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(1000) prior to
% staring part a) to ensure consistent results.

clear all
close all
clc
rng(1000);                          % specify seed
```

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 = 50;
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);
```

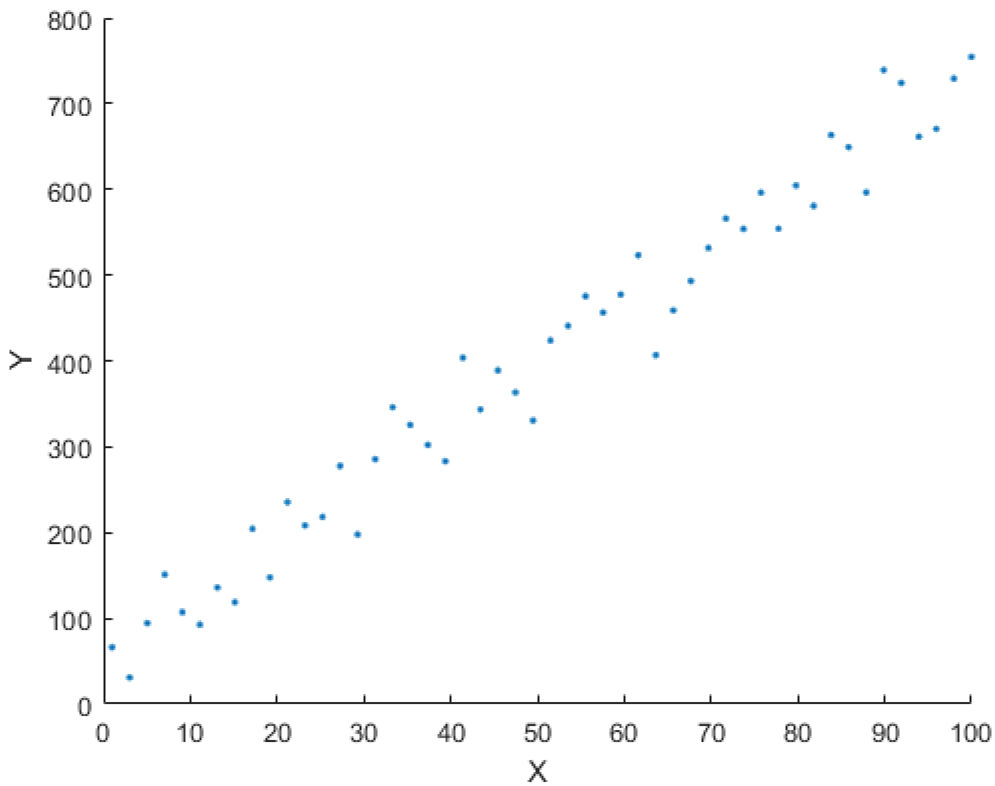
c)

```
% 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
scatter(X,Y, '.')
xlabel('X'), ylabel('Y')
```



e)

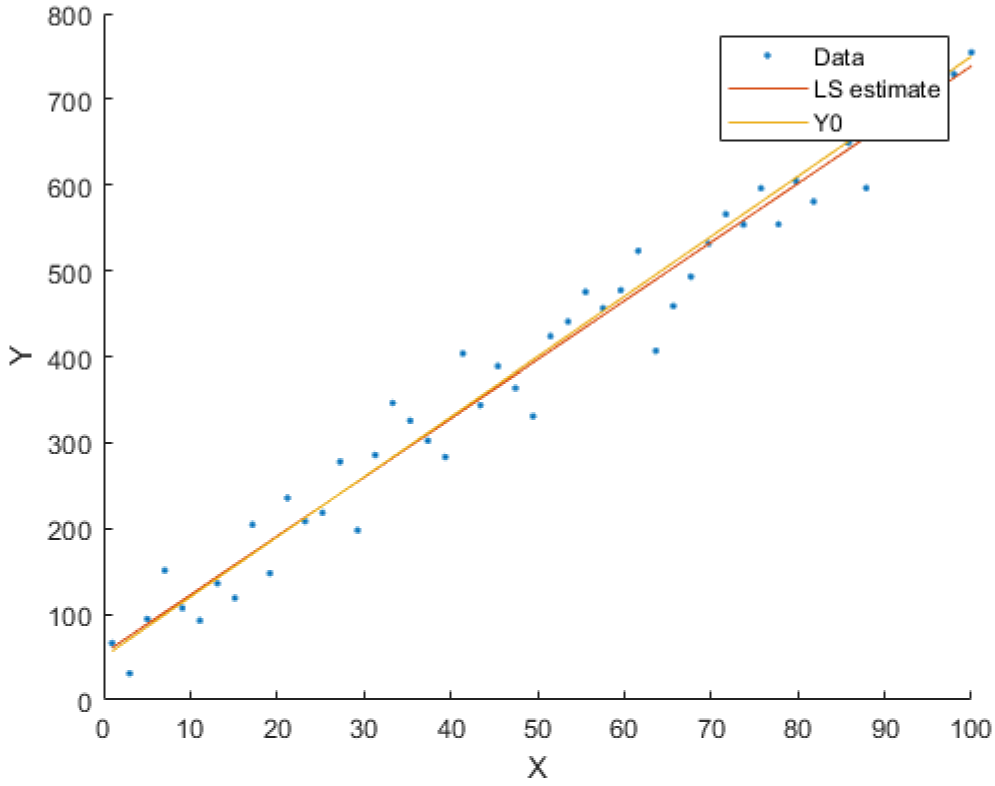
```
% Using the backslash operator, fit a least squares linear model to
% predict y using x. First, create the regression matrix using
% Phi=[ones(size(x)) x]. Comment on the model obtained. How do
% \hat{\beta}_0 and \hat{\beta}_1 compare to \beta_0 and \beta_1?
```

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

```
theta =

   54.1113
    6.8456
```

f)

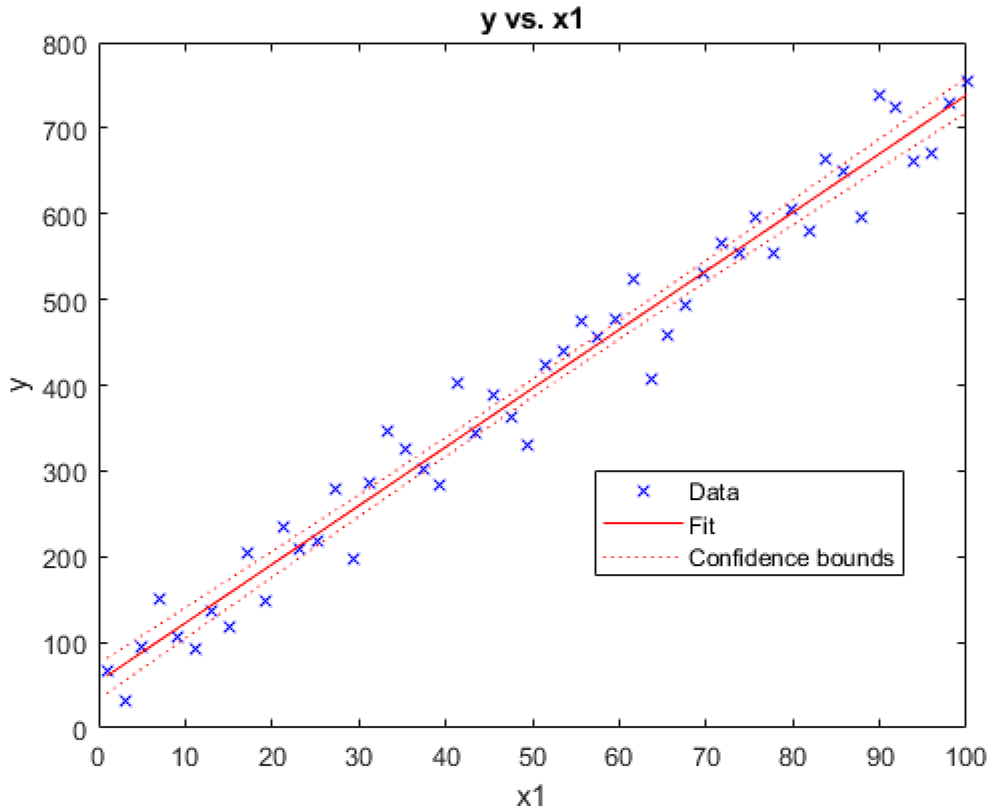
```
% Calculate the predictions \hat{Y} on the training data set. Display the
 least
% squares line on the scatterplot obtained in d) using hold all. Additonally,
 draw the
% population regression line on the plot.

Y_hat = Phi*theta;

hold all
plot(X,Y_hat)
plot(X,Y0)
legend('Data','LS estimate','Y0')
```



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
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)     54.111      10.335     5.2355    3.5989e-06
    x1              6.8456     0.17724     38.623    8.2597e-38


Number of observations: 50, Error degrees of freedom: 48
Root Mean Squared Error: 36.5
R-squared: 0.969,  Adjusted R-Squared: 0.968
F-statistic vs. constant model: 1.49e+03, p-value = 8.26e-38
```



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)        49.331       15.645      3.1532     0.0028124
    x1                 7.1296       0.7156      9.9631    3.6082e-13
    x1^2           -0.0028118    0.0068604    -0.40986       0.68377


Number of observations: 50, Error degrees of freedom: 47
Root Mean Squared Error: 36.9
```

```
R-squared: 0.969,  Adjusted R-Squared: 0.968
F-statistic vs. constant model: 733, p-value = 3.7e-36
```

Published with MATLAB® R2022a