function [ ans ] = LeastSquares( x , y )

%LEASTSQUARES This function takes in two matrices x and y, which are the

%original data points, and then outputs the ans matrix which contains the b and m

%for the line of best fit.

% For linear - create the matix A with x being the first column and b or

% ones being the second column

a(:,1)= x;

a(:,2)= ones(size(x));

% For quadratic - create the matrix A with x^2 being the first column, x

% being the second column, and b or ones being the third column

a1(:,1)= x .^ 2;

a1(:,2)= x;

a1(:,3)= ones(size(x));

% y stays the same for both

b(:,1)=y;

%Transpose of matrix A for linear

aT= transpose(a);

%Transpose of matrix A1 for quadratic

aT1 = transpose(a1);

%Answer stored in a matrix called mAns for the linear model

mAns = inv(aT \* a) \* (aT \*b);

%Answer stored in a matrix called mAns1 for the quadratic model

mAns1 = inv(aT1 \* a1) \* (aT1 \*b);

%For the linear model. m is the slope while the yint is the y-intercept

%inside of the mAns matrix

m = mAns(1);

yint= mAns(2);

%For the quadratic model. ax is a, bx is b, and yint1 is the y-intercept

%inside of the mAns1 matrix

ax = mAns1(1);

bx = mAns1(2);

yint1 = mAns1(3);

ans = sprintf('Linear best-fit model: y = %d x + %d\n Quadratic best-fit model: y = %d x^2 + %d x + %d\n', m, yint, ax, bx, yint1);

% constructing the best fit line using the estimated slope and constant

yEst = mAns(1)\*x + mAns(2);

yEst1 = mAns1(1)\*x.^2 + mAns1(2)\*x + mAns1(3);

plot (yEst)

hold on

plot (yEst1)

legend('Estimated linear line', 'Estimated quadratic line')

grid on

ylabel('Observations')

xlabel('x-axis')

title('Least squares Best-fit')

end

Output:



