Ayush Gaggar HOMEWORK 6 11/19/20

1. a) General:

$$f_3(x) = \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} f(x_0) + \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} f(x_1) +$$

$$\frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} f(x_2) + \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} f(x_2)$$

Actual: $(x_0, f(x_0)) = (27.5, 6.430) \longrightarrow (x_3, f(x_3)) = (51.5, 6.261)$

$$f_3(x) = \frac{6.430(x-35.5)(x-43.5)(x-51.5)}{(27.5-35.5)(27.5-43.5)(27.5-51.5)} + \frac{6.490(x-27.5)(x-43.5)(x-51.5)}{(35.5-27.5)(35.5-43.5)(35.5-51.5)} +$$

$$\frac{6.371(x-35.5)(x-27.5)(x-51.5)}{(43.5-35.5)(43.5-27.5)(43.5-51.5)} + \frac{6.261(x-35.5)(x-43.5)(x-27.5)}{(51.5-35.5)(51.5-43.5)(51.5-27.5)}$$

b)

i	x_i	$f(x_i)$
0	27.5	6.430
1	35.5	6.490
2	43.5	6.371
3	51.5	6.261

1) $a_1(27.5)^2 + b_1(27.5) + c_1 = 6.430$

2) $a_1(35.5)^2 + b_1(35.5) + c_1 = 6.490$

3) $a_2(35.5)^2 + b_2(35.5) + c_2 = 6.490$

4) $a_2(43.5)^2 + b_2(43.5) + c_2 = 6.371$

5) $a_3(43.5)^2 + b_3(43.5) + c_3 = 6.371$

6) $a_3(51.5)^2 + b_3(51.5) + c_3 = 6.261$

7) $2a_1(35.5) + b_1 = 2a_2(35.5) + b_2$

8) $2a_2(43.5) + b_2 = 2a_3(43.5) + b_3$

9) $a_1 = 0$

From Matlab: $a_1 = 0$, $b_1 = 0.0075$, $c_1 = 6.2237$; $a_2 = -0.0009$, $b_2 = 0.0580$, $c_2 = 5.5945$; $a_3 = -0.0045$, $b_3 = 0.4157$, $c_3 = -3.1120$

4. a) $T(t) = T_0 e^{-ct}$

$$\text{error} = (T(t) - T_0 e^{-ct}) \Rightarrow e = \sum_{i=1}^{n} (T(t_i) - T_0 e^{-ct_i})^2$$

b) $\ln(T(t)) = \ln T_0 - Ct$. let $\ln(T(t)) = y_m$; $\ln T_0 = a_0$, $-C = a_1$

$$y_m = a_0 + a_1 t$$

$$e = \sum_{i=1}^{n} (y_i - y_m)^2$$

$$= \sum_{i=1}^{n} (y_i - a_0 - a_1 t_i)^2$$

$$\frac{\partial e}{\partial a_0} = \sum_{i=1}^{n} (2(y_i - a_0 - a_1 t)(-1) = 0$$

$$1 \quad \sum_{i=1}^{n} y_i = a_0 n + a_1 \sum_{i=1}^{n} t_i$$

$$\frac{\partial e}{\partial a_1} = \sum_{i=1}^{n} 2(y_i - a_0 - a_1 t_i)(-t_i) = 0$$

$$2 \quad \sum_{i=1}^{n} t_i y_i = a_0 \sum_{i=1}^{n} t_i + a_1 \sum_{i=1}^{n} t_i^2$$

c)

$$\begin{bmatrix} n & \sum_{i=1}^{n} t_i \\ \sum_{i=1}^{n} t_i & \sum_{i=1}^{n} t_i^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} t_i y_i \end{bmatrix}$$

Assignment 6

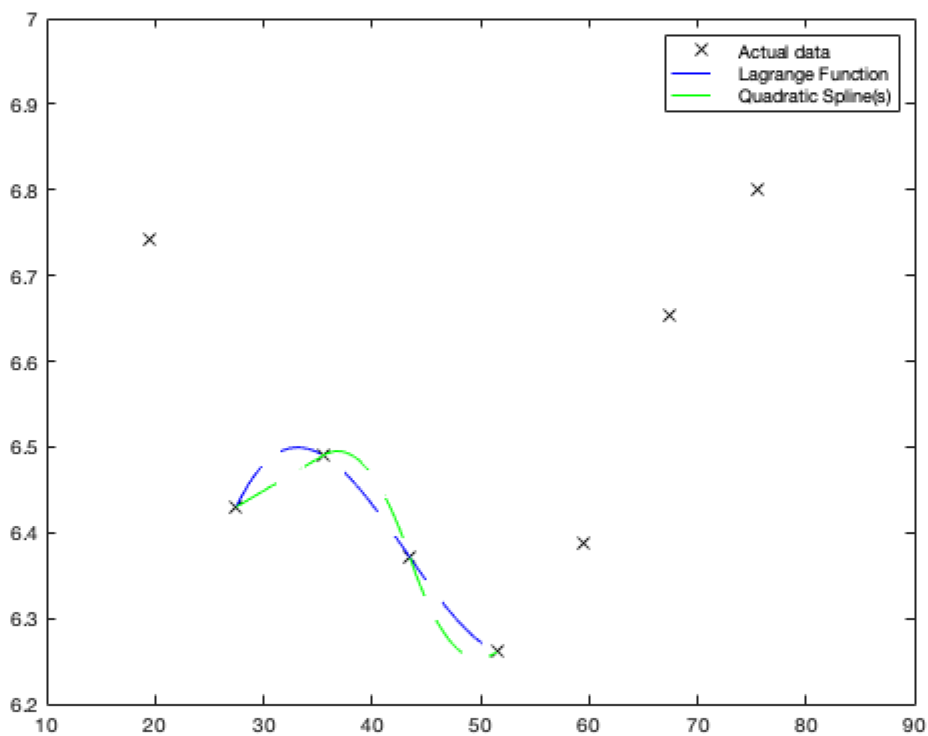
Table of Contents

Problem 1: Lagrange and Spline Interpolation

```
clc; close all; clear all;
% Part b, coefficients of quadratic spline
x0 = 27.5; x1 = 35.5; x2 = 43.5; x3 = 51.5;
fx0 = 6.430; fx1 = 6.490; fx2 = 6.371; fx3 = 6.261;
% rows of matrix, from equations numbered on work on paper
% qn = [a1, b1, c1, a2, b2, c2, a3, b3, c3]
q1 = [x0^2, x0, 1, 0, 0, 0, 0, 0, 0];
q2 = [x1^2, x1, 1, 0, 0, 0, 0, 0, 0];
q3 = [0, 0, 0, x1^2, x1, 1, 0, 0, 0];
q4 = [0, 0, 0, x2^2, x2, 1, 0, 0, 0];
q5 = [0, 0, 0, 0, 0, 0, x2^2, x2, 1];
q6 = [0, 0, 0, 0, 0, 0, x3^2, x3, 1];
q7 = [2*x1, 1, 0, -2*x1, -1, 0, 0, 0, 0];
q8 = [0, 0, 0, 2*x2, 1, 0, -2*x2, -1, 0];
q9 = [1, 0, 0, 0, 0, 0, 0, 0, 0];
A = [q1; q2; q3; q4; q5; q6; q7; q8; q9];
B = [fx0; fx1; fx1; fx2; fx2; fx3; 0; 0; 0];
coeff = A\B;

x = linspace(27.5, 51.5, 1400);
age = [19.5, 27.5, 35.5, 43.5, 51.5, 59.5, 67.5, 75.5, 83.5];
wellbeing = [6.742, 6.430, 6.490, 6.371, 6.261, 6.388, 6.653, 6.801,
 6.922];

lagrange1 = fx0*(x-x1).*(x-x2).*(x-x3)/((x0-x1)*(x0-x2)*(x0-x3));
lagrange2 = fx1*(x-x0).*(x-x2).*(x-x3)/((x1-x0)*(x1-x2)*(x1-x3));
lagrange3 = fx2*(x-x1).*(x-x0).*(x-x3)/((x2-x0)*(x2-x1)*(x2-x3));
lagrange4 = fx3*(x-x1).*(x-x2).*(x-x0)/((x3-x1)*(x3-x2)*(x3-x0));
lagrange = lagrange1 + lagrange2 + lagrange3 + lagrange4;

qspline1 = coeff(1)*linspace(27.5, 35.5).^2 + coeff(2)*linspace(27.5,
 35.5) + coeff(3);
qspline2 = coeff(4)*linspace(35.5, 43.5).^2 + coeff(5)*linspace(35.5,
 43.5) + coeff(6);
qspline3 = coeff(7)*linspace(43.5, 51.5).^2 + coeff(8)*linspace(43.5,
 51.5) + coeff(9);
```

```
figure(1);
plot(age, wellbeing, 'xk');
hold all
plot(x, lagrange, '--b');
plot(linspace(27.5, 35.5), qspline1, '--g');
plot(linspace(35.5, 43.5), qspline2, '--g');
plot(linspace(43.5, 51.5), qspline3, '--g');
legend('Actual data', 'Lagrange Function', 'Quadratic Spline(s)');

% Interpolated values at age = 40
x = 40;
lagrange1_40 = fx0*(x-x1)*(x-x2)*(x-x3)/((x0-x1)*(x0-x2)*(x0-x3));
lagrange2_40 = fx1*(x-x0)*(x-x2)*(x-x3)/((x1-x0)*(x1-x2)*(x1-x3));
lagrange3_40 = fx2*(x-x1)*(x-x0)*(x-x3)/((x2-x0)*(x2-x1)*(x2-x3));
lagrange4_40 = fx3*(x-x1)*(x-x2)*(x-x0)/((x3-x1)*(x3-x2)*(x3-x0));
lagrange_40 = lagrange1_40 + lagrange2_40 + lagrange3_40 +
 lagrange4_40;
qspline2_40 = coeff(4)*x^2 + coeff(5)*x + coeff(6);
fprintf('Using the Lagrange Method, Prof. Tolley''s wellbeing score is
 %f\n', lagrange_40);
fprintf('Using Quadratic Splines, Prof. Tolley''s wellbeing score is
 %f\n', qspline2_40);
```

Using the Lagrange Method, Prof. Tolley's wellbeing score is 6.433040
Using Quadratic Splines, Prof. Tolley's wellbeing score is 6.467113



Problem 2: Linear Regression

```
clc; close all; clear all;
[xcor, ycor] = readvars('data_testing.txt');
% Part a
n = numel(xcor);
a1 = (n*sum(xcor.*ycor) - sum(xcor)*sum(ycor))/(n*sum(xcor.^2) -
 (sum(xcor))^2);
a0 = sum(ycor)/n - a1*sum(xcor)/n;
fprintf('The coefficients a0 and a1 respectively are %f and %f.\n',
 a0, a1);

% Part b
Sr = sum((ycor - a0 - a1*xcor).^2);
S = sqrt(Sr/(n-2));
fprintf('The Standard Error of Regression S = %f.\n', S);

% Part c
St = sum((ycor - sum(ycor)/n).^2);
r2 = (St - Sr)/St;
fprintf('The Coefficient of Determination r^2 = %f.\n', r2);

The coefficients a0 and a1 respectively are 12.304799 and 84.103919.
The Standard Error of Regression S = 183.744892.
The Coefficient of Determination r^2 = 0.672728.
```

Problem 3: Polynomial Regression

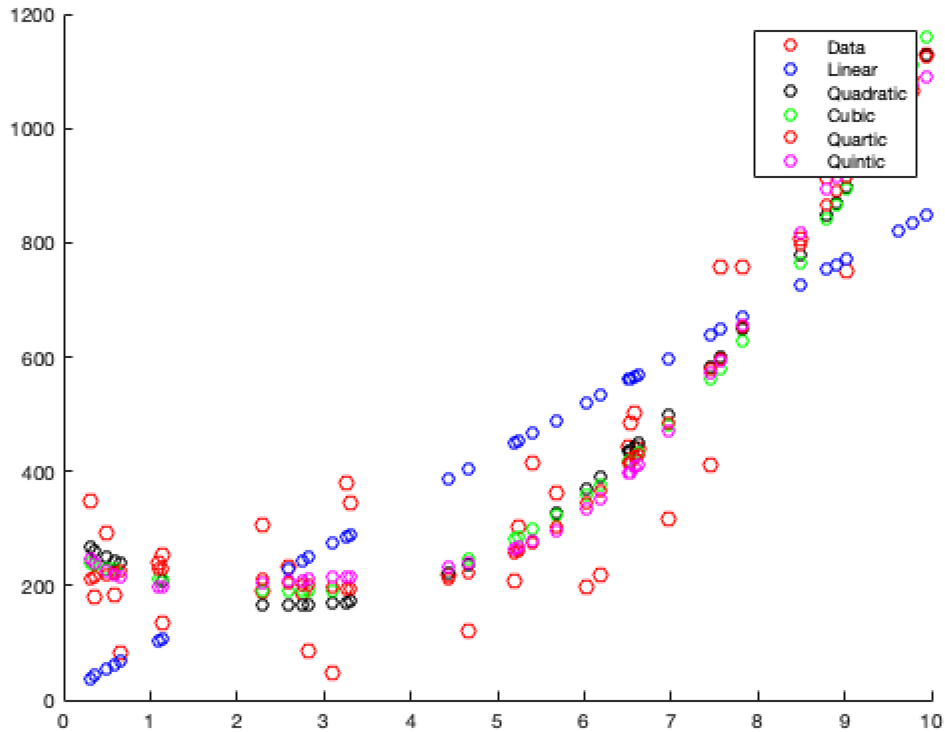
```
clc;
% Part a
linear = polyfit(xcor, ycor, 1);
quadratic = polyfit(xcor, ycor, 2);
cubic = polyfit(xcor, ycor, 3);
quartic = polyfit(xcor, ycor, 4);
quintic = polyfit(xcor, ycor, 5);

figure(1);
hold all;
scatter(xcor, ycor, 55, 'or');
scatter(xcor, polyval(linear, xcor), 'ob');
scatter(xcor, polyval(quadratic, xcor), 'ok');
scatter(xcor, polyval(cubic, xcor), 'og');
scatter(xcor, polyval(quartic, xcor), 'or');
scatter(xcor, polyval(quintic, xcor), 'om');
legend('Data', 'Linear', 'Quadratic', 'Cubic', 'Quartic', 'Quintic');

% Part b
Sr = sum((ycor - polyval(linear, xcor)).^2);
S = sqrt(Sr/(n-2));
fprintf('Linear Reg: The Standard Error of Regression S = %f.\n', S);
St = sum((ycor - sum(ycor)/n).^2);
r2 = (St - Sr)/St;
```

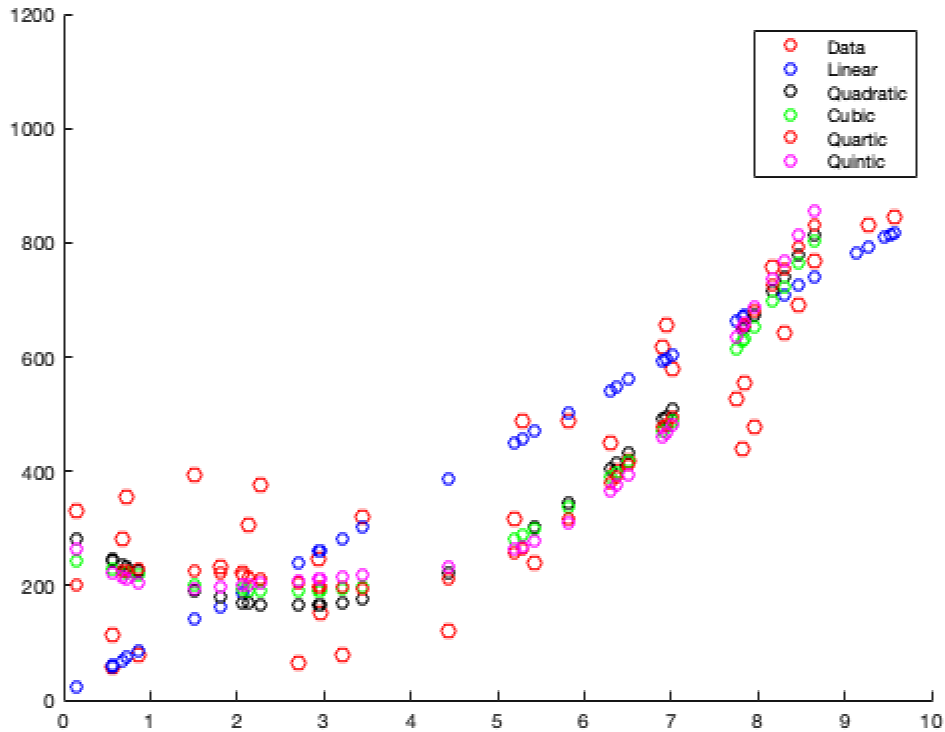
```
fprintf('Linear Reg: The Coefficient of Determination r^2 = %f.\n\n',
 r2);

Sr = sum((ycor - polyval(quadratic, xcor)).^2);
S = sqrt(Sr/(n-3));
fprintf('Quadratic Reg: The Standard Error of Regression S = %f.\n',
 S);
St = sum((ycor - sum(ycor)/n).^2);
r2 = (St - Sr)/St;
fprintf('Quadratic Reg: The Coefficient of Determination r^2 = %f.\n
\n', r2);

Sr = sum((ycor - polyval(cubic, xcor)).^2);
S = sqrt(Sr/(n-4));
fprintf('Cubic Reg: The Standard Error of Regression S = %f.\n', S);
St = sum((ycor - sum(ycor)/n).^2);
r2 = (St - Sr)/St;
fprintf('Cubic Reg: The Coefficient of Determination r^2 = %f.\n\n',
 r2);

Sr = sum((ycor - polyval(quartic, xcor)).^2);
S = sqrt(Sr/(n-5));
fprintf('Quartic Reg: The Standard Error of Regression S = %f.\n', S);
St = sum((ycor - sum(ycor)/n).^2);
r2 = (St - Sr)/St;
fprintf('Quartic Reg: The Coefficient of Determination r^2 = %f.\n\n',
 r2);

Sr = sum((ycor - polyval(quintic, xcor)).^2);
S = sqrt(Sr/(n-6));
fprintf('Quintic Reg: The Standard Error of Regression S = %f.\n', S);
St = sum((ycor - sum(ycor)/n).^2);
r2 = (St - Sr)/St;
fprintf('Quintic Reg: The Coefficient of Determination r^2 = %f.\n\n',
 r2);

% Part c
%{
Based on this data alone, the best fit would be the Quintic
 Regression,
as it has the highest r^2 value, meaning a greater percentage of data
fits the quintic regression as compared to the other regression
 models.
%}

% Part d
[xcor_new, ycor_new] = readvars('data_validation.txt');

figure(2); % Represents plots for data validation data set
hold all;
scatter(xcor_new, ycor_new, 55, 'or');
scatter(xcor_new, polyval(linear, xcor_new), 'ob');
scatter(xcor_new, polyval(quadratic, xcor_new), 'ok');
scatter(xcor_new, polyval(cubic, xcor_new), 'og');
```

```
scatter(xcor_new, polyval(quartic, xcor_new), 'or');
scatter(xcor_new, polyval(quintic, xcor_new), 'om');
legend('Data', 'Linear', 'Quadratic', 'Cubic', 'Quartic', 'Quintic');

fprintf('\nPart d, calculating relevant statistics on data validation
 set.\n');
Sr = sum((ycor_new - polyval(linear, xcor_new)).^2);
S = sqrt(Sr/(n-2));
fprintf('Linear Reg: The Standard Error of Regression S = %f.\n', S);
St = sum((ycor_new - sum(ycor_new)/n).^2);
r2 = (St - Sr)/St;
fprintf('Linear Reg: The Coefficient of Determination r^2 = %f.\n\n',
 r2);

Sr = sum((ycor_new - polyval(quadratic, xcor_new)).^2);
S = sqrt(Sr/(n-3));
fprintf('Quadratic Reg: The Standard Error of Regression S = %f.\n',
 S);
St = sum((ycor_new - sum(ycor_new)/n).^2);
r2 = (St - Sr)/St;
fprintf('Quadratic Reg: The Coefficient of Determination r^2 = %f.\n
\n', r2);

Sr = sum((ycor_new - polyval(cubic, xcor_new)).^2);
S = sqrt(Sr/(n-4));
fprintf('Cubic Reg: The Standard Error of Regression S = %f.\n', S);
St = sum((ycor_new - sum(ycor_new)/n).^2);
r2 = (St - Sr)/St;
fprintf('Cubic Reg: The Coefficient of Determination r^2 = %f.\n\n',
 r2);

Sr = sum((ycor_new - polyval(quartic, xcor_new)).^2);
S = sqrt(Sr/(n-5));
fprintf('Quartic Reg: The Standard Error of Regression S = %f.\n', S);
St = sum((ycor_new - sum(ycor_new)/n).^2);
r2 = (St - Sr)/St;
fprintf('Quartic Reg: The Coefficient of Determination r^2 = %f.\n\n',
 r2);

Sr = sum((ycor_new - polyval(quintic, xcor_new)).^2);
S = sqrt(Sr/(n-6));
fprintf('Quintic Reg: The Standard Error of Regression S = %f.\n', S);
St = sum((ycor_new - sum(ycor_new)/n).^2);
r2 = (St - Sr)/St;
fprintf('Quintic Reg: The Coefficient of Determination r^2 = %f.\n\n',
 r2);

% Part e
%{
Based on this new data, the best fit would be the Cubic Regression,
as it has the highest r^2 value, meaning a greater percentage of data
fits the cubic regression as compared to the other regression models.
My answer is different from part c, because I am comparing various
 polynomial
```

```
fits to find the regression with the largest r^2 value. We are testing
 the
polynomial regressions calculated in part b to see if they still work
 well
with a validation set of data. We can expect to see different
 statistical
values, and we want to pick the regression with the largest r^2 value
to pick the regression model that fits the data the best.
%}
```

```
Linear Reg: The Standard Error of Regression S = 183.744892.
Linear Reg: The Coefficient of Determination r^2 = 0.672728.

Quadratic Reg: The Standard Error of Regression S = 107.787443.
Quadratic Reg: The Coefficient of Determination r^2 = 0.890344.

Cubic Reg: The Standard Error of Regression S = 107.642542.
Cubic Reg: The Coefficient of Determination r^2 = 0.893594.

Quartic Reg: The Standard Error of Regression S = 107.416985.
Quartic Reg: The Coefficient of Determination r^2 = 0.896983.

Quintic Reg: The Standard Error of Regression S = 107.216669.
Quintic Reg: The Coefficient of Determination r^2 = 0.900299.


Part d, calculating relevant statistics on data validation set.
Linear Reg: The Standard Error of Regression S = 162.644233.
Linear Reg: The Coefficient of Determination r^2 = 0.678141.

Quadratic Reg: The Standard Error of Regression S = 125.871052.
Quadratic Reg: The Coefficient of Determination r^2 = 0.812303.

Cubic Reg: The Standard Error of Regression S = 124.755775.
Cubic Reg: The Coefficient of Determination r^2 = 0.820598.

Quartic Reg: The Standard Error of Regression S = 130.128030.
Quartic Reg: The Coefficient of Determination r^2 = 0.810236.

Quintic Reg: The Standard Error of Regression S = 136.884740.
Quintic Reg: The Coefficient of Determination r^2 = 0.796017.
```


Data
Linear
Quadratic
Cubic
Quartic
Quintic
0
200
400
600
800
1000
1200
0
1
2
3
4
5
6
7
8
9
10

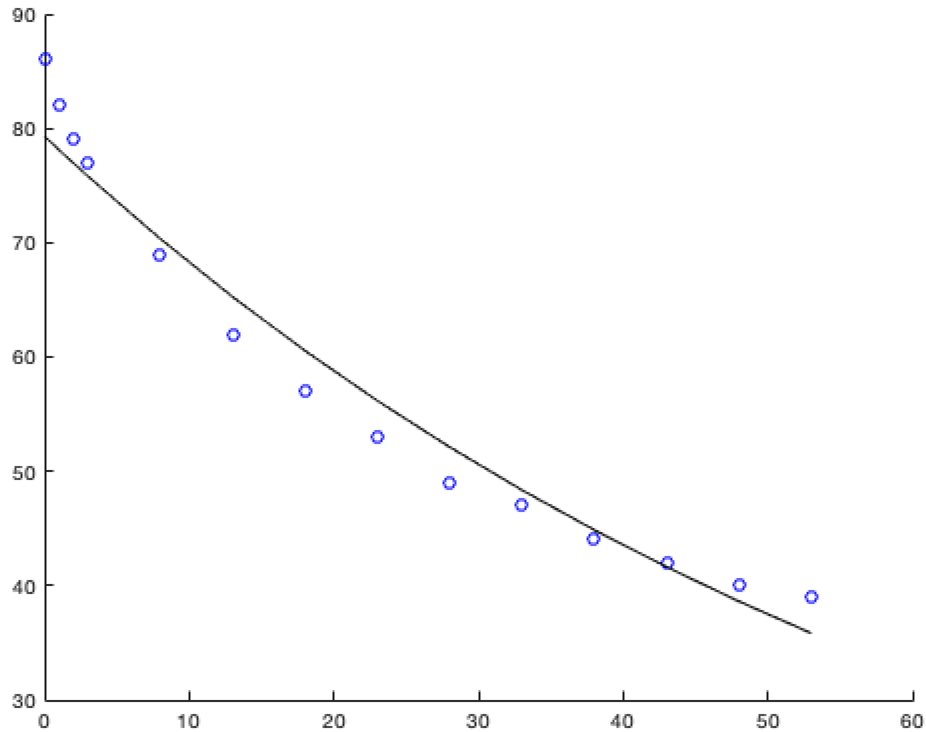

Data
Linear
Quadratic
Cubic
Quartic
Quintic
0
200
400
600
800
1000
1200
0
1
2
3
4
5
6
7
8
9
10

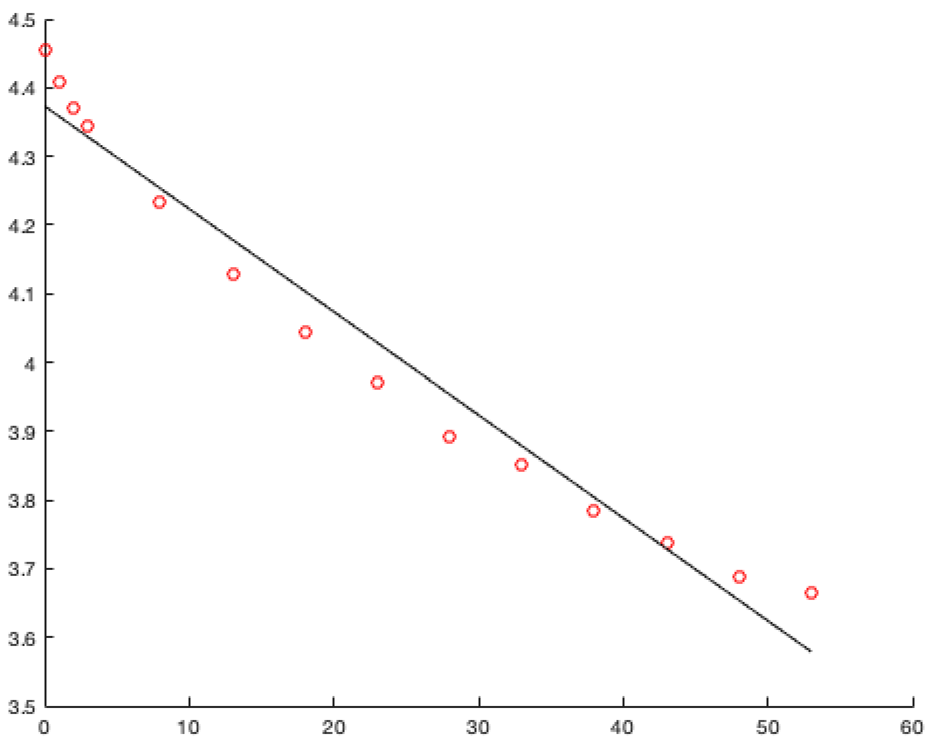
Problem 4: Regression with Linearized Data

Part d

```
clc; close all; clear all;
t = [0 1 2 3 8 13 18 23 28 33 38 43 48 53];
temp_original = [86 82 79 77 69 62 57 53 49 47 44 42 40 39];
temp = log(temp_original);
n = numel(t);

A = [n, sum(t); sum(t), sum(t.^2)];
B = [sum(temp); sum(temp.*t)];
coeff = A\B;
fprintf('The coefficients of a least squared fit would be a0 = %f and
 a1 = %f.\n', coeff(1), coeff(2));
fprintf('Thus, T0 = %f and C = %f\n', exp(coeff(1)), -1*coeff(2));

% Part e
figure(1); %Part i
scatter(t, temp_original, 'ob');
hold on;
plot(t, exp(coeff(1))*exp(coeff(2)*t), '-k');

figure(2) % Part ii - linearized
scatter(t, temp, 'or');
hold on;
temp_plot =  coeff(1) + coeff(2)*t;
plot(t, temp_plot, '-k');
```

```
The coefficients of a least squared fit would be a0 = 4.373213 and a1
 = -0.014983.
Thus, T0 = 79.298029 and C = 0.014983
```


90
80
70
60
50
40
30
0
10
20
30
40
50
60


4.5
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.6
3.5
0
10
20
30
40
50
60

Published with MATLAB® R2020a