ECE 8540 Analysis of Tracking Systems
Lab 1 - Model Fitting

Submitted By:
Shashi Shivaraju
C88650674

Clemson University
December 21, 2018

Abstract

This report explains the process involved in modeling of data using normal equations of least squares problems.

Contents

List of Figures

Chapter 1

Introduction

This report considers the problem of fitting a set of data points (x_i, y_i) to a suitable linear model. A model is a description of a shape or system using equations.A model is said to be linear,if it can be represented by a linear combinations of any number of specified functions, i.e. for the provided data set the relationship between the variables y and x is linear.Fitting can be described as the process of identifying values for the equation such that it best resembles the given data.

Linear model fitting has many practical applications. If the goal is prediction or error reduction,it can be used to fit a predictive model to an observed data set of values of the response and variables. Such developed models can be used to make a prediction of the system response for modified or additional variables. It can also be used to quantify the relationship between the response and the variables of the system.

Linear models can be fitted using different techniques such as the least squares approach, least absolute deviations regression or by minimizing a penalized version of the least squares cost function as in ridge regression and lasso regression.

This report describes the linear model fitting using the most commonly used least square approach. In this process, we use normal equations to formulate our model and analyze how well the data set fits our derived model. This technique is known as **linear regression**.

Chapter 2

Methods

Model fitting is a process which involves selecting a suitable model which best resembles the given data set. The process of selecting a suitable model can be based on trial and error method and it can be assisted by plotting the given dataset for visualization.

Let us first consider a generic linear model and describe the process of linear regression using least square approach and normal equations. Further we will utilize the results from the generic approach and try to fit three different datasets to their suitable models.

2.1 Generic Linear Model Fitting

Let the generic linear model be represented as

$$y = a_1 f_1(x) + a_2 f_2(x) + ... + a_M f_M(x) \tag{2.1}$$

where $a_1 ... a_M$ are the unknowns (M of them). The terms $f_1(x), f_2(x), ..., f_M(x)$ are called basis functions. The basis functions do not need to be linear but the unknowns must all be linear constants.

Equation (2.1) can be written as

$$y = \sum_{j=1}^{M} a_j f_j(x) \tag{2.2}$$

Given a set of points (x_i, y_i), we desire to find the general solution to equation (2.2) that best fits the data. Let the data be denoted as

$$(x_i, y_i) \quad i = 1...N \tag{2.3}$$

where N indicates the total number of data points.

Let us define the residual function e_i for each point as:

$$e_i = \left(y_i - \sum_{j=1}^{M} a_j f_j(x_i) \right) \tag{2.4}$$

We define the chi-squared error metric as the difference between the best fitting solution and the collective set of data:

$$\chi^2(a_1, a_2, ..., a_M) = \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{M} a_j f_j(x_i) \right)^2 \tag{2.5}$$

To find the best possible values for the unknowns $a_1...a_M$ we use differential equations to solve for the minimum chi-squared error. We take the partial derivatives of χ^2 with respect to $a_1...a_M$, set them equal to zero, and solve for $a_1...a_M$. There are M partial derivative equations.

In general form, the set of M equations can be written as:

$$\forall k = 1...M \quad \frac{\partial \chi^2}{\partial a_k} = \sum_{i=1}^{N} 2 \left(y_i - \sum_{j=1}^{M} a_j f_j(x_i) \right) (-f_k(x_i)) \tag{2.6}$$

To solve for the unknowns $a_1...a_M$ we set all these equations equal to zero:

$$\forall k = 1...M \quad \sum_{i=1}^{N} f_k(x_i) \left(y_i - \sum_{j=1}^{M} a_j f_j(x_i) \right) = 0 \tag{2.7}$$

Rearranging the terms and expanding the sums, we obtain

$$\forall k = 1...M \quad \sum_{i=1}^{N} f_k(x_i) y_i = \sum_{i=1}^{N} \sum_{j=1}^{M} f_k(x_i) f_j(x_i) a_j \tag{2.8}$$

We can use matrix notation to simplify the equations and define the following matrices:

$$A = \begin{bmatrix} f_1(x_1) & f_2(x_1) & \cdots & f_M(x_1) \\ f_1(x_2) & f_2(x_2) & \cdots & f_M(x_2) \\ \vdots & & & \vdots \\ f_1(x_N) & f_2(x_N) & \cdots & f_M(x_N) \end{bmatrix} \tag{2.9}$$

$$x = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_M \end{bmatrix} \tag{2.10}$$

$$b = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \tag{2.11}$$

Using these matrices, equation (2.8) can be rewritten in matrix form as

$$A^T b = A^T A x \tag{2.12}$$

We desire to solve for the unknowns in matrix x.

$$A^T A x = A^T b \tag{2.13}$$

Since $A^T A$ is by definition a square matrix and is therefore invertible. Multiplying both sides of equation (2.13) by this inverse gives

$$(A^T A)^{-1} A^T A x = (A^T A)^{-1} A^T b \tag{2.14}$$

Any matrix multiplied by its inverse yields the identity matrix, so that the left side of this equation simplifies:

$$x = (A^T A)^{-1} A^T b \tag{2.15}$$

Equation (2.15) is called the **solution to the normal equations.** By properly constructing the matrices A, x and b, the solution to any problem in the form of equation (2.1) can be found using equation (2.15).

2.2 Model Fitting for Dataset 1 and Dataset 2

We desire to fit suitable models to the provided Dataset 1 and Dataset 2.

Dataset 1 consists of five points as shown below:

$$Dataset1 = (x_i, y_i) = \{(5,1), (6,1), (7,2), (8,3), (9,5)\} \tag{2.16}$$

Dataset 2 consists of six points as shown below:

$$Dataset2 = (x_i, y_i) = \{(5,1), (6,1), (7,2), (8,3), (9,5), (8,14)\} \tag{2.17}$$

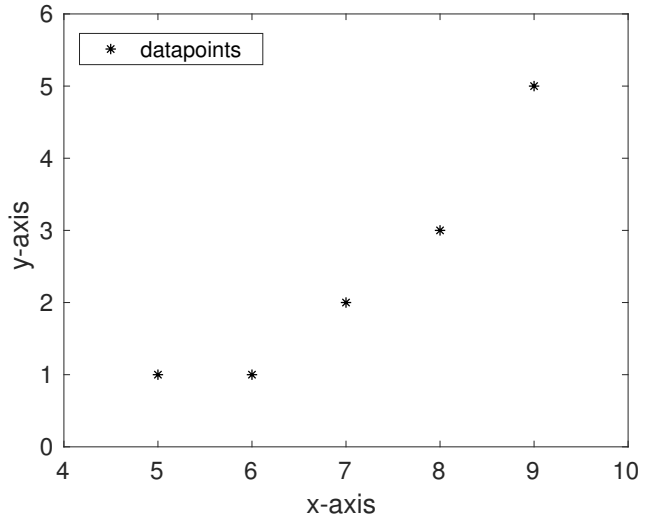

Figure 2.1: Plot of Dataset 1.

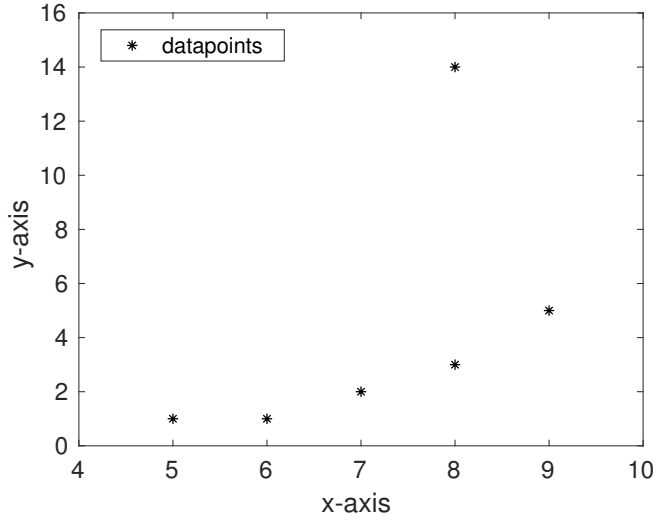

Figure 2.2: Plot of Dataset 2.

The plots of the Dataset 1 and Dataset 2 are shown in figures (2.1) and (2.2) respectively. On visualization of the plots of the Dataset 1 and Dataset 2, a straight line seems to a suitable model for fitting.

Let the 2D line be represented by the equation

$$y = ax + b \tag{2.18}$$

where 'a' is the slope of the line and 'b' is the y intercept.This equation is a special form of equation (2.1). If $a_1 = a$, $a_2 = b$, $f_1(x) = x$, and $f_2(x) = 1$, then equation (2.1) simplifies to (2.18). Here both 'a' and 'b' are the unknowns of the linear model.

Let the dataset be denoted as

$$(x_i, y_i) \quad i = 1...N \tag{2.19}$$

where N indicates the total number of data points.

We can now fit the 2D line equation (2.18) to the given data (2.19) by constructing the matrices according to equations (2.9),(2.10) and (2.11). The corresponding matrices are:

$$A = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix} \tag{2.20}$$

$$x = \begin{bmatrix} a \\ b \end{bmatrix} \tag{2.21}$$

$$b = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \tag{2.22}$$

For Dataset 1 represented by (2.16),the corresponding matrices according to equations (2.20),(2.21) and (2.22) are:

$$A = \begin{bmatrix} 5 & 1 \\ 6 & 1 \\ 7 & 1 \\ 8 & 1 \\ 9 & 1 \end{bmatrix} \tag{2.23}$$

$$x = \begin{bmatrix} a \\ b \end{bmatrix} \tag{2.24}$$

$$b = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 3 \\ 5 \end{bmatrix} \tag{2.25}$$

For Dataset 1,the model unknowns are found by solving equation 2.15 for the matrix x.

$$x = \begin{bmatrix} 1.0 \\ -4.6 \end{bmatrix} \tag{2.26}$$

where $a = 1.0$ and $b = -4.6$

Thus the equation of the model which can be fitted to Dataset 1 is

$$y = (1)x - 4.6 \tag{2.27}$$

For Dataset 2 represented by (2.17),the corresponding matrices according to equations (2.20),(2.21) and (2.22) are:

$$A = \begin{bmatrix} 5 & 1 \\ 6 & 1 \\ 7 & 1 \\ 8 & 1 \\ 9 & 1 \\ 8 & 1 \end{bmatrix} \tag{2.28}$$

$$x = \begin{bmatrix} a \\ b \end{bmatrix} \tag{2.29}$$

$$b = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 3 \\ 5 \\ 14 \end{bmatrix} \tag{2.30}$$

For Dataset 2,the model unknowns are found by solving equation 2.15 for the matrix x.

$$x = \begin{bmatrix} 1.8154 \\ -8.6769 \end{bmatrix} \tag{2.31}$$

where $a = 1.8154$ and $b = -8.6769$

Thus the equation of the model which can be fitted to Dataset 2 is

$$y = (1.8154)x - 8.6769 \tag{2.32}$$

2.3 Model Fitting for Dataset 3

We desire to fit a suitable model to the provided Dataset 3 obtained during 3398 meals eaten by 83 different individuals. The data in consideration is bites vs. kilocalories/bites for a given meal. The plot of bites vs. kilocalories/bites for all the meals is shown in figure (2.3).

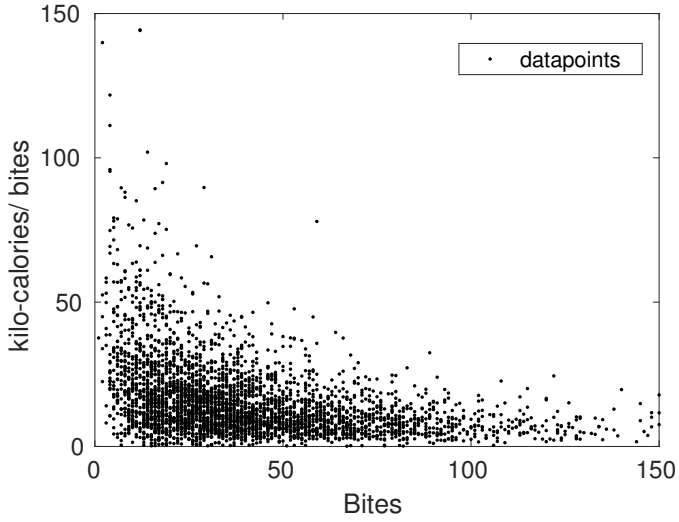

Figure 2.3: Plot of Dataset 3.

Since the data points overlap in figure (2.3),for better visualization let us plot the data points of every fifth meal as shown in figure (2.4).

The visualization of the plot revealed that a linear 2D line would not be a suitable model for this data.The data points of Dataset 3 along with their line fitted model is shown in figure(2.5). Next model considered for fitting was exponential ($y = ae^{bx}$) but it was observed that the model did not fit well at points where bite count was below 20.

Next we considered power function ($y = ax^b$) as a fitting model for the data. The equation of the power function is linearized using the rules of logarithms.

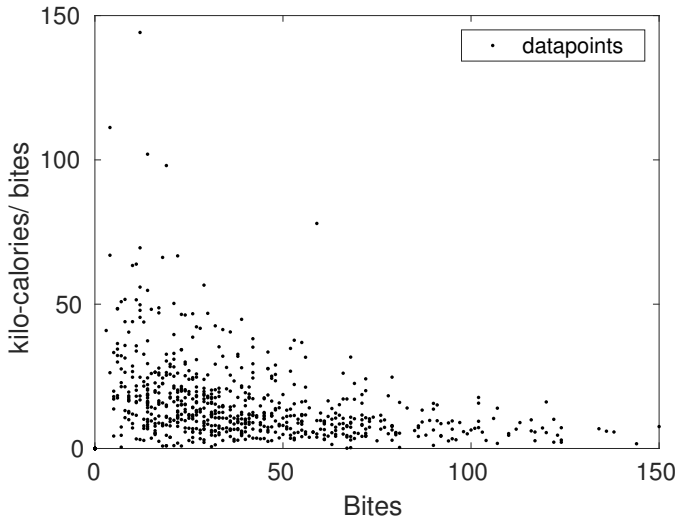

Figure 2.4: Plot of down sampled Dataset 3.

Let the power function be represented by the equation

$$y = ax^b \tag{2.33}$$

Using logarithms on both sides of the equation (2.33) to transform it in linear form,we get

$$ln(y) = bln(x) + ln(a) \tag{2.34}$$

This equation is a special form of equation (2.1). If $a_1 = b$, $a_2 = ln(a)$, $f_1(x) = ln(x)$, and $f_2(x) = 1$, then equation (2.1) simplifies to (2.34). Here both 'a' and 'b' are the unknowns of the linear model.

The corresponding matrices according to equations (2.9),(2.10) and (2.11) are:

$$A = \begin{bmatrix} ln(x_1) & 1 \\ ln(x_2) & 1 \\ ln(x_3) & 1 \\ \vdots & \vdots \\ ln(x_n) & 1 \end{bmatrix} \tag{2.35}$$

$$x = \begin{bmatrix} b \\ ln(a) \end{bmatrix} \tag{2.36}$$

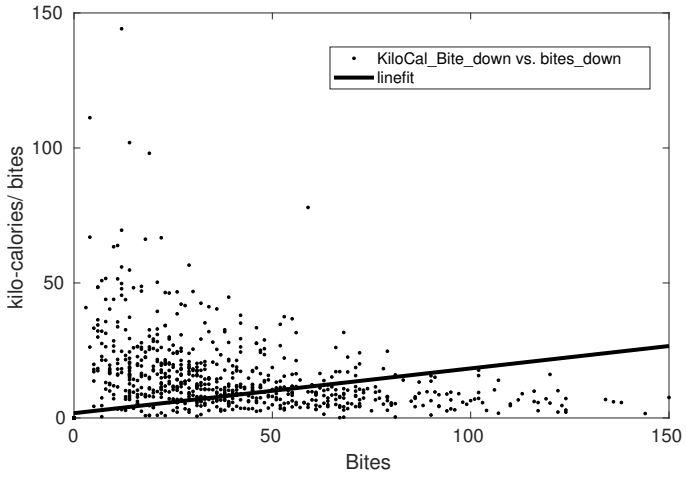

Figure 2.5: Plot of Dataset 3 with fitted with 2D line model.

$$b = \begin{bmatrix} ln(y_1) \\ ln(y_2) \\ ln(y_3) \\ \vdots \\ ln(y_n) \end{bmatrix} \tag{2.37}$$

For Dataset 3,the model unknowns are found by solving equation 2.15 for the matrix x.

$$x = \begin{bmatrix} -0.4601 \\ 4.0555 \end{bmatrix} \tag{2.38}$$

where $ln(a) = 4.0555$,$a = 57.7140$ and $b = -0.4601$

Thus the equation of model which can be fitted to Dataset 3 is

$$y = (57.7140)x^{-0.4601} \tag{2.39}$$

Please refer the Appendix for the implementation of section 2.2 and section 2.3 in Matlab.

Chapter 3

Results

3.1 Models for Dataset 1 and Dataset 2

The models fitted with Dataset 1 and Dataset 2 are given by equations (2.27) and (2.32) respectively. The data points of Dataset 1 and Dataset 2 along with their fitted models is shown in figure(3.1).

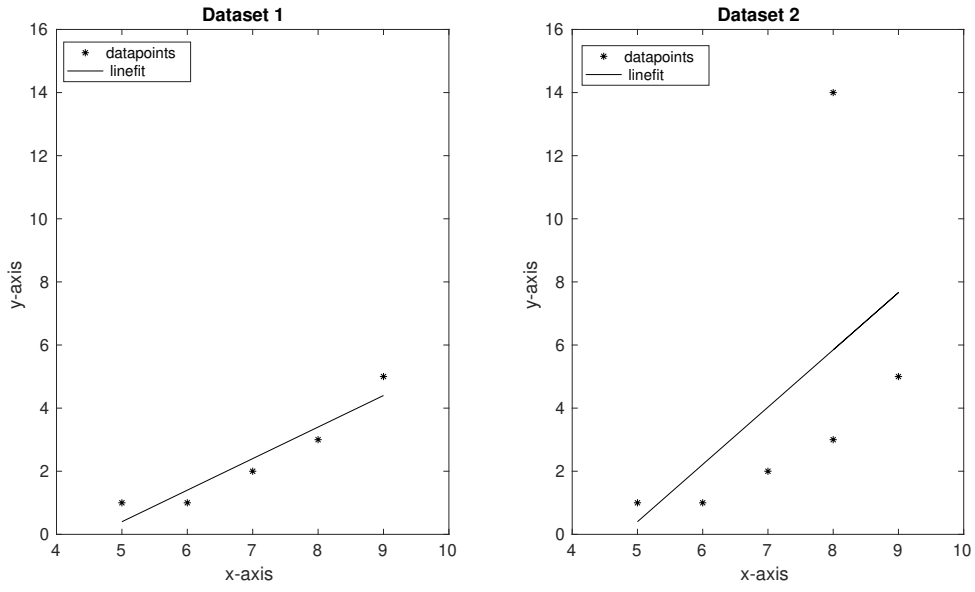

Figure 3.1: Plot of Dataset 1 and Dataset 2 with their fitted models.

From the plot shown in figure(3.1),we can observe that the slope of the fitted line for Dataset 2 has increased when compared to the fitted line for Dataset 1.This is due to the addition of data point $(8, 14)$ in Dataset 2,which is an outlier in the dataset.

3.2 Model for Dataset 3

The model fitted with Dataset 3 is given by equations (2.39). The data points of Dataset 3 along with its fitted model are shown in figure(3.2).

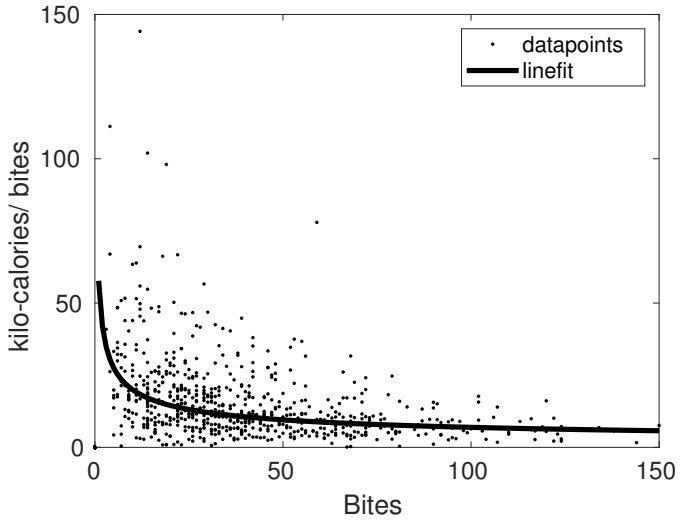

Figure 3.2: Plot of Dataset 3 with its fitted model.

From the plot shown in figure(3.2),we can observe that the power equation model fitted with Dataset 3 is a better fit when compared with the fit of other models like a 2D line with Dataset 3 as shown in figure(2.5).

Chapter 4

Conclusion

The report describes in detail the analytical process involved in fitting a dataset to a suitable linear model.

Model fitting for Dataset 1 and Dataset 2 showed us the effect of outlier data points on the fitted model.

Model fitting for Dataset 3 involved fitting a model to a very random set of data points. Even though there is no perfectly fitting model present for the dataset, we used our visualization and intuition to select different models and find the best fitting model among them. The power function was selected as a suitable model for Dataset 3 even though it is non-linear in nature. We used the rule of logarithms to convert the power equation into an equation of linear combinations with respect to the unknowns.Thus the result showcased that the technique of linear regression can be used to fit models whose basis functions are non-linear in nature but are having unknowns which are linear constants.

Appendix

Matlab Code

```
1
2  % FILE NAME      : Model_Fitting.m
3  %
4  % DESCRIPTION    : Code to fit a model to data
5  %
6  % PLATFORM       : Matlab
7  %
8  % DATE           NAME
9  % 29-Aug-2018    Shashi Shivaraju
10
11
12 clear; %clear all the varaibles
13 clc; %clear the screeen
14
15 %Part 1 of the lab
16 %data points
17 x1 = [5 6 7 8 9];
18 y1 = [1 1 2 3 5];
19
20 %declare  the matrices of the normal equation
21 A1 = [5 1;6 1;7 1;8 1;9 1];
22 b1 = [1;1;2;3;5];
23
24 X1 = (A1'*A1)\(A1'*b1);
25
26 %plot the data and the line
27 figure(1);
28 plot(x1,y1,'k*');
29 hold on;
30 y1 = X1(1,1)*x1+X1(2,1);
31 plot(x1,y1);
32 hold off;
33 axis([4 10 0 6]);
34 set(gca,"FontSize",14);
35 xlabel('x-axis');
36 ylabel('y-axis');
37 legend('datapoints','linefit')
38
39 %Part 2 of the lab
```

```
%data points
x2 = [5 6 7 8 9 8];
y2 = [1 1 2 3 5 14];

%declare the matrices of the normal equation
A2 = [5 1;6 1;7 1;8 1;9 1;8 1];
b2 = [1;1;2;3;5;14];
X2 = (A2'*A2)\(A2'*b2); %X = inv(A'*A)*A'*b

%plot the data and the line
figure(2);
plot(x2,y2,'k*');
axis([4 10 0 16]);
hold on;
y2 = X2(1,1)*x2+X2(2,1);
plot(x2,y2);
hold off;
set(gca,"FontSize",14,"YTick",(0:2:16));
xlabel('x-axis');
ylabel('y-axis');
legend('datapoints','linefit')


%Part 3 of the lab
%read the data from the file
T = readtable('83people-all-meals.txt');
Data = table2array(T);
%y-axis data
KiloCal_Bite = (Data(:,4)./Data(:,3));
%x-axis data
bites = Data(:,3);

bites_down = zeros(length(bites),1) ;
KiloCal_Bite_down = zeros(length(bites),1);

%declare the matrices of the normal equation
A3 = [log(bites) ones(length(bites),1)];
b3 = log(KiloCal_Bite);

%solve for the unknowns
X3 = (A3'*A3)\(A3'*b3); %X = inv(A'*A)*A'*b

a3 = exp(X3(2,1));
b3 = X3(1,1);

```

```
%plot the data and the line
figure(3)
plot(bites_down, KiloCal_Bite_down,'k.');
hold on;
x3 = 0:max(bites);
y3 = a3 * x3.^b3;
plot(x3,y3,'k','LineWidth',3);
hold off;
axis([0 150 0 150]);
set(gca,"FontSize",14);
xlabel('Bites');
ylabel('kilo-calories/ bites');
legend('datapoints','linefit')
```

References

1.Lecture notes of Dr.Adam Hoover

http://cecas.clemson.edu/ ahoover/ece854/lecture-notes/lecture-normeqs.pdf

2.TeX Live - TeX Users Group

https://www.tug.org/texlive/