Code for Question 2:

clc;

clear all;

close all;

% Open the data files

load x\_train.mat;

load y\_train.mat;

load x\_test.mat;

load y\_test.mat;

% Plot the scatter plot

figure(1)

scatter(x\_train,y\_train);

title('Scatter plot between x train and y train')

xlabel('x train')

ylabel('y train')

% Size of data points of training set

row = ((size(x\_train,1)));

col = ((size(x\_train,2)));

% Augment the training data and make polynomials of degree [1, 2, 3, 7, 10]

ON = ones(row,col);

X\_1dim = [ON x\_train];

X\_2dim = [ON x\_train x\_train.^2];

X\_3dim = [ON x\_train x\_train.^2 x\_train.^3];

X\_7dim = [ON x\_train x\_train.^2 x\_train.^3 x\_train.^4 x\_train.^5 x\_train.^6 x\_train.^7];

X\_10dim = [ON x\_train x\_train.^2 x\_train.^3 x\_train.^4 x\_train.^5 x\_train.^6 x\_train.^7 x\_train.^8 x\_train.^9 x\_train.^10];

% Make the Weight Vectors

Weight\_1dim = pinv(X\_1dim'\*X\_1dim)\*X\_1dim'\*y\_train;

Weight\_2dim = pinv(X\_2dim'\*X\_2dim)\*X\_2dim'\*y\_train;

Weight\_3dim = pinv(X\_3dim'\*X\_3dim)\*X\_3dim'\*y\_train;

Weight\_7dim = pinv(X\_7dim'\*X\_7dim)\*X\_7dim'\*y\_train;

Weight\_10dim = pinv(X\_10dim'\*X\_10dim)\*X\_10dim'\*y\_train;

% Define the Mean Square Error Variables

MSE\_1 = 0;

MSE\_2 = 0;

MSE\_3 = 0;

MSE\_4 = 0;

MSE\_5 = 0;

% Compute the Mean Square Error

for i = 1 : row

MSE\_1 = MSE\_1 + ((y\_train(i) - Weight\_1dim'\*X\_1dim(i,:)'))^2;

MSE\_2 = MSE\_2 + ((y\_train(i) - Weight\_2dim'\*X\_2dim(i,:)'))^2;

MSE\_3 = MSE\_3 + ((y\_train(i) - Weight\_3dim'\*X\_3dim(i,:)'))^2;

MSE\_4 = MSE\_4 + ((y\_train(i) - Weight\_7dim'\*X\_7dim(i,:)'))^2;

MSE\_5 = MSE\_5 + ((y\_train(i) - Weight\_10dim'\*X\_10dim(i,:)'))^2;

end

MSE\_1 = MSE\_1/25;

MSE\_2 = MSE\_2/25;

MSE\_3 = MSE\_3/25;

MSE\_4 = MSE\_4/25;

MSE\_5 = MSE\_5/25;

% Plot the graph of MSE vs Polynomial Degree

deg = [1;2;3;7;10];

MSE = [MSE\_1;MSE\_2;MSE\_3;MSE\_4;MSE\_5];

figure(2)

plot(deg,MSE,'-o')

title('Plot of MSE vs polynomial degree for training sample')

xlabel('Polynomial degree')

ylabel('Mean Square Error value')

txt1 = num2str(MSE\_1);

text(deg(1),MSE(1),txt1);

txt2 = num2str(MSE\_2);

text(deg(2),MSE(2),txt2);

txt3 = num2str(MSE\_3);

text(deg(3),MSE(3),txt3);

txt4 = num2str(MSE\_4);

text(deg(4),MSE(4),txt4);

txt5 = num2str(MSE\_5);

text(deg(5),MSE(5),txt5);

% Size of data points of test set

row1 = ((size(x\_test,1)));

col1 = ((size(x\_test,2)));

% Augment the test data and make polynomials of degree [1, 2, 3, 7, 10]

ON1 = ones(row1,col1);

XT\_1dim = [ON1 x\_test];

XT\_2dim = [ON1 x\_test x\_test.^2];

XT\_3dim = [ON1 x\_test x\_test.^2 x\_test.^3];

XT\_7dim = [ON1 x\_test x\_test.^2 x\_test.^3 x\_test.^4 x\_test.^5 x\_test.^6 x\_test.^7];

XT\_10dim = [ON1 x\_test x\_test.^2 x\_test.^3 x\_test.^4 x\_test.^5 x\_test.^6 x\_test.^7 x\_test.^8 x\_test.^9 x\_test.^10];

% Define the Mean Square Error Variables

MSE\_t1 = 0;

MSE\_t2 = 0;

MSE\_t3 = 0;

MSE\_t4 = 0;

MSE\_t5 = 0;

% Compute the Mean Square Error

for i = 1 : row1

MSE\_t1 = MSE\_t1 + ((y\_test(i) - Weight\_1dim'\*XT\_1dim(i,:)'))^2;

MSE\_t2 = MSE\_t2 + ((y\_test(i) - Weight\_2dim'\*XT\_2dim(i,:)'))^2;

MSE\_t3 = MSE\_t3 + ((y\_test(i) - Weight\_3dim'\*XT\_3dim(i,:)'))^2;

MSE\_t4 = MSE\_t4 + ((y\_test(i) - Weight\_7dim'\*XT\_7dim(i,:)'))^2;

MSE\_t5 = MSE\_t5 + ((y\_test(i) - Weight\_10dim'\*XT\_10dim(i,:)'))^2;

end

MSE\_t1 = MSE\_t1/25;

MSE\_t2 = MSE\_t2/25;

MSE\_t3 = MSE\_t3/25;

MSE\_t4 = MSE\_t4/25;

MSE\_t5 = MSE\_t5/25;

% Plot the graph of MSE vs Polynomial Degree

deg = [1;2;3;7;10];

MSE = [MSE\_t1;MSE\_t2;MSE\_t3;MSE\_t4;MSE\_t5];

figure(3)

plot(deg,MSE,'-o')

title('Plot of MSE vs polynomial degree for test sample')

xlabel('Polynomial degree')

ylabel('Mean Square Error value')

txt1 = num2str(MSE\_t1);

text(deg(1),MSE(1),txt1);

txt2 = num2str(MSE\_t2);

text(deg(2),MSE(2),txt2);

txt3 = num2str(MSE\_t3);

text(deg(3),MSE(3),txt3);

txt4 = num2str(MSE\_t4);

text(deg(4),MSE(4),txt4);

txt5 = num2str(MSE\_t5);

text(deg(5),MSE(5),txt5);

% Ridge Regression

% Values of lambda

lambda = [1e-5, 1e-3, 0.1, 1, 10];

% Size of data points of training set

row = ((size(x\_train,1)));

col = ((size(x\_train,2)));

% Augment the training data and make polynomial of degree 7

ON = ones(row,col);

X\_7dim = [ON x\_train x\_train.^2 x\_train.^3 x\_train.^4 x\_train.^5 x\_train.^6 x\_train.^7];

% Define the MSE variable

MSE = zeros(size(lambda));

% Compute the Mean Square Error

for j = 1:size(lambda,2)

for i = 1:row

weight = pinv(lambda(j)\*eye(8) + X\_7dim'\*X\_7dim)\*X\_7dim'\*y\_train;

MSE(j) = MSE(j) + ((y\_train(i) - weight'\*X\_7dim(i,:)'))^2 + lambda(j)\*(weight'\*weight)/25;

end

weight;

end

MSE = MSE/25;

% Plot the curve for training dataset error

figure(4)

loglog(lambda, MSE)

title('Plot of MSE vs log(lambda)')

xlabel('lambda')

ylabel('Mean Square Error value')

figure(5)

lambdas = log10(lambda);

scatter(lambdas,MSE)

hold on;

plot(lambdas,MSE)

title('Plot of MSE vs log(lambda)')

xlabel('Log(lambda)')

ylabel('Mean Square Error value')

% Size of data points of test set

row1 = ((size(x\_test,1)));

col1 = ((size(x\_test,2)));

% Augment the test data and make polynomials of degree [1, 2, 3, 7, 10]

ON1 = ones(row1,col1);

XT\_7dim = [ON1 x\_test x\_test.^2 x\_test.^3 x\_test.^4 x\_test.^5 x\_test.^6 x\_test.^7];

% Define the MSE variable

MSE = zeros(size(lambda));

% Compute the Mean Square Error

for j = 1:size(lambda,2)

for i = 1:row1

weight = pinv(lambda(j)\*eye(8) + X\_7dim'\*X\_7dim)\*X\_7dim'\*y\_train;

MSE(j) = MSE(j) + ((y\_test(i) - weight'\*XT\_7dim(i,:)'))^2 + lambda(j)\*(weight'\*weight)/25;

end

weight;

end

MSE = MSE/48

% Plot the curve for training dataset error

figure(6)

loglog(lambda, MSE)

title('Plot of MSE vs log(lambda)')

xlabel('lambda')

ylabel('Mean Square Error value')

figure(7)

lambdas = log10(lambda);

scatter(lambdas,MSE)

hold on;

plot(lambdas,MSE)

title('Plot of MSE vs log(lambda)')

xlabel('Log(lambda)')

ylabel('Mean Square Error value')