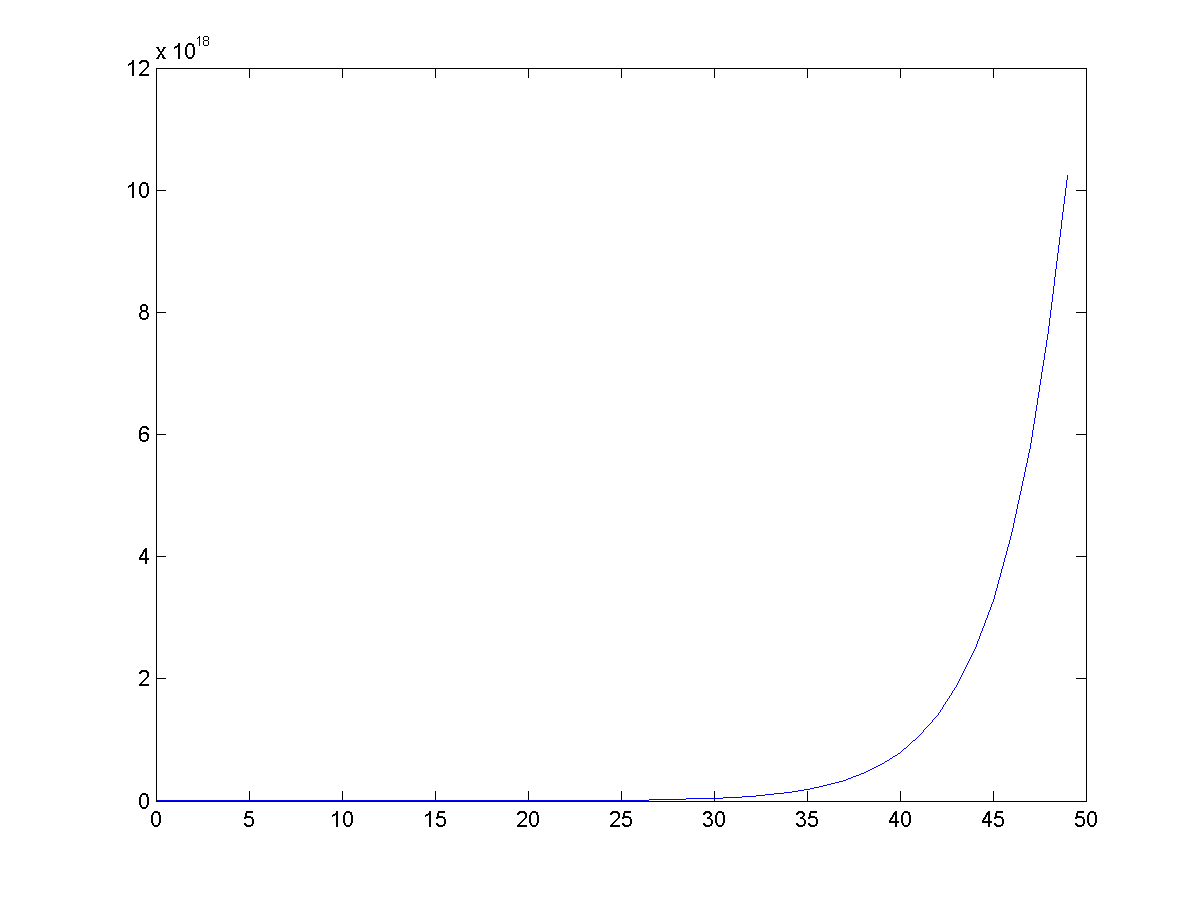
Fig.5

Fig.6

Alpha on the Fig.5 is 0.01. Alpha on Fig.6 is 1.5. Alpha with a smaller value will have a more linear cost J. Alpha with a bigger value will spike at the last iterations. Both are unacceptable regression models.

Best learning rate is 0.8 which is shown in fig.7.

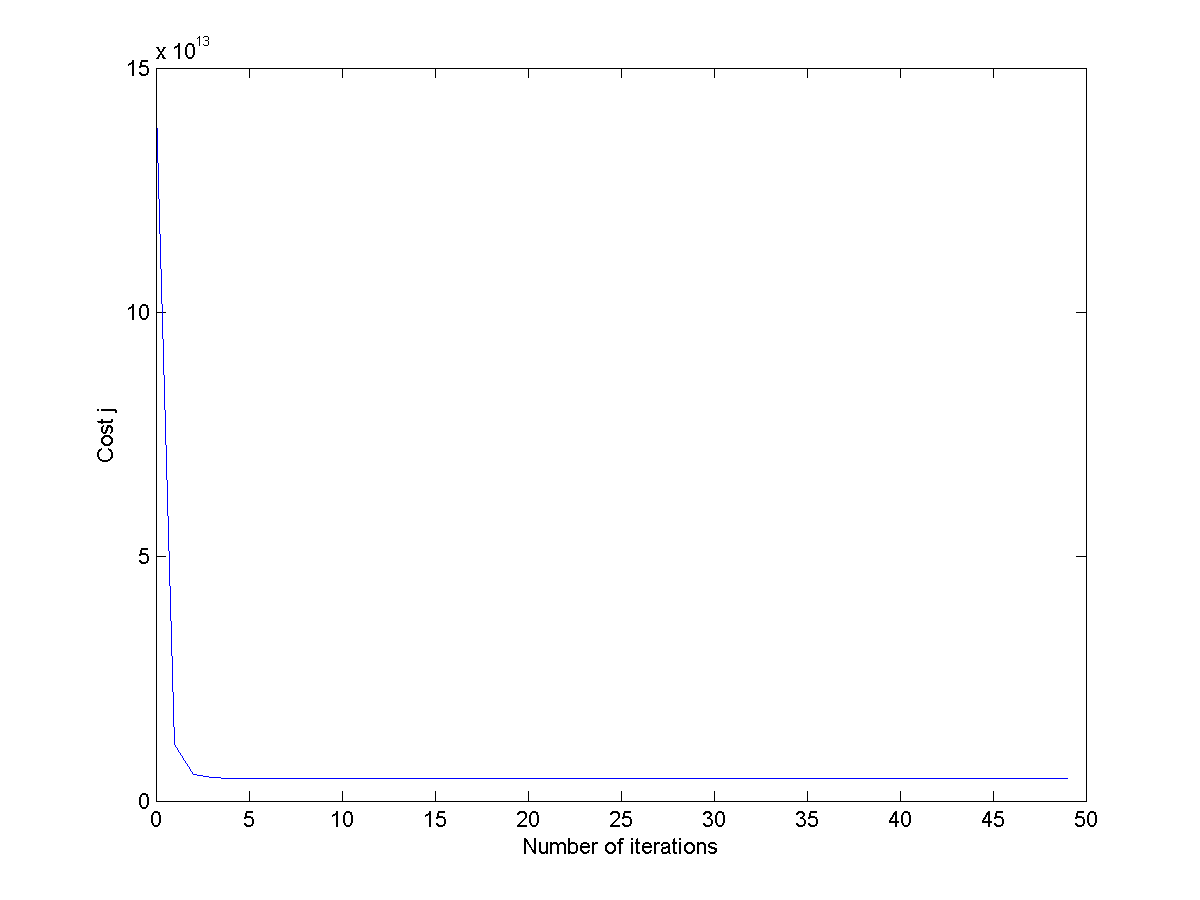
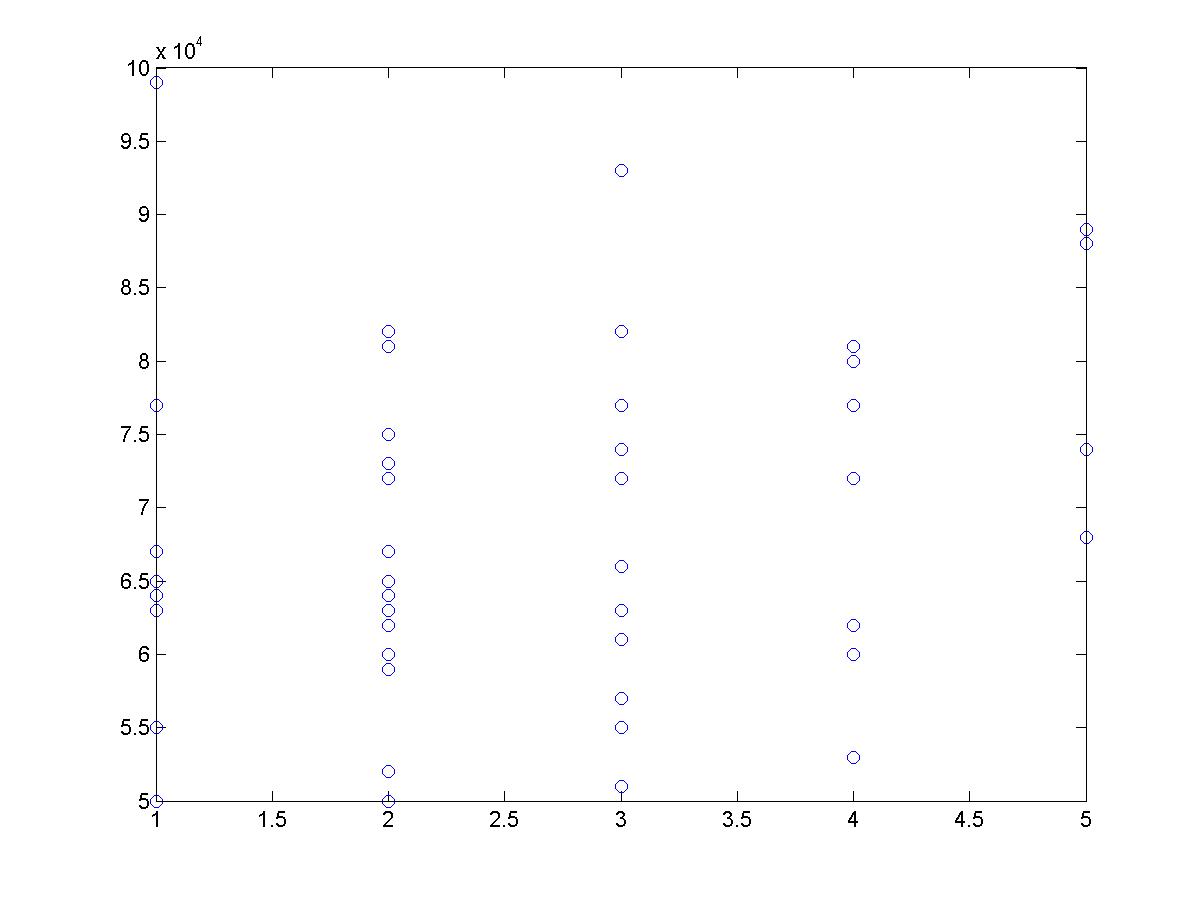


Fig.7

## Procedure 2.2

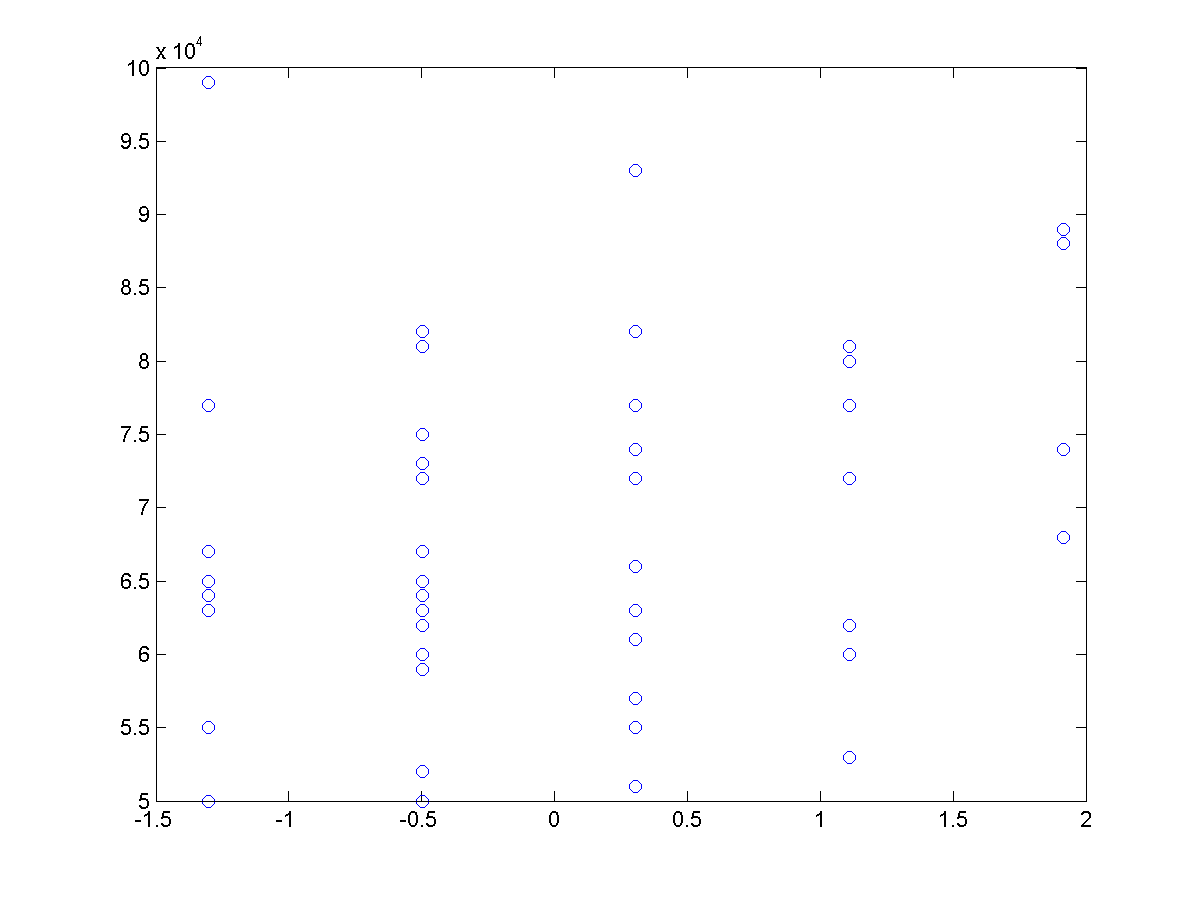
Theta = [3.404126595744681e+05; 1.106310500926826e+05; -6.649474084656331e+03]

House price = 2.930814643434721e+05

## Procedure 2.3

Theta = [8.959790954279870e+04; 1.392106740176292e+02; -8.738019112327724e+03]

House price = 2.930814643434721e+05

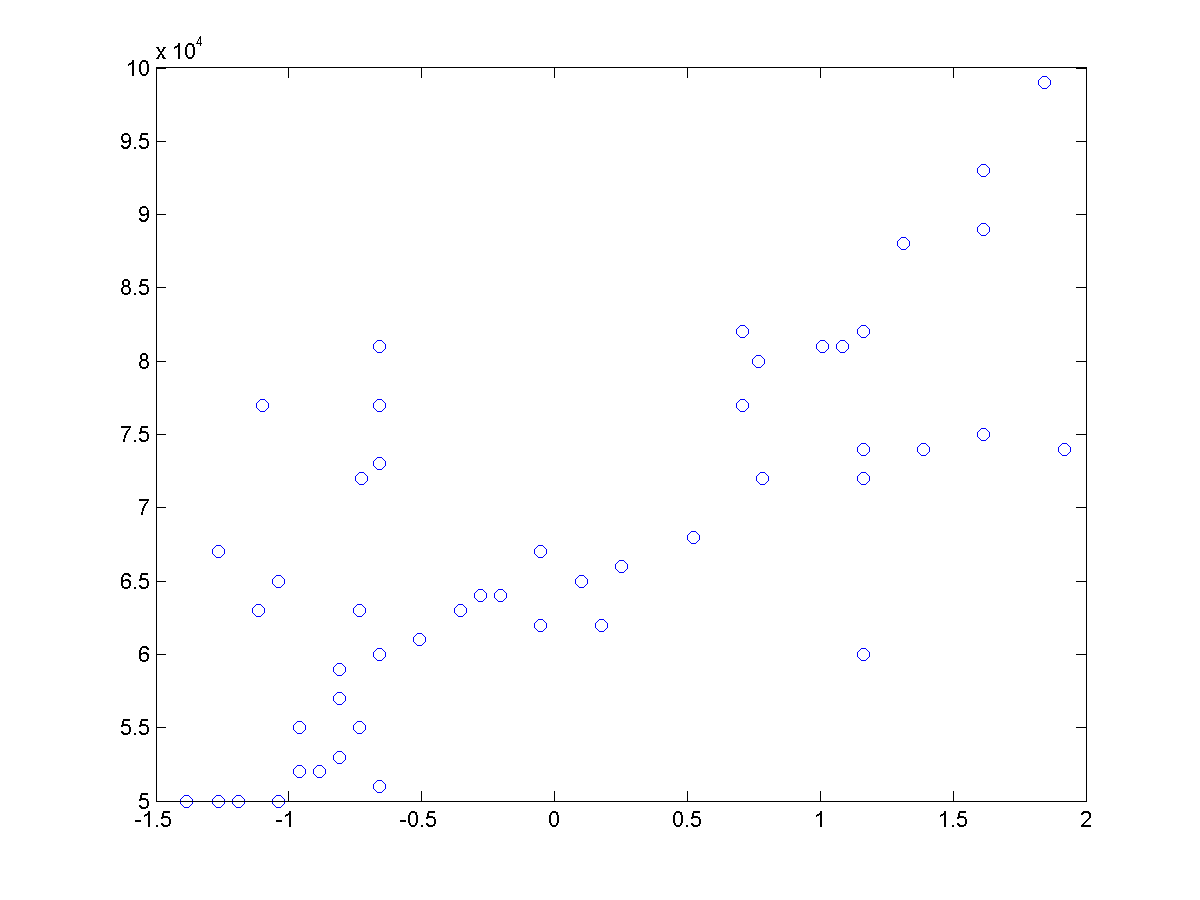
Same price? Answer: Yes

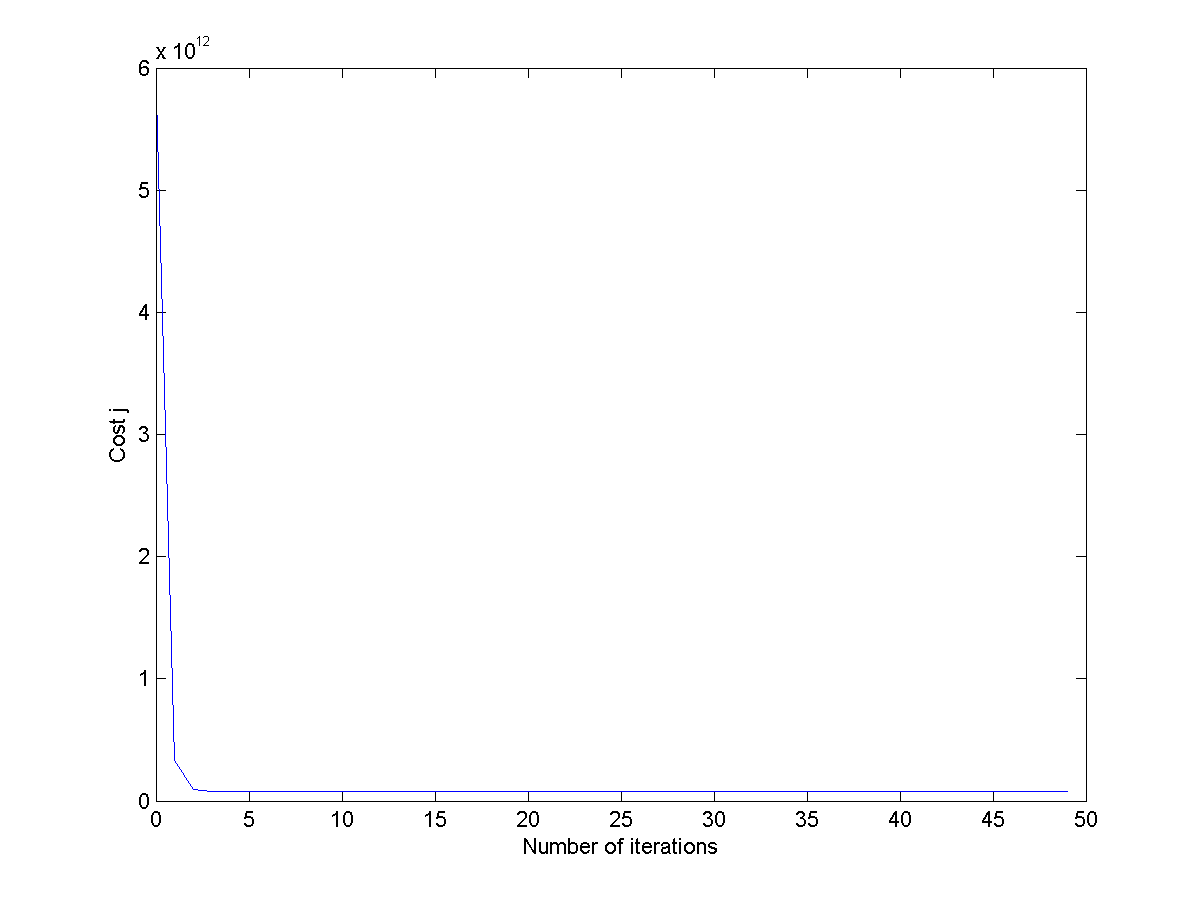
## Procedure 2.4

Same codes but change m to 50 since own data has 50 elements.

## Own Data

2.0: (1)

 (2)

2.1:

Alpha = 0.08

2.2:

Theta = [67760; 1.046810142886752e+04; -2.537768929891468e+03];

House price = 6.723689708232108e+04

2.3:

Theta = [4.717146225136247e+04; 15.871179017825920; -2.040670184882147e+03]

House price = 6.723689707612881e+04

Same price? Answer: Yes

# Codes

x = load('ex2x.dat');

y = load('ex2y.dat');

plot(x(:,1),y,'o');

print('plot1','-dpng');

plot(x(:,2),y,'o');

print('plot2','-dpng');

x = [ones(47,1),x];

%preprocessing

sigma = std(x);

mu = mean(x);

x(:,2) = (x(:,2) - mu(2))./ sigma(2);

x(:,3) = (x(:,3) - mu(3))./ sigma(3);

plot(x(:,2),y,'o');

print('plot3','-dpng');

plot(x(:,3),y,'o');

print('plot4','-dpng');

%cost J

alpha = 0.8;

j = zeros(50,1);

theta = zeros(size(x(1,:)))';

for num = 1:50

j(num) = (0.5 \* 47) \* (x \* theta - y)' \* (x \* theta - y);

theta = theta - (alpha / 47) \* x' \*(x \* theta - y);

end

figure;

plot(0:49,j(1:50),'-');

xlabel('Number of iterations');

ylabel('Cost j');

%too small alpha

alpha = 0.01;

>> theta = zeros(size(x(1,:)))';

for num = 1:50

j(num) = (0.5 \* 47) \* (x \* theta - y)' \* (x \* theta - y);

theta = theta - (alpha / 47) \* x' \*(x \* theta - y);

end

plot(0:49,j(1:50),'-');

%too big alpha

alpha = 1.5;

theta = zeros(size(x(1,:)))';

for num = 1:50

j(num) = (0.5 \* 47) \* (x \* theta - y)' \* (x \* theta - y);

theta = theta - (alpha / 47) \* x' \*(x \* theta - y);

end

plot(0:49,j(1:50),'-');

%house price

s\_area = (1650 - mu(2))./ sigma(2);

s\_room = (3 - mu(3))./ sigma(3);

price = s\_room \* theta(3) + s\_area \* theta(2) + theta(1);

%normal equation (no preprocessing)

theta = inv(x' \* x) \* x' \* y;

price = theta(1) + 1650 \* theta(2) + 3 \* theta(3);

%own data

x1 = [720 2140 910 1155 1200 2400 1500 2900 800 1000 1100 1150 1200 1200 1200 900 950 800];

x1 = [x1, 850 1000 1100 1200 1150 1050 1300 2400 2500 2700 2300 2100 1400 1450 1600 1800 2100];

x1 = [x1, 2400 2700 2550 2150 1980 1750 2350 2400 2700 2850 1700 1600 1300 1100 950];

x2 = [1 4 1 2 2 4 2 5 1 1 2 2 3 2 2 1 1 1 1 2 4 3 3 2 3 3 5 5 4 2 3 1 2 3 4 5 2 3 4 5 2 2 ];

x2 = [x2, 3 3 1 2 4 3 3 2];

x = [x1', x2'];

y = [50000 80000 77000 72000 73000 60000 64000 74000 50000 55000 59000 63000 77000 81000 60000];

y = [y, 63000 65000 67000 50000 52000 53000 51000 55000 52000 61000 72000 88000 89000 81000 82000];

y = [y, 63000 64000 67000 66000 77000 74000 75000 74000 72000 68000 62000 81000 82000 93000 99000 65000 62000 61000 57000 50000];

y = y'

# CONCLUSION

Overall, the experiment showed how Multivariate Linear Regression is utilized in solving problems involving multiple independent variable. The experiment was sucessfully executed when the result from the machine computed output, which used gradient descent, was compared to the actual computation from normal equations without the other methods such as preprocessing. The housing prices from the gradient descent and normal equation is compared to be approximately equal, and this concludes the correct usage of the gradient descent for multivariate linear regression. Alpha or the learning rate was an important factor in using gradient descent, since it alters the accuracy of results and the group successfully figured out that smaller learning rate results to a more linear curve while a larger alpha would result to almost ideal output but with spikes at the latter part of the graph. Thus, a good learning curve should not be too high nor too low. This concludes the learning for this experiment.

# REFERENCE

[1] Y. Personnel, "Yale Education," Yale University,[Online].Available: http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm. [Accessed 8 September 2015].