ECE 8540 Analysis of Tracking Systems
Assignment 3

Vivek Koodli Udupa
C12768888

September - 25, 2018

1 Introduction

This report considers the problem of modeling a linear fit. Models are tools that are used to easily understand scientific ideas or to explain how data and events are related to each other. Scientists often use models to predict what might happen in a particular circumstance based on current available information. Weather prediction is a good example where a forecast is given based on past and current weather conditions.

The mathematical models can be linear or non-linear in nature. A line of the form $y = ax + b$ and an exponential function of the form $y = e^{ax}$ are examples for linear and non-linear models respectively. The difference between linear and non-linear models is not straight forward as it seems. It would be wrong to assume that linear equations only produce straight lines whereas non linear equations produce curves. Linear equations can produce curved lines too! So in that case how do we know whether a model is linear or non-linear?

Statistically speaking, a model is linear in terms of its parameters if all its terms are either constants or a parameter multiplied by an independent variable. $y = a_0 + a_1x_1 + ... + a_nx_x$ represents a linear model. Model of any other form is non-linear[1].

Through the course of the report, we will try to fit a model to the given data set by formulating appropriate matrices and finding the model parameters using normal equations. Initially a 2D line is fit to the given data set of five points. Later the effect of an additional point on the slope of the line will be observed. For the final part of the lab, we will fit an equation of the form $y = ax^b$ by converting the equation into linear in terms of its parameters and then computing the parameters.

2 Methods

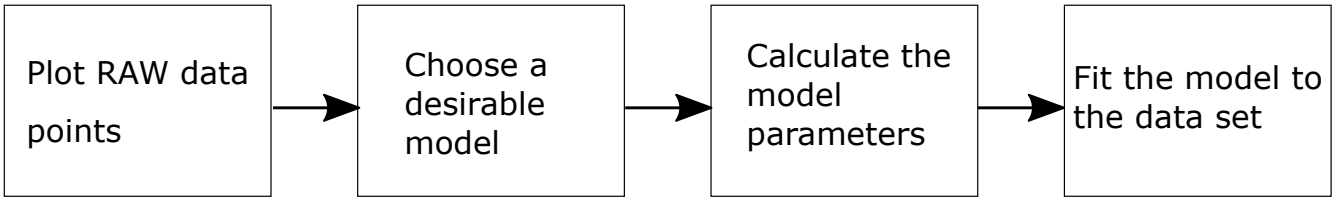

Figure 1: Flowchart for model fitting

This section walks us through the entire process of fitting a model to the given data set. The process can be divided into 4 stages, as shown in Figure 1. Initially, the raw data is plotted for visualization. This gives us a basic idea of what the model must look like. Based on this visual analysis, a desirable model that most likely fits is chosen. Once we have the equation representing the model, we must calculate the values of the constants and other included parameters. This is the basic idea of how to fit a model. We will go through these steps in greater detail as we model the fit for particular data set. We were given two data sets for this lab. Data set 1 is used for part A and part B of the lab and data set 2 is used for the part C of the lab.

2.1 General solution to normal equations

The equation of a line is a special case of a generalized fitting of any function consisting of linear combination of terms.

$$y = a_1 f_1(x) + a_2 f_2(x) + ... + a_M f_M(x) \tag{1}$$

where $a_1...a_M$ are linear unknowns. The terms $f_1(x)....f_M(x)$ are called basis functions. There is no need for the basis function to be linear. But the unknowns must all be linear constants.

Given a set of data points, say $(x_i, y_i) \quad i = 1...N$, we wish to find the solution to Equation 1 that best fits the data.

Now we define the residual e_i for each point as :

$$e_i = \left(y_i - \sum_{j=1}^{M} a_j f_j(x_i) \right) \tag{2}$$

To find the difference between the collective set of data and the best fitting solution, we define the chi-squared error metric:

$$\chi^2(a_1, a_2, ..., a_M) = \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{M} a_j f_j(x_i) \right)^2 \tag{3}$$

The best possible values for the unknowns can be found by using differential equations to solve for the minimum chi-squared error. We take the partial derivatives of χ^2 with respect to $a_1....a_M$, equate them to zero and solve for $a_1....a_M$.
In general form, the set of M equations can be written as :

$$\forall K = 1...M \quad \frac{\partial \chi^2}{\partial a_k} = \sum_{i=1}^{N} 2 \left(y_i - \sum_{j=1}^{M} a_j f_j(x_i) \right) (-f_k(x_i)) \tag{4}$$

To solve for the unknowns, we set all the equations to zero:

$$\forall K = 1...M \quad \frac{\partial \chi^2}{\partial a_k} = \sum_{i=1}^{N} (f_k(x_i)) \left(y_i - \sum_{j=1}^{M} a_j f_j(x_i) \right) = 0 \tag{5}$$

Rearranging the terms and expanding the sums, we obtain

$$\forall K = 1...M \quad \sum_{i=1}^{N} f_k(x_i) y_i = \sum i = 1^N \sum_{j=1}^{M} f_k(x_i) f_j(x_i) a_j \tag{6}$$

For the purpose of simplification, we will use matrix notation. We define the following matrices:

$$A = \begin{bmatrix} f_1(x_1) & f_2(x_1) & .... & f_M(x_1) \\ f_1(x_2) & f_2(x_2) & .... & f_M(x_2) \\ . & . & & . \\ . & . & & . \\ . & . & & . \\ f_1(x_N) & f_2(x_N) & .... & f_M(x_N) \end{bmatrix} \tag{7}$$

$$x = \begin{bmatrix} a_1 \\ a_2 \\ . \\ . \\ . \\ a_M \end{bmatrix} \tag{8}$$

$$b = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_M \end{bmatrix} \tag{9}$$

Using the matrices, we can rewrite equation 6 in matrix form as:

$$A^T b = A^T A x \tag{10}$$

Our goal is to solve for the unknowns in matrix x. Note that $A^T A$ is by definition a square matrix and is therefore invertible. Multiplying both sides of equation 10 by this inverse and then rearranging gives:

$$(A^T A)^{-1} A^T A x = (A^T A)^{-1} A^T b \tag{11}$$

Any matrix multiplied by its inverse yields the identity matrix. Thus the left side of the equation simplifies to:

$$x = (A^T A)^{-1} A^T b \tag{12}$$

Equation 12 is known as the solution to the normal equations. Given properly constructed matrices A, x and b, the solution to any problem in the form of equation 1 can be found using equation 12.

2.2 Part A and Part B

Data set A points: (5 , 1); (6 , 1); (7 , 2); (8 , 3); (9 , 5)
Data set B points: (5 , 1); (6 , 1); (7 , 2); (8 , 3) ;(9 , 5); (8 , 14)
Figure 2 and Figure 3 represent the plot of Data set A and Data set B respectively.

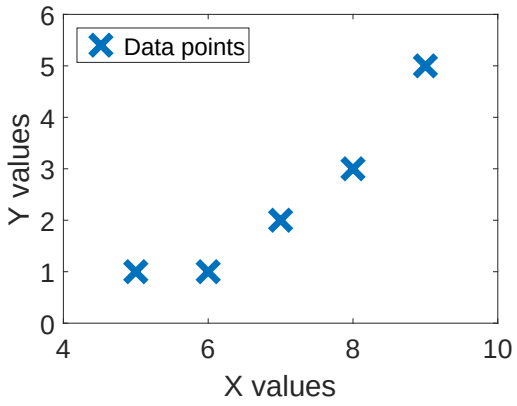

Figure 2: Plot of Raw data set A points

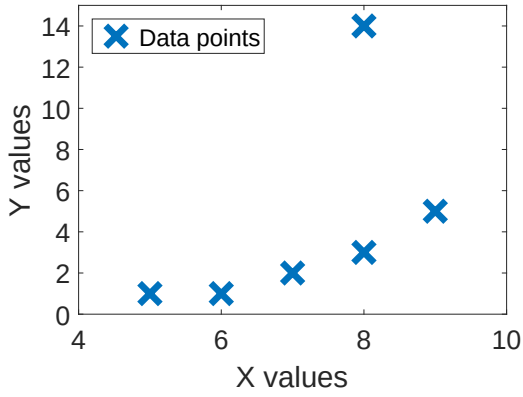

Figure 3: Plot of Raw data set B points

The first step in modeling a fit was to visualize the data. The next step is to find a model that might fit this visualization.

The y value of the data points seems to increase linearly as we traverse along the x-axis. Thus a 2D line of the form $y = ax + b$ might fit well. It is to be noted that the point (8,14) in Part B data set is considered as an outlier as its value is far off from other points.

Now we will formulate the matrices that were specified in equation 7, equation 8 and equation 9 .

$$A = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ . & . \\ . & . \\ x_N & 1 \end{bmatrix} \tag{13}$$

$$x = \begin{bmatrix} a \\ b \end{bmatrix} \tag{14}$$

$$A = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_N \end{bmatrix} \tag{15}$$

Now we shall fill in the matrices with appropriate values from data set A.

$$A = \begin{bmatrix} 5 & 1 \\ 6 & 1 \\ 7 & 1 \\ 8 & 1 \\ 9 & 1 \end{bmatrix} \tag{16}$$

$$b = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 3 \\ 5 \end{bmatrix} \tag{17}$$

$$x_u = \begin{bmatrix} a \\ b \end{bmatrix} \tag{18}$$

With the above matrices, the value of x_u can be calculated using equation 12. Using MATLAB for calculations, we get the following values for x_u :

$$x_u = \begin{bmatrix} 1.00 \\ -4.60 \end{bmatrix} \tag{19}$$

Therefore the 2D line fit for Data set A is given by

$$y_i = 1.00(x_i) - 4.60 \tag{20}$$

Following the similar method, the matrices constructed for Part B take the following form:

$$A = \begin{bmatrix} 5 & 1 \\ 6 & 1 \\ 7 & 1 \\ 8 & 1 \\ 9 & 1 \\ 8 & 14 \end{bmatrix} \tag{21}$$

$$b = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 3 \\ 5 \\ 14 \end{bmatrix} \tag{22}$$

$$x_u = \begin{bmatrix} a \\ b \end{bmatrix} \tag{23}$$

By using the same method used for Part A, we find the value of the unknowns for Part B of the lab as follows:

$$x_u = \begin{bmatrix} 1.8154 \\ -8.6769 \end{bmatrix} \tag{24}$$

Thus the line fit for Part B of the lab takes the form :

$$y_i = 1.8154(x_i) - 8.6769 \tag{25}$$

2.3 Part C

For the third part of the lab, the data set named "83people-all-meals.txt" is given. The data are for 3,398 meals eaten by 83 different people. The first column is the participant ID, the second column is the meal ID. The third column is the number of bites taken in the meal, and the fourth column is the number of kilo calories consumed.

In order to visualize the data, we have to plot the data. We will be plotting column 3 (i.e., number of bites taken in a meal) against the dividend of column 4 and column 3 (i.e., kilo-calories consumed per bite). This will give us the statistic of number of kilo-calories consumed per bite. Figure 4 is the raw data plot of bites taken in any meal against the kilo-calories consumed per bite, without any kind of data pre-processing. On general observation, we can say that bites taken has sort of an exponential relationship with the calories consumed. That is, the calorie per bite is drops exponentially as the number of bites taken increases. It has to be kept in mind that the total number of calories consumed per meal is not effected by the number of bites taken and this

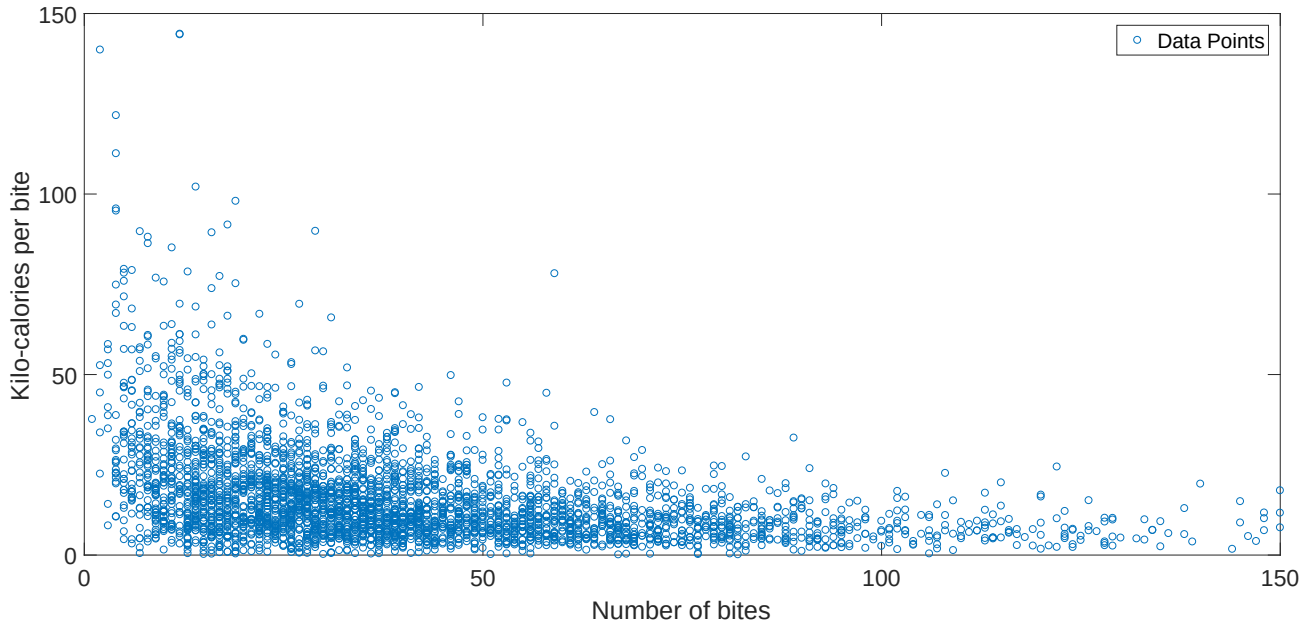

Figure 4: Unprocessed Part 3 Data set

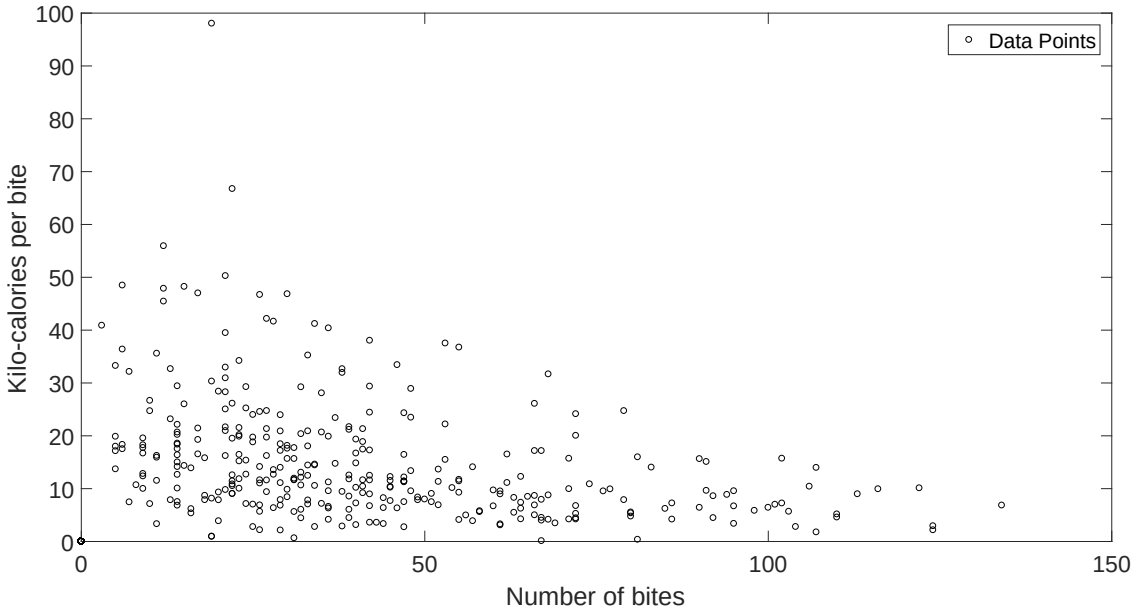

Figure 5: Plot of pre-processed Part 3 data

is just a statistic of the calories consumed per bite for that meal.

The raw data shown in Figure 4 has a lot of redundancy. Multiple data points overlap and the visualization of the data is not efficient. Thus for the purpose of better visualization, we will purge few of these redundant data points. Figure 5 shows the same statistics as before, Kilo calories consumed per bite against the number of bites taken, but instead of plotting the entire data set every tenth data entry is considered. This removes a lot of overlapping and the data interpretation becomes easier.

First we will try to fit a 2D line to this model and see whether it is a good fit. But as the observation revealed, the relationship in this case is more of an exponential than linear. So we will also consider a power fit model of the form $y = ax^b$ and see if it fits better than linear 2D line.

Now using the same procedure as explained in section 2.1 the value of the line parameters for this case was found to be [-0.1771, 23.4417]. Thus the linear line fit will be of the form

$$y = -1.771(x) + 23.4417 \tag{26}$$

Now we will try to fit a power model of the form:

$$y = ax^b \tag{27}$$

Equation 27 has two unknowns, 'a' and 'b' and the equation is not linear in terms of its parameters. In order to convert the model to a linear form, we will take log on both sides

$$log(y) = log(ax^b) \tag{28}$$

we know that $log(m * n) = log(m) + log(n)$ and also $log(m^n) = n * log(m)$. Thus on using these two logarithmic properties, equation 28 simplifies to:

$$log(y) = log(a) + log(x^b) \tag{29}$$

$$log(y) = log(a) + blog(x) \tag{30}$$

Equation 30 can be re-weitten as :

$$v = k + bu \tag{31}$$

where,

$$v = log(y) \quad k = log(a) \quad u = log(x) \tag{32}$$

We will formulate the matrices as specified in equation 7, equation 8 and equation 9 :

$$A = \begin{bmatrix} log(x_1) & 1 \\ . & . \\ . & . \\ log(x_N) & 1 \end{bmatrix} \tag{33}$$

$$x = \begin{bmatrix} a \\ b \end{bmatrix} \tag{34}$$

$$b = \begin{bmatrix} log(y_1) \\ . \\ . \\ log(y_N) \end{bmatrix} \tag{35}$$

Now using equation 12, we get the values of the unknowns as a = 57.7128 and b = -0.4601. Thus the model will be of the form

$$y = (57.7128)x^{-0.4601} \tag{36}$$

3 Results

In this section, we will go through the plots of the linear model fit and analyze them.

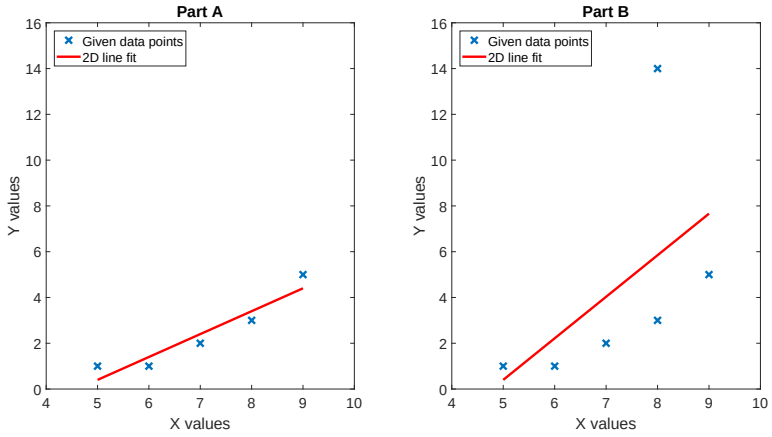

Figure 6: 2D line fit for part A and Part B of lab

Figure 6 shows the 2D line fit, whose parameters are given by equation 20 and equation 25. Part A had 5 data points and the line fit model. Part B clearly shows the effect of the outlier point (8, 14) on the linear fitting model. The slope of the Part B model almost doubled due to the high y-value of the extra point. The slope for Part A was 1.00 which increased to 1.8154 when the extra data point was included.

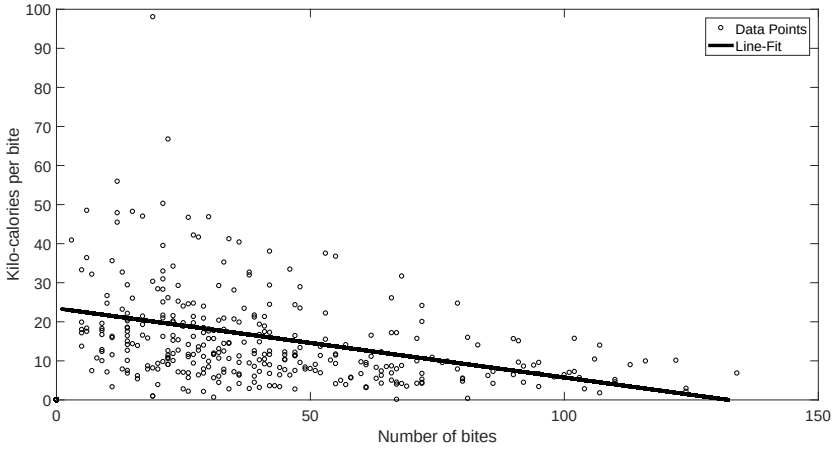

Figure 7: 2D line fit for part C of lab

Figure 7 shows the line fit for the data set 2 used for part C of this lab assignment. The model is of the form $y = -0.1771(x) + 23.4417$. This model is represented in the equation 26

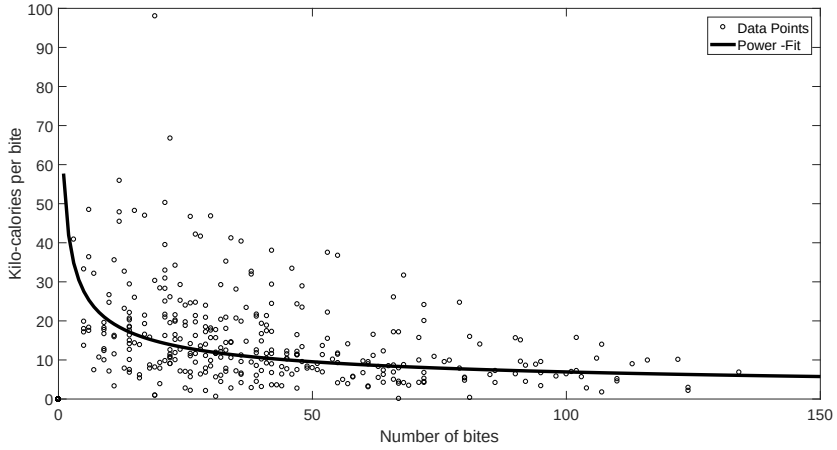

Figure 8: Power fit for part C of lab

Figure 8 represents the power fit for the data set 2 used for part C of this lab assignment. The model shown in equation 36. This power model is defined as $y = (57.7128)x^{(-0.4601)}$

4 Conclusion

In this lab, we derive the generalized solution, represented by equation 12, that can be used to fit any linear model that is of the form defined in equation 1 .

4.1 Part A and Part B

For this section of the lab assignment, we plotted the given data set A, which is shown in Figure 2 and Figure 3. Our goal was to fit a 2D line and observe the effects of the extra point (8,14) on the line fit.
Figure 6 shows the line fit. The graph in the left represents the part A fit and the right graph represents the part B. They have been arranged next to each other for the purpose of comparison. The line fir of part A gives a good approximation of the given data points. The addition of the outlier point, which has a high y-axis value, pulls the fitting line higher towards that point. The slope, which was 1.00 approximately doubles to 1.8154. This is the effect of the outlier point on the 2D line fit.

4.2 Part C

Our goal for this section was to find a model for the data set represented by Figure 5. We choose to try out a linear 2D line fit shown in Figure 7 and a power fit shown is Figure 8. The line fit was not a good approximation as the error around 0 - 50 bite range is high. The power model seems to be a much better fit compared to the 2D line.
Only two models were considered in this paper. It would be interesting to try out other models which could give better approximations than the ones discussed.

References

[1] http://statisticsbyjim.com/regression/difference-between-linear-nonlinear-regression-m

[2] http://cecas.clemson.edu/~ahoover/ece854/lecture-notes/lecture-normeqs.pdf

[3] https://www.latex-tutorial.com/

[4] https://www.sharelatex.com/blog/latex-guides/beginners-tutorial.html

5 Appendix

5.1 MATLAB Code

```
close all
clear

%Dataset one
px1 = [5, 6, 7, 8, 9];
py1 = [1, 1, 2, 3, 5];

%M = 2, N = length(x)

A1 = [px1' , ones(length(px1),1)]; %A Matrix
b1 = py1'; %Leftovers

x1 = inv(A1' * A1) * A1' * b1 %Calculating the unknowns

Y1 = x1(1) * px1 + x1(2); %Final Y1


%Plotting the graph
figure(2)
ax1 = subplot(1,2,1)
plot ( px1, py1, 'x', 'markers',12,'LineWidth',3 ); %Given dataset
   plot
hold on
plot ( px1 , Y1 , 'r-','LineWidth',3);%2d Line fit plot
hold off;
title('Part A');
ylabel('Y values');
xlabel('X values');
legend('Given data points', '2D line fit','Location','northwest');
```

```
29 set(gca,'FontSize',18);
30
31 %Dataset 2
32 px2 = [5, 6, 7, 8, 8, 9];
33 py2 = [1, 1, 2, 3, 14, 5];
34
35 A2 = [px2' , ones(length(px2),1)];
36 b2 = py2';
37
38 x2 = inv(A2' * A2) * A2' * b2
39
40 Y2 = x2(1) * px2 + x2(2);
41
42 %Plotting the Graph
43 ax2 = subplot(1,2,2)
44 plot ( px2, py2, 'x','markers',12,'LineWidth',3 ); %Given dataset plot
45 hold on
46 plot ( px2 , Y2 , 'r-','LineWidth',3); %2d Line fit plot
47 hold off;
48 title('Part B');
49 ylabel('Y values');
50 xlabel('X values');
51 legend('Given data points', '2D line fit','Location','northwest');
52 set(gca,'FontSize',18);
53 axis([ax1,ax2], [4 10 0 16])
54 %Plotting Raw Data Points
55 figure(99)
56 plot(px1, py1, 'X','markers',22,'LineWidth',5)
57 ylabel('Y values');
58 xlabel('X values');
59 %title('Part A Dataset');
60 legend('Data points','location','NorthWest');
61 set(gca,'FontSize',20,'YTick',[0:1:15]);
62 axis([4 10 0 6])
63
64 figure(100)
65 plot(px2, py2, 'X','markers',22,'LineWidth',5)
66 ylabel('Y values');
67 xlabel('X values');
68 %title('Part B Dataset');
69 set(gca,'FontSize',20,'YTick',[0:2:15]);
70 legend('Data points','location','NorthWest');
71 axis([4 10 0 15])
72
```

```
73
74 %Part Three ( Power Function)
75
76 %Read the dataset from file into a table
77 Tble = readtable('83people-all-meals.txt');
78
79 %Convert the table into an array
80 A1 = table2array(Tble);
81
82 %Extract the 3rd column(i.e Bites taken)
83 Bites = A1(:,3);
84
85 %Calculate Kilocalories per bite
86 CalBite = A1(:,4) ./ A1(:,3) ;
87
88 %extract every 7th data
89 Bites_reduced = zeros(length(A1(:,1)),1);
90 CalBite_reduced = zeros(length(A1(:,1)),1);
91 j = 1;
92 for i = 1:10:(length(Bites))
93     Bites_reduced(j,1) = Bites(i,1);
94     CalBite_reduced(j,1) = CalBite(i,1);
95     j = j+ 1;
96 end
97
98 %PArt 3 Raw data plot
99 figure(1)
100 plot(Bites_reduced,CalBite_reduced,'ko','markers',5);
101 xlabel("Number of bites");
102 ylabel("Kilo-calories per bite")
103 legend("Data Points");
104 set(gca,'FontSize',20);
105 axis([0 150 0 100])
106
107 %setting up the Matrices
108
109 % y3 = ax^b;
110 % V = K + bu3
111 % V = log(y3)
112 % K = log(a)
113 % u3 = log(x)
114
115 u3 = log(Bites);
116 v = log(CalBite);
```

```
117
118 A3 = [u3 ones(length(Bites),1)];
119 b3 = [v];
120
121 x3 = inv(A3' * A3) * A3' * b3;
122
123 a3 = exp(x3(2,1));
124 b3 = x3(1,1);
125
126 X3 = 0:max(Bites);
127 Y3 = a3 * X3.^b3;
128
129 % Plotting the graph
130 figure(3)
131 plot(Bites_reduced,CalBite_reduced, 'ko','markersize',5,'linewidth',1)
    ;
132 hold on;
133 plot(X3,Y3,'k-','LineWidth',4);
134 hold off;
135 xlabel("Number of bites");
136 ylabel("Kilo-calories per bite");
137 legend('Data Points','Power-Fit');
138 set(gca,'FontSize',18);
139 axis([0 150 0 100])
140
141 %Part 3 - Polynomial
142
143 A4 = [Bites ones(length(Bites),1)];
144 b4 = [CalBite];
145 x4 = inv(A4' * A4) * A4' * b4;
146
147 X4 = 0:max(Bites);
148 Y4 = x4(1,1) .* Bites + x4(2,1);
149
150 % Plotting the graph
151 figure(4)
152 plot(Bites_reduced,CalBite_reduced, 'ko','markersize',5,'linewidth',1)
    ;
153 hold on;
154 plot(Bites,Y4,'k-','LineWidth',4);
155 hold off;
156 xlabel("bites");
157 xlabel("Number of bites");
158 ylabel("Kilo-calories per bite");
```

```
159 legend('Data Points','Line-Fit');
160 set(gca,'FontSize',18);
161 axis([0 150 0 100])
```