Clemson University

Analysis of Tracking Systems

ECE 8540

Lab-3 Report

Submitted by:

Priyanshu Rawat

September 30, 2022

Contents

1 Introduction

In today's modern world, a plethora of data is generated every second. Whether in engineering or non-engineering applications this data can be used to gather invaluable pieces of information that might help engineers build better products, scientist invent newer technologies, and financial analysts predict stock prices. However, to find a meaning in this humongous amount of data experts have been using some techniques to find mathematical models which correlates the data to the problem of interest.

1.1 What is a model?

In the world of physics, "models" might refer to things in the real world that convey the essence of a thing being modeled. For instance, a model of a building implies a scaled-down structure of the building. Similarly, in biology a model of a cell would refer to a scaled-up model of things, like a model of the cell which would show its different parts.

In mathematics, or statistics in particular a model would refer to a similarly condensed description, but for data rather than a physical structure. A statistical model would generally be simpler than the data being described.

The basic structure of a statistical model is:

$$data = model + error$$

The above expression expresses the idea that any data can be broken into two portions: one portion that is described by a mathematical model, which expresses the values that we expect the data to take given our knowledge, and another portion that we refer to as the error that reflects the difference between the model's predictions and the observed data.

Ideally, we would like to use our model to predict the value of the data for any given observation, then the above equation would look like this:

$$\hat{data_i} = model_i$$

The "hat" refers to the prediction and overall the expression conveys that the predicted value of the data for observation is equal to the value of the model for that observation. Once we have a prediction from the model, we can then compute the error:

$$\hat{error_i} = model_i \text{ - } \hat{data_i}$$

That is, the error for any observation within a data is the difference between its observed value and the predicted value from the model.

1.2 Fitting models to data

Model fitting is a mathematical technique to measure how well a model generalizes to the data. A model that is well fitted is capable of producing accurate outcomes. Not every model can fit a data perfectly, a model matching the data too closely is referred to be over-fitted, whereas a model that doesn't match the data closely is referred as under-fitted.

1.3 Problem Statement

Develoving code for each of the problems to fit a model to data:

1. Fit a 2D line to the five x,y data points: (5,1);(6,1);(7,2);(8,3);(9,5)
   (a) Assess the results after including the data point (8,14)
2. Fit a model to the data in the file "83people-all-meals.txt". 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 kilocalories consumed. Plot the data using the third column (bites) and the dividend of the fourth and third columns (kilocalories per bite). Assess what type of model looks appropriate for fitting to this data?

2 Methods

The given problem deals with line fitting and curve fitting problem. Both of these techniques are useful in analyzing the trend of the given data. For the 1st part of the problem statement we need to find a equation of a line which best represents the nature of data distribution.

2.1 Line fitting

Line fitting is the process of finding a 2D straight line that has the best fit to a series of data points such line is also known as a *a line of best fit*

A *line of best fit* refers to a line that minimizes the distance between the line itself and the observations in the data set. The line of best fit would accurately represent the trend or correlation between the independent and dependent variable(s).

2.1.1 Methodology

To get a line of best fit for N number of data points, the first step is to calculate the residual which is given by the equation below, (where x_i and y_i are the data points)

$$e_i = y_i - ax_i - b$$

The next step involves adding all the residuals for every point in the given data set. After which, we take the square of this sum; just to simplify the following math by making this a convex quadratic function with a minimum. Lastly we take the derivative of the resulting equation with respect to a and b and equate it to 0. Now, we have two equation with two unknowns a and b. The final two equations after rearranging would look something like below -

$$\sum_{i=1}^{N} x_i y_i \text{ - } a \sum_{i=1}^{N} x_i^2 \text{ - } b \sum_{i=1}^{N} x_i = 0$$

$$\sum_{i=1}^{N} y_i \text{ - } a \sum_{i=1}^{N} x_i \text{ - } b \sum_{i=1}^{N} 1 = 0$$

2.1.2 Solution

For the given line fitting problem, the plot for the raw data points before and after including the data point $(8, 14)$ is depicted in figure below.

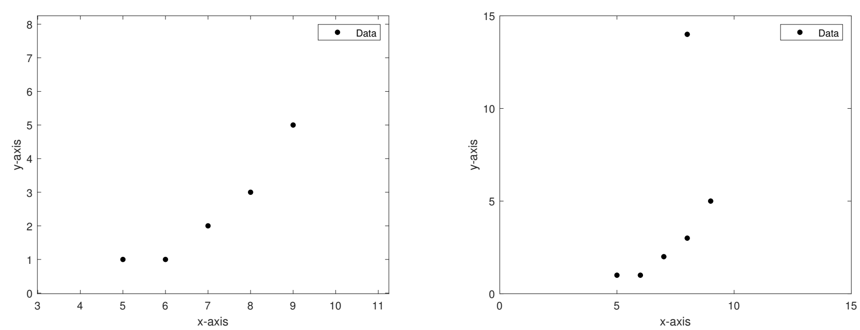


Figure 1: Raw data points without and with the outlier

The equation of lines representing best fit for raw points before and after adding the data point (8,14) respectively is obtained from solving the normal equations to find the two linear unknowns for both cases. The lines of best fit for both cases are depicted below in the plots -

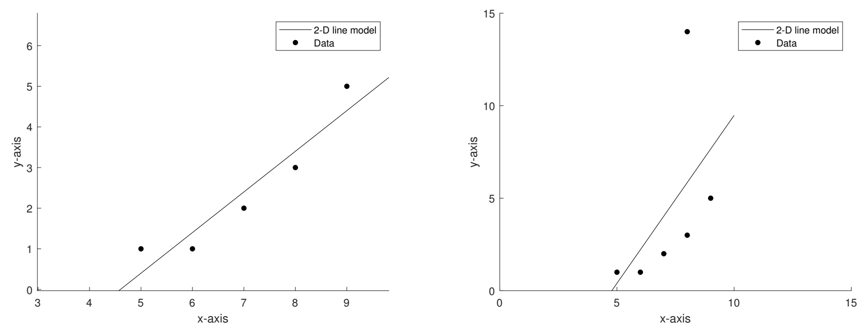


Figure 2: Line of best fit for raw data points without and with the outlier

2.2 Curve fitting

Curve fitting is the process of establishing a mathematical relationship or a best fit curve to a given set of data points. It mainly deals with finding a specific non-linear model that provides the best fit to the specific curves in the given data set. Curved relationships between variables are not as straightforward to fit and interpret as linear relationships in line fitting.

2.2.1 Methodology

The technique used for line fitting can be generalized to the fitting of any function consisting of linear combinations of terms. Such a function can be defined as -

$$y = a_1 f_1(x) + a_2 f_2(x) + ... + a_M f_M(x)$$

In the equation a_1 ... a_M are the unknowns and the terms $f_1(x)$ ... $f_M(x)$ are the basis functions. Here, all the unknowns must be linear constants, and the basis functions can be anything. Now, similar to line fitting, we evaluate the residual for each data point and square it. Now, to find the minimum, hence the solution for this newly obtained convex function, we need to equate the derivates of this function taken with respect to all unknowns from a_1 ... a_M.

After simplifying the equations above, we can rewrite the equations using the following matrices -

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

$$x = \begin{bmatrix} a1 \\ a2 \\ : \\ : \\ a_M \end{bmatrix}$$

$$b = \begin{bmatrix} y1 \\ y2 \\ : \\ : \\ y_N \end{bmatrix}$$

After writing the equations using the matrices above, we can rearrange the matrices to find the unknown matrix x, which is given by -

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

2.2.2 Solution

For the given curve-fitting problem, the plots below show the data points and a cropped view depicting the trend of the data distribution. Note that the data is plotted after estimating the kilocalories consumed per bite.

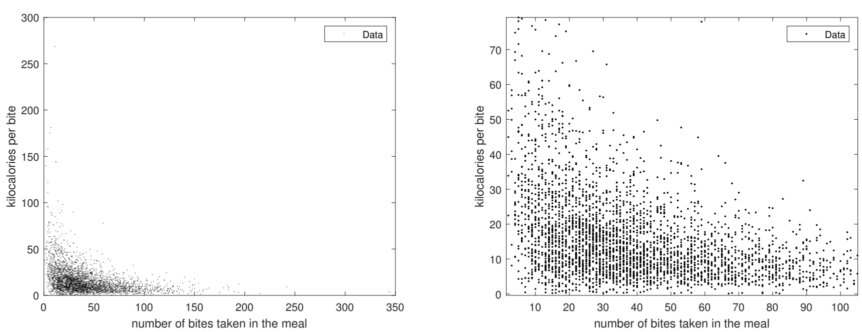


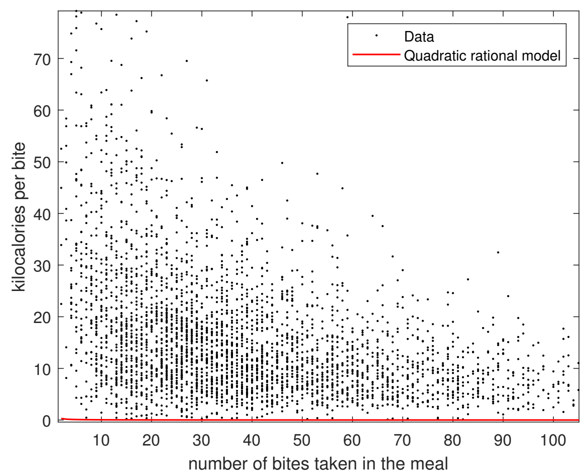
Figure 3: Plots depicting the data points and the trend in a cropped view

Fitting an quadratic rational model with equation: $y = a_1(x/(x^2 - 1))$

After solving for the one unknown basis function, we get the following model:

$$y = 0.44(x/(x^2 - 1))$$

The model is depicted in the figure below -

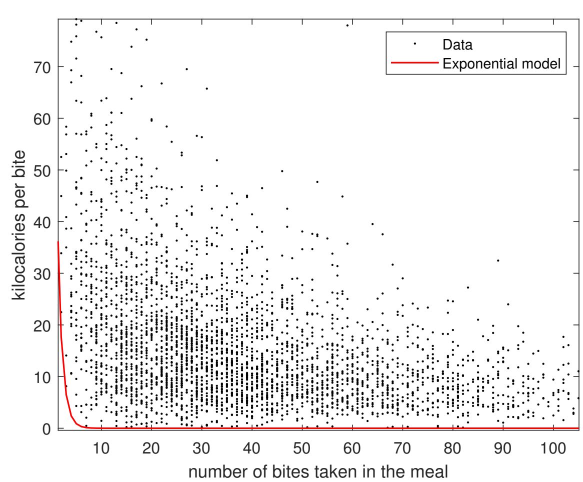


Fitting an exponential model with equation: $y = a_1 e^{-x}$

After solving for the one unknown basis function, we get the following model:

$$y = 130.35e^{-x}$$

The model is depicted in the figure below -

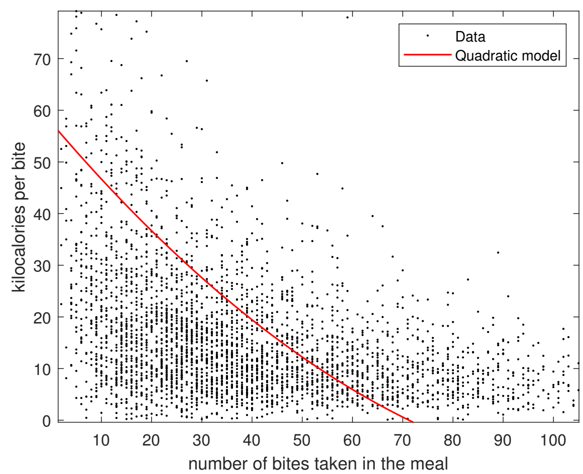


Fitting an quadratic model with equation: $y = a_1x^2 + a_2x + a_3$

After solving for the three unknown basis functions, we get the following model:

$$y = 0.004x^2 - 1.14x + 57.63$$

The model is depicted in the figure below -

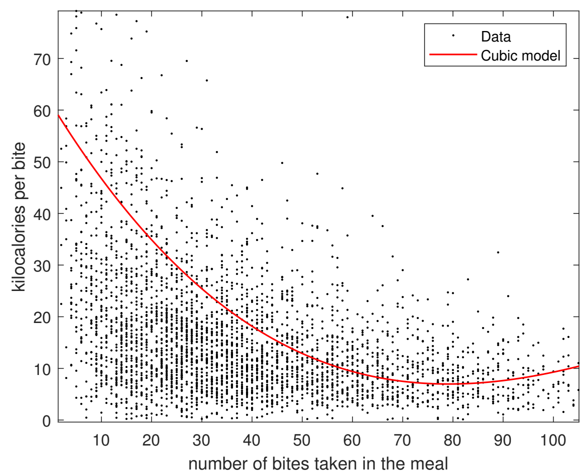


Fitting an cubic model with equation: $y = a_1x^3 + a_2x^2 + a_3x + a_4$

After solving for the four unknown basis functions, we get the following model:

$$y = -0.00003x^3 + 0.014x^2 - 1.58x + 61.22$$

The model is depicted in the figure below -



3 Results and discussion

The problem of line fitting and curve fitting had some interesting points to discuss which are summarized here.

3.1 Line fitting problem

In the line fitting problem, it can be seen that the line of best fit depicts the nature of data accurately when the new data point (8,14) was not included. However, when the data point (8,14) was included, the line of best fit moved towards it; thus showing a distorted or inaccurate spread of the data points.

This was because the new data point (8,14) is an outlier of the set; hence model obtained by solving the normal equation is shifted away from other points toward this new data point leading to a misleading model. This is mainly because the outlier (8,14) has a larger residual value than other data points within the set, which is then augmented when calculating the sum of the squares.

3.2 Curve fitting problem

In the curve fitting problem, it can be seen that two of the models, namely - the *quadratic rational* and *exponential* model, inaccurately fit the data without showing the true meaning and spread of the data.

Whereas the *quadratic* and the *cubic* model represent the spread of the data with much higher accuracy, which means that for any give number of bites per meal, these model can predict the kcal consumed per bite reliably.

For instance, if 20 bites are taken per meal then,

- The *quadratic rational* model predicts that 0.225 kcal is consumed per bite
- The *Exponential* model predicts that 0.0000002 kcal is consumed per bite
- The *quadratic* model predicts that 36.66 kcal is consumed per bite
- The *cubic* model predicts that 34.98 kcal is consumed per bite

Seeing the predictions above and also the nature of the model it can concluded that the data spread was best represented by the *quadratic* and the *cubic* model.

4 Conclusion

In general, for any data fitting problem, it can be concluded that outliers in the data set can distort the data representation, and making deductions based on the obtained model can be misleading. Hence, it is necessary to look at and, if possible, remove any potential outliers from the data set to find a more accurate fitting model. An example of this can be seen in the line fitting problem when the outlier (8,14) was added to the existing data set.

On the other hand, in the curve fitting problem, it is of utmost importance to study the data's spread before assuming any non-linear model. Once the nature of the spread is established, only then a non-linear model should be considered and solved.

5 Appendix

5.1 Code: Line fitting problem

```
data_points = [5,1; 6,1; 7,2; 8,3; 9,5];

% Plotting the raw data points
figure;
plot(data_points(:,1), data_points(:,2), '.black', 'MarkerSize',15);
% title('Raw data points');
ylabel('y-axis');
xlabel('x-axis');
legend('Data points');
xlim([0 10]);
ylim([0 10]);
xlim([2.99 11.25])
ylim([-0.02 8.25])
% Number of given data points is 5
N = 5;

% Number of basis function is 2 for the assumed 2-D line model
M = 2;

% Matrix A of order N X M
A = [5 1; 6 1; 7 1; 8 1; 9 1];

% Matrix B of order N x 1
B = [1; 1; 2; 3; 5];

% Solving the linear unknowns
X = inv(transpose(A)*A)*transpose(A)*B;

% Linear unkowns
a1 = X(1,1)
a2 = X(2,1)

% Plotting the 2-D line model and the data point
x = 4:0.5:10;
y = a1*x + a2;
figure;
hold on;
plot(x,y, 'black');
plot(data_points(:,1), data_points(:,2), '.black', 'MarkerSize',15);
```

```
% title('Raw data points with 2-D line model');
ylabel('y-axis');
xlabel('x-axis');
legend('2-D line model', 'Data');
xlim([0 10]);
ylim([0 10]);
hold off;
```

5.2 Code: Line fitting problem (after adding (8,14))

```
% Given data points inlcuding (8,14)
data_points_new = [5,1; 6,1; 7,2; 8,3; 9,5; 8,14];

% Plotting the raw data points
figure;
plot(data_points_new(:,1), data_points_new(:,2), '.black', 'MarkerSize',15)
% title('New raw data points');
ylabel('y-axis')
xlabel('x-axis')
legend('Data')
xlim([0 15]);
ylim([0 15]);

% Number of given data points is 6
N_new = 6;

% Number of basis function is 2 for the assumed 2-D line model
M_new = 2;

% Matrix A of order N X M
A_new = [5 1; 6 1; 7 1; 8 1; 9 1; 8,1];

% Matrix B of order N x 1
B_new = [1; 1; 2; 3; 5; 14];

% Solving the linear unknowns
X_new = inv(transpose(A_new)*A_new)*transpose(A_new)*B_new;

% Linear unkowns
a1_new = X_new(1,1)
a2_new = X_new(2,1)

% Plotting the 2-D line model and the data point
x_new = 4:0.5:10;
```

```
y_new = a1_new*x_new + a2_new;
figure;
hold on;
plot(x_new,y_new, 'black');
plot(data_points_new(:,1), data_points_new(:,2),  '.black', 'MarkerSize',15);
% title('New raw data points with 2-D line model');
ylabel('y-axis');
xlabel('x-axis');
legend('2-D line model', 'Data', 'Location','northeast');
xlim([0 15]);
ylim([0 15]);
hold off
```

5.3 Code: Curve fitting problem

```
% Loading the data from text file
filename = '83people-all-meals.txt';
delimiterIn = ' ';
data = importdata(filename,delimiterIn);

% Saving data in variables
participant_id = data(:,1);
meal_id = data(:,2);
num_bites_in_meal = data(:,3);
num_kcal_consumed = data(:,4);

% Estimating kilocalories consumed per bite (dividend of the fourth and third column)
kcal_per_bite = rdivide(num_kcal_consumed, num_bites_in_meal);

% Plotting the data
plot(num_bites_in_meal, kcal_per_bite, '.black', 'MarkerSize',2);
% title('Plot of the required data');
xlabel('number of bites taken in the meal');
ylabel('kilocalories per bite');
legend('Data');

plot(num_bites_in_meal, kcal_per_bite, '.black', 'MarkerSize',5);
% title('Zoomed plot of the required data');
xlabel('number of bites taken in the meal');
ylabel('kilocalories per bite');
legend('Data');
xlim([1.4 105.1])
ylim([-0.4 79.2])
```

5.3.1 Non-linear quadratic model

```
Model-1: Fitting a non-linear quadratic model with equation ‘y = a1*x^2 + a2*x + a3‘
% Number of given data points is 3398
N = 3398;

% Number of basis function is 3
M = 3;

% y = a1*x2 + a2*x + a3
x1 = power(kcal_per_bite,2);
x2 = power(kcal_per_bite,1);

% Matrix A of order N X M
A = [x1, x2, ones(N,1)];

% Matrix B of order N x 1
B = num_bites_in_meal;

% Solving the linear unknowns
X = inv(transpose(A)*A)*transpose(A)*B;

% Linear unkowns
a1 = X(1,1)
a2 = X(2,1)
a3 = X(3,1)

% Plotting the 2-D line model and the data point
x = 0:1:250;
y = a1*power(x,2) + a2*power(x,1) + a3;

plot(num_bites_in_meal, kcal_per_bite, '.black', 'MarkerSize',4);
xlim([1.4 105.1])
ylim([-0.4 79.2])
hold on;
plot(x,y, Color='red', LineWidth=1);
legend('Data', 'Quadratic model', 'Location','northeast');
% title('Fitting a quadratic model');
xlabel('number of bites taken in the meal');
ylabel('kilocalories per bite');
hold off;

% Evaluating the model fit by calculating the residual squared
residual = sum(num_bites_in_meal) - (sum(a1*power(kcal_per_bite,2) + a2*power(kcal_per_bite
residualsquared_quad = residual^
```

5.3.2 Non-linear quadratic rational model

```
Model-2: Fitting a non-linear quadratic rational model with equation ‘y = a1*(x/(x^2-1))‘
% Number of given data points is 3398
N = 3398;

% Number of basis function is 1
M = 1;

% Matrix A of order N X M
A = kcal_per_bite./(kcal_per_bite.^2-1);

% Matrix B of order N x 1
B = num_bites_in_meal;

% Solving the linear unknowns
X = inv(transpose(A)*A)*transpose(A)*B;

% Linear unkowns
a1 = X

% Plotting the 2-D line model and the data point
x = 0:1:250;
y = a1*(x./(x.^2-1));
plot(num_bites_in_meal, kcal_per_bite, '.black', 'MarkerSize',4);
xlim([1.4 105.1])
ylim([-0.4 79.2])
hold on;
plot(x,y, Color='red', LineWidth=1);
legend('Data', 'Quadratic rational model', 'Location','northeast');
% title('Fitting a quadratic rational function');
xlabel('number of bites taken in the meal');
ylabel('kilocalories per bite');
hold off;

% Evaluating the model fit by calculating the residual squared
residual = sum(num_bites_in_meal) - sum(a1*(kcal_per_bite./(kcal_per_bite.^2-1)));
residualsquared_ratqaud = residual^2
```

5.3.3 Non-linear cubic model

```
Model-3: Fitting a non-linear cubic model with equation ‘y = a1*x^3 + a2*x^2 + a3*x1 + a4‘
```

```
% Number of given data points is 3398
N = 3398;

% Number of basis function is 4
M = 4;

% y = a1*x3 + a2*x2 + a3*x1 + a4
x3 = power(kcal_per_bite,3);
x2 = power(kcal_per_bite,2);
x1 = power(kcal_per_bite,1);

% Matrix A of order N X M
A = [x3, x2, x1, ones(N,1)];

% Matrix B of order N x 1
B = num_bites_in_meal;

% Solving the linear unknowns
X = inv(transpose(A)*A)*transpose(A)*B;

% Linear unkowns
a1 = X(1,1)
a2 = X(2,1)
a3 = X(3,1)
a4 = X(4,1)

% Plotting the 2-D line model and the data point
x = 0:1:200;
y = a1*power(x,3) + a2*power(x,2) + a3*x + a4;

plot(num_bites_in_meal, kcal_per_bite, '.black', 'MarkerSize',4);
xlim([1.4 105.1])
ylim([-0.4 79.2])
hold on;
plot(x,y, Color='red', LineWidth=1);
legend('Data', 'Cubic model', 'Location','northeast');
% title('Fitting a cubic model');
xlabel('number of bites taken in the meal');
ylabel('kilocalories per bite');
hold off;

% Evaluating the model fit by calculating the residual squared
residual = sum(num_bites_in_meal) - sum(a1*power(kcal_per_bite,3) + a2*power(kcal_per_bite,2
residualsquared_cubic = residual^2
```

5.3.4 Non-linear exponential model

```
Model-4: Fitting a non-linear exponential fucntion with equation ‘y = a1*e^-x‘
% Number of given data points is 3398
N = 3398;

% Number of basis function is 1
M = 1;

% Matrix A of order N X M
A = [exp(-kcal_per_bite)];

% Matrix B of order N x 1
B = num_bites_in_meal;

% Solving the linear unknowns
X = inv(transpose(A)*A)*transpose(A)*B;

% Linear unkowns
a1 = X

% Plotting the 2-D line model and the data point
x = 0:1:200;
y = a1*exp(-x);
plot(num_bites_in_meal, kcal_per_bite, '.black', 'MarkerSize',4);
xlim([1.4 105.1])
ylim([-0.4 79.2])
hold on;
plot(x,y, Color='red', LineWidth=1);
legend('Data', 'Exponential model', 'Location','northeast');
% title('Fitting an exponential model');
xlabel('number of bites taken in the meal');
ylabel('kilocalories per bite');
hold off;

% Evaluating the model fit by calculating the residual squared
residual = sum(num_bites_in_meal) - sum(a1*exp(-kcal_per_bite));
residualsquared_exp = residual^2
Evaluating all the models based on their residual squared value.
% Finding the model with the least residual value out of the four models
residual_values = [residualsquared_quad, residualsquared_ratqaud, residualsquared_cubic, res
min(residual_values)
```