Table of Contents

Exercise 1: Solve a simple linear regression problem

Farzad Rezazadeh Pilehdarboni, 25.05.2022

```
% In this excercise you will create some simulated data and will fit simple
% linear regression models to it. Make sure to use rng(1998) prior to
% staring part a) to ensure consistent results.

clear all
close all
clc
rng(1000);                          % specify seed
i = 0;
NAll = [10 100 1000];

for li = 1:length(NAll)

i = i + 2;
```

a)

```
% Using the linspace() function, create a vector, x, containing N=100
% equidistant observations between 1 and 100. This represents a feature, X.
% By default, linspace() generates a row vector. Use the transpose X = X'
% to obtain a column vector.

N = NAll(li);
X = linspace(1,100,N);              % generate equidistant vector of length N
X = X';
```

b)

```
% Using the randn() function, create a vector, eps, containing 100
```

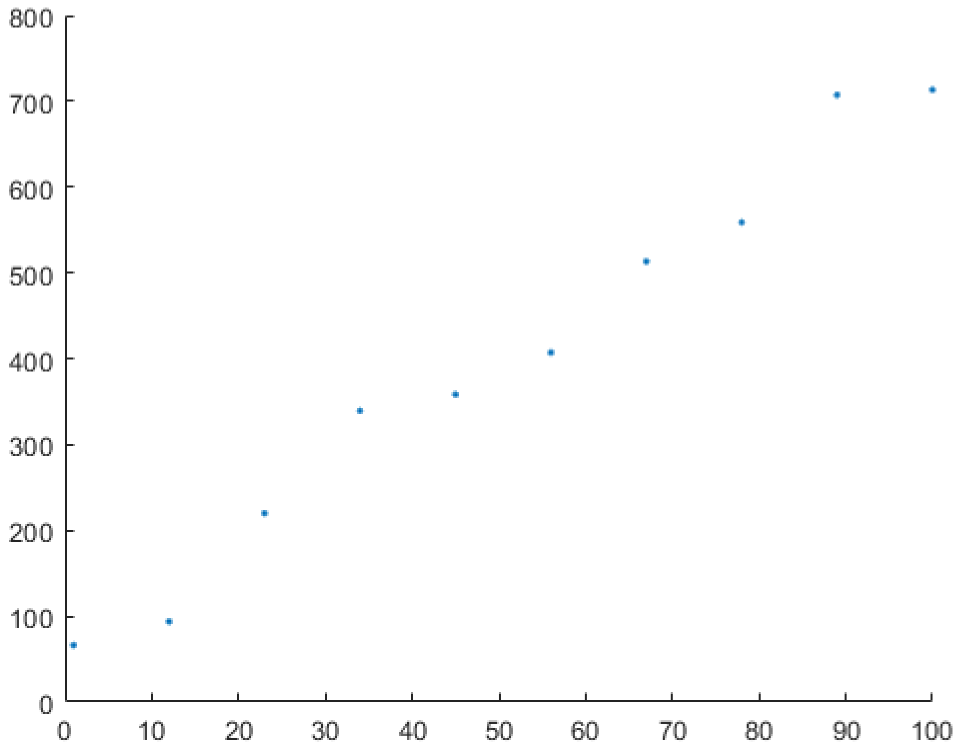
```
% observations drawn from N(0,30^2) distribution, i.e., a normal distribution
% with mean zero and standard deviation 30.

sd = 30;
eps = sd*randn(N,1);
```

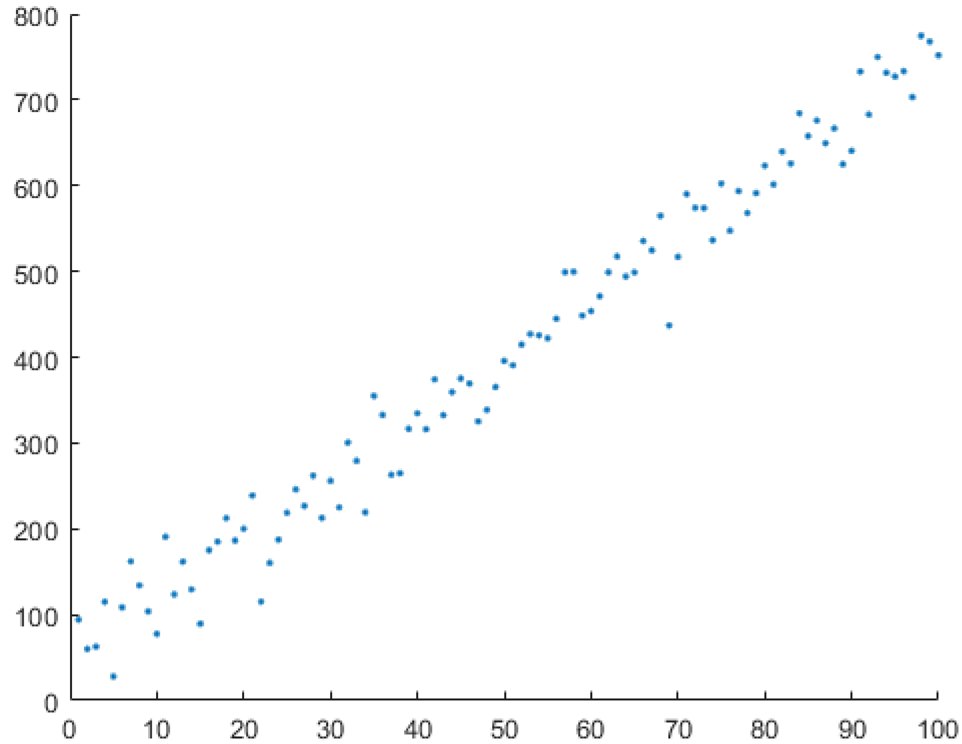
c)

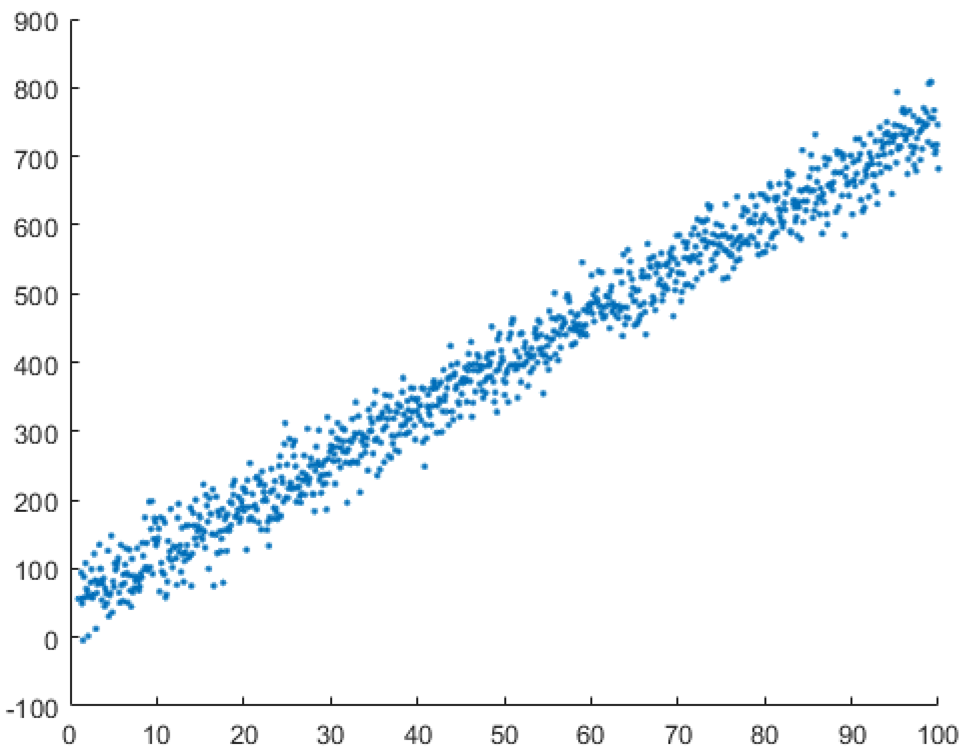
```
% Using x and eps, generate a vector y according to the model
% Y0 = 50 + 7*X
% and add the noise to obtain the disturbed output Y = Y0 + eps.
% What is the length of the vector y? What are the values of \beta_0 and
% \beta_1 in this linear model?

beta_0 = 50;
beta_1 = 7;
Y0 = beta_0 + beta_1*X;
Y = Y0 + eps;      % generate disturbed output
```

d)

```
% Using scatter(), create a scatterplot displaying the relationship
% between x and y. Comment on what you observe.
figure(i)
scatter(X,Y, '.')
```




800
700
600
500
400
300
200
100
0
0
10
20
30
40
50
60
70
80
90
100

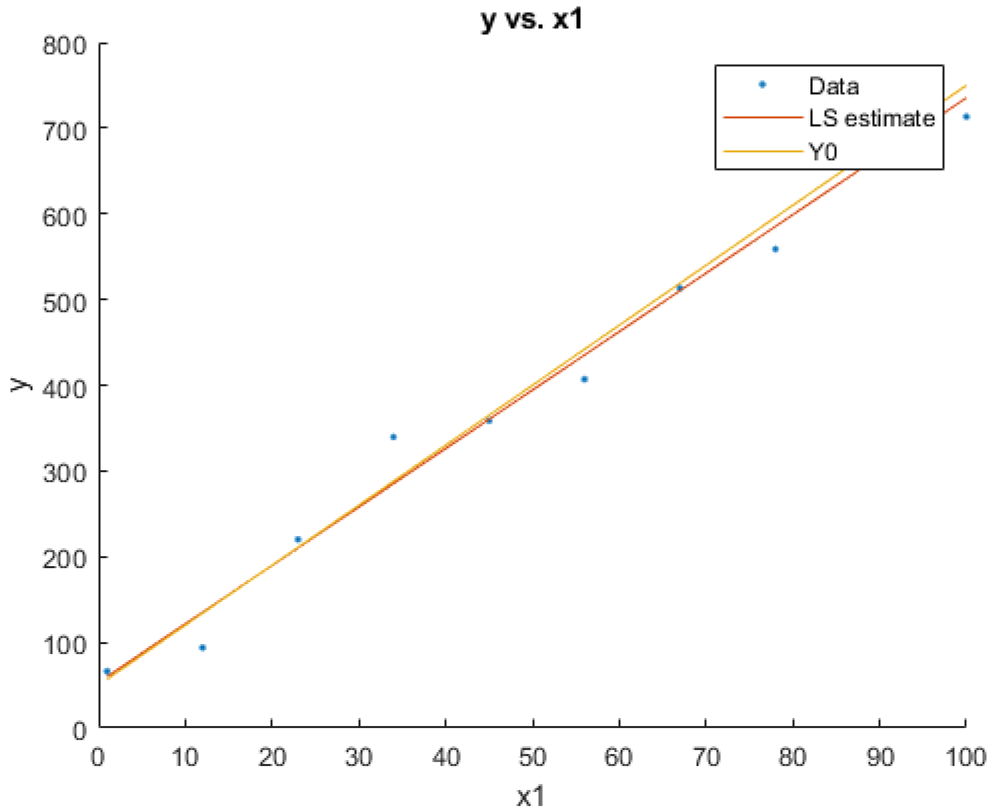

900
800
700
600
500
400
300
200
100
0
-100
0
10
20
30
40
50
60
70
80
90
100

e)

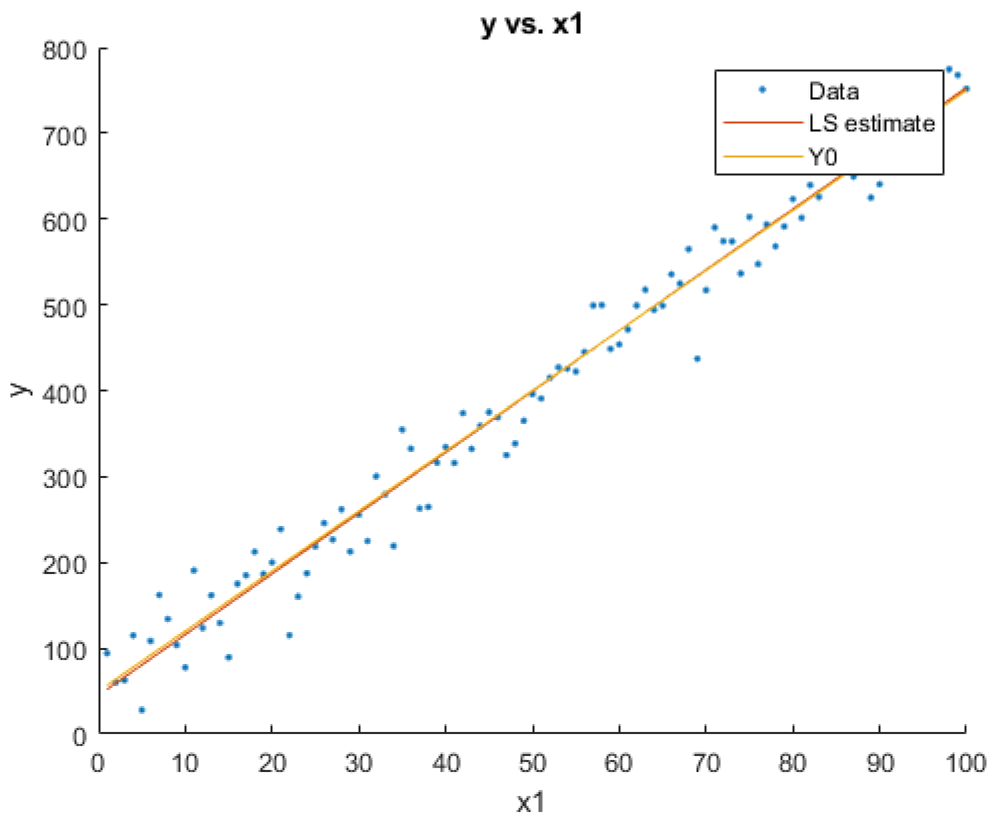
```
% Using the backslash operator, fit a least squares linear model to
% predict y using x. First, create the regression matrix using
% Phi=[ones(size(x)) x]. Comment on the model obtained. How do
% \hat{\beta}_0 and \hat{\beta}_1 compare to \beta_0 and \beta_1?

Phi = [ones(size(X)) X];
theta = Phi\Y;
```

f)

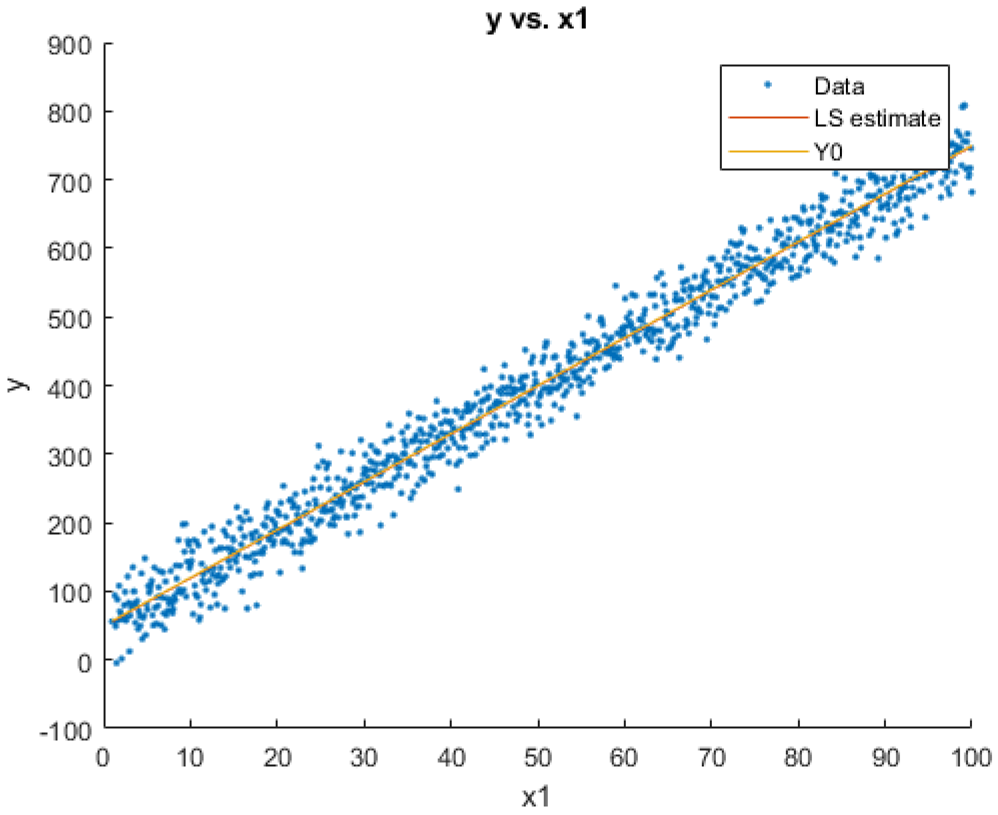
```
% Calculate the predictions \hat{Y} on the training data set. Display the
 least
% squares line on the scatterplot obtained in d) using hold all. Additonally,
 draw the
% population regression line on the plot.

Y_hat = Phi*theta;

hold all
plot(X,Y_hat)
plot(X,Y0)
legend('Data','LS estimate','Y0')
title('y vs. x1')
xlabel('x1')
ylabel('y')
```



y vs. x1
Data
LS estimate
Y0
y
x1



y vs. x1
Data
LS estimate
Y0
y
x1

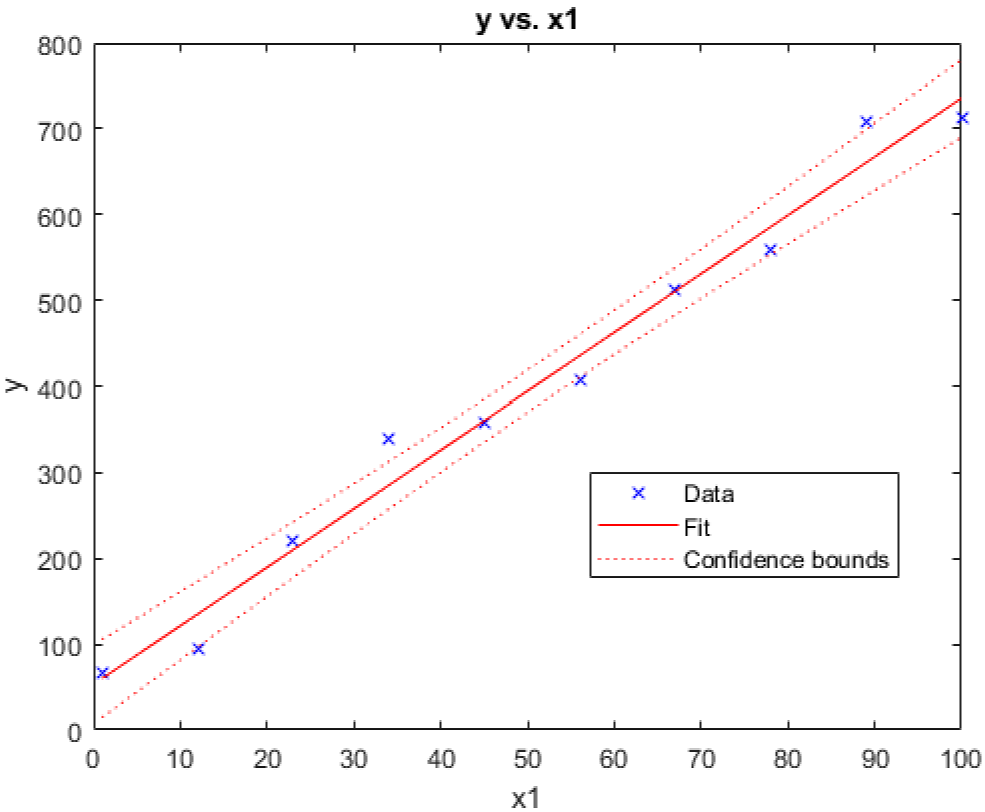


g)

```
% Alternatively, use the Matlab function fitlm to generate a linear
% regression model object. Plot the results by the use of plot(myModel) in a
 new figure.
myModel = fitlm(X,Y)

% By default, LinearModel assumes that you want to model the relationship
% as a straight line with an intercept term. The expression "y ~ 1 + x1"
% describes this model. Formally, this expression translates as "Y is
% modeled as a linear function which includes an intercept and a variable".
% Once again note that we are representing a model of the form Y = mX + B...
% The next block of text includes estimates for the coefficients, along
% with basic information regarding the reliability of those estimates.
% Finally, we have basic information about the goodness-of-fit including
% the R-square, the adjusted R-square and the Root Mean Squared Error.

figure(i+1)
plot(myModel)

% Notice that this simple command creates a plot with a wealth of information
 including
%
%    - A scatter plot of the original dataset
%    - A line showing our fit
```

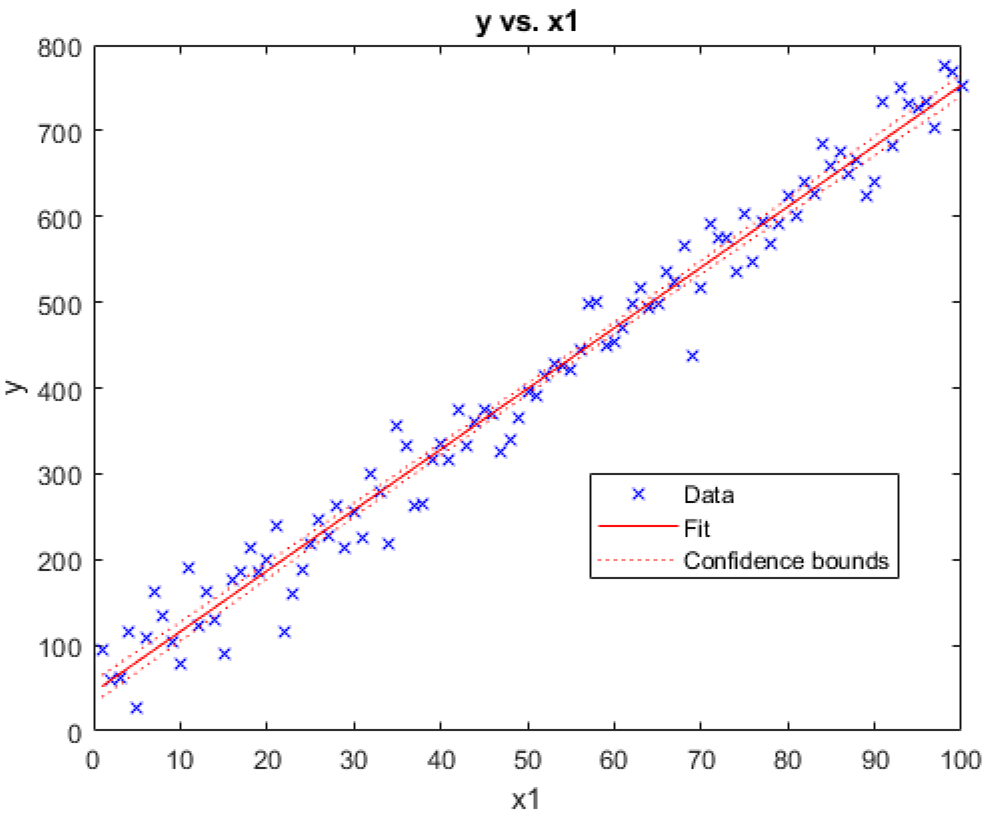
```
%    - Confidence intervals for the fit
%
% MATLAB has also automatically labelled our axes and added a legend.
```

```
myModel = 


Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate      SE        tStat       pValue  
                   ________    _______    _______    __________

    (Intercept)     53.322      20.016     2.6639       0.02863
    x1              6.8212     0.33601       20.3    3.6234e-08


Number of observations: 10, Error degrees of freedom: 8
Root Mean Squared Error: 33.6
R-squared: 0.981,  Adjusted R-Squared: 0.979
F-statistic vs. constant model: 412, p-value = 3.62e-08
```



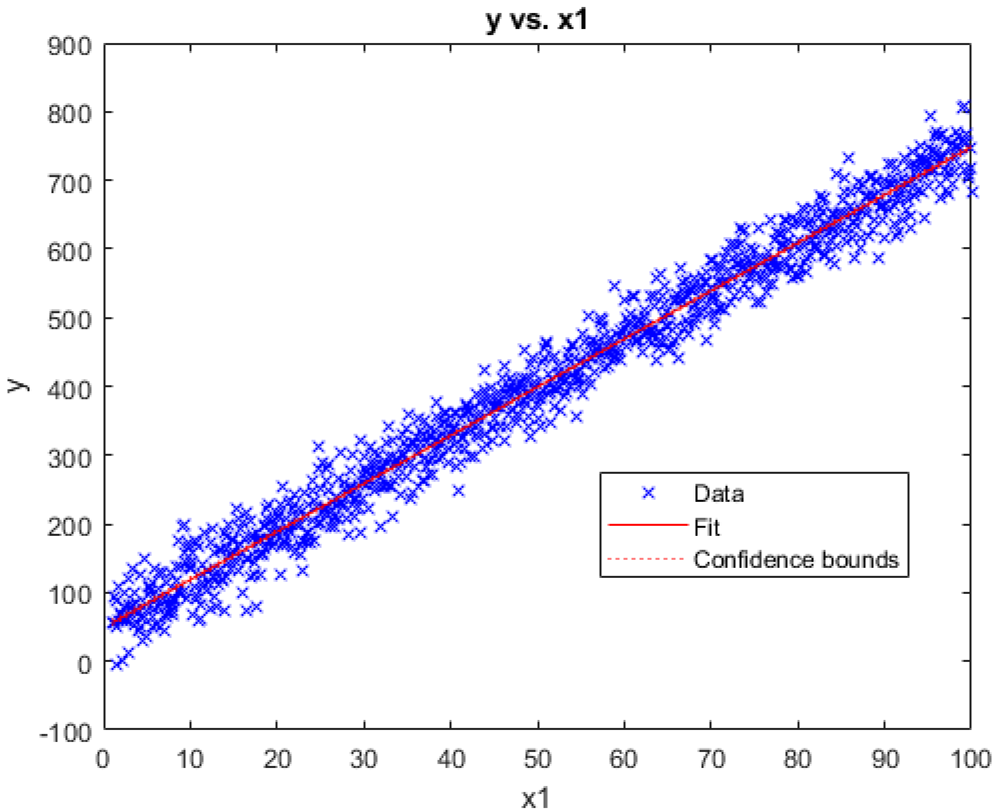

```
myModel = 
```

```
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate      SE        tStat       pValue   
                   ________    _______    ______    __________

    (Intercept)     45.194      6.5505    6.8993    5.1832e-10
    x1              7.0793     0.11261    62.863    5.2049e-81


Number of observations: 100, Error degrees of freedom: 98
Root Mean Squared Error: 32.5
R-squared: 0.976,  Adjusted R-Squared: 0.976
F-statistic vs. constant model: 3.95e+03, p-value = 5.2e-81
```




```
myModel = 


Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate      SE        tStat       pValue
```

```
                  _________    _________    _______    _____________

    (Intercept)      48.97       1.9671     24.894     8.5785e-107
    x1              6.9985     0.033893     206.49               0


Number of observations: 1000, Error degrees of freedom: 998
Root Mean Squared Error: 30.7
R-squared: 0.977,  Adjusted R-Squared: 0.977
F-statistic vs. constant model: 4.26e+04, p-value = 0
```



h)

```
% Now fit a polynomial regression model that prdicts y using x and x^2.
% Is ther evidence that the quadratic term improves the model fit? Explain
% your answer.

myModel2 = fitlm(X,Y,'poly2')


myModel2 = 


Linear regression model:
    y ~ 1 + x1 + x1^2
```

```
Estimated Coefficients:
                   Estimate        SE         tStat      pValue
                  __________    ________    _______    _________

    (Intercept)        47.57      29.139     1.6325      0.14659
    x1                7.1955      1.3443     5.3524    0.0010614
    x1^2          -0.0037059    0.012832    -0.2888       0.7811


Number of observations: 10, Error degrees of freedom: 7
Root Mean Squared Error: 35.7
R-squared: 0.981,  Adjusted R-Squared: 0.976
F-statistic vs. constant model: 182, p-value = 9.14e-07


myModel2 =


Linear regression model:
    y ~ 1 + x1 + x1^2

Estimated Coefficients:
                  Estimate         SE        tStat       pValue
                 _________    _________    ______    __________

    (Intercept)     58.148       9.8479    5.9047    5.2119e-08
    x1              6.3173      0.45007    14.036    4.3162e-25
    x1^2         0.0075447    0.0043174    1.7475      0.083712


Number of observations: 100, Error degrees of freedom: 97
Root Mean Squared Error: 32.2
R-squared: 0.977,  Adjusted R-Squared: 0.976
F-statistic vs. constant model: 2.02e+03, p-value = 9.14e-80


myModel2 =


Linear regression model:
    y ~ 1 + x1 + x1^2

Estimated Coefficients:
                   Estimate         SE         tStat         pValue
                  __________    _________    ________    ___________

    (Intercept)       48.046       3.0232      15.892     7.2975e-51
    x1                7.0524      0.13807      51.079    1.3781e-280
    x1^2          -0.0005337    0.0013252    -0.40274        0.68722


Number of observations: 1000, Error degrees of freedom: 997
Root Mean Squared Error: 30.7
R-squared: 0.977,  Adjusted R-Squared: 0.977
```

```
F-statistic vs. constant model: 2.13e+04, p-value = 0
```

i)

```
% Repeat a)-g) with different noise levels by sampling from N=(0,\sigma^2)
% with \sigma={30,60,100} to generate different \eps. Describe your results.

end
```

Published with MATLAB® R2022a