```
% Mahri K[redacted]
% BE 3343
% Date: 05/10/2022
% Student ID: [redacted]
%Single Student Project
% Chapter questions
```

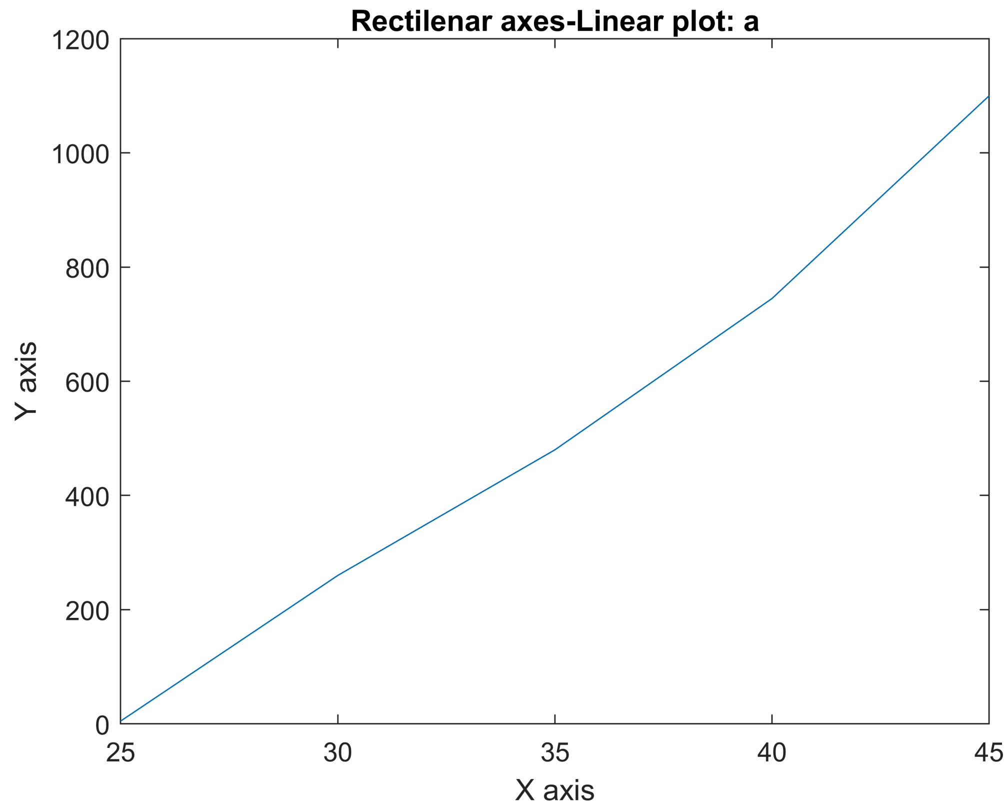
```
% Question 1 (chapter 6 question 2)

% Description: in each of the following problems, determine the best
% function y(x) (linear, exponential, or power function) to describe the
% data. Plot the function on the same plot with the data. Label and format
% the plots appropriately.

% a

% Define x and y
x1=[25 30 35 40 45]
```

```
x1 = 1×5
    25    30    35    40    45
```

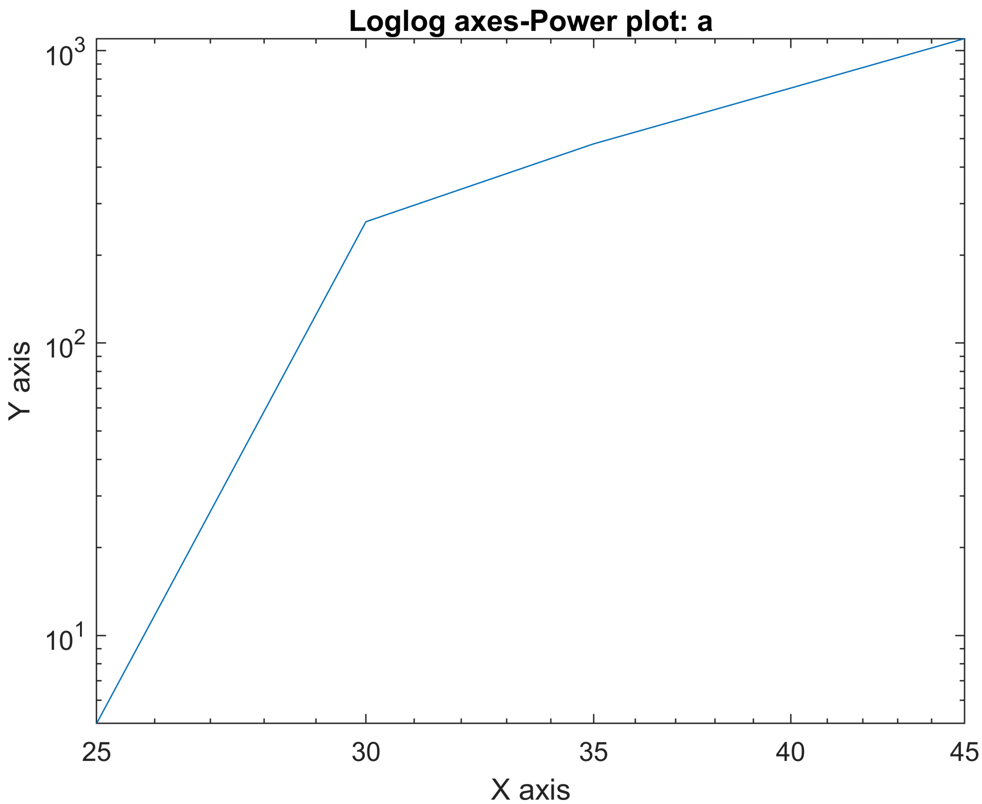
```
y1=[5 260 480 745 1100]
```

```
y1 = 1×5
         5         260         480         745        1100
```

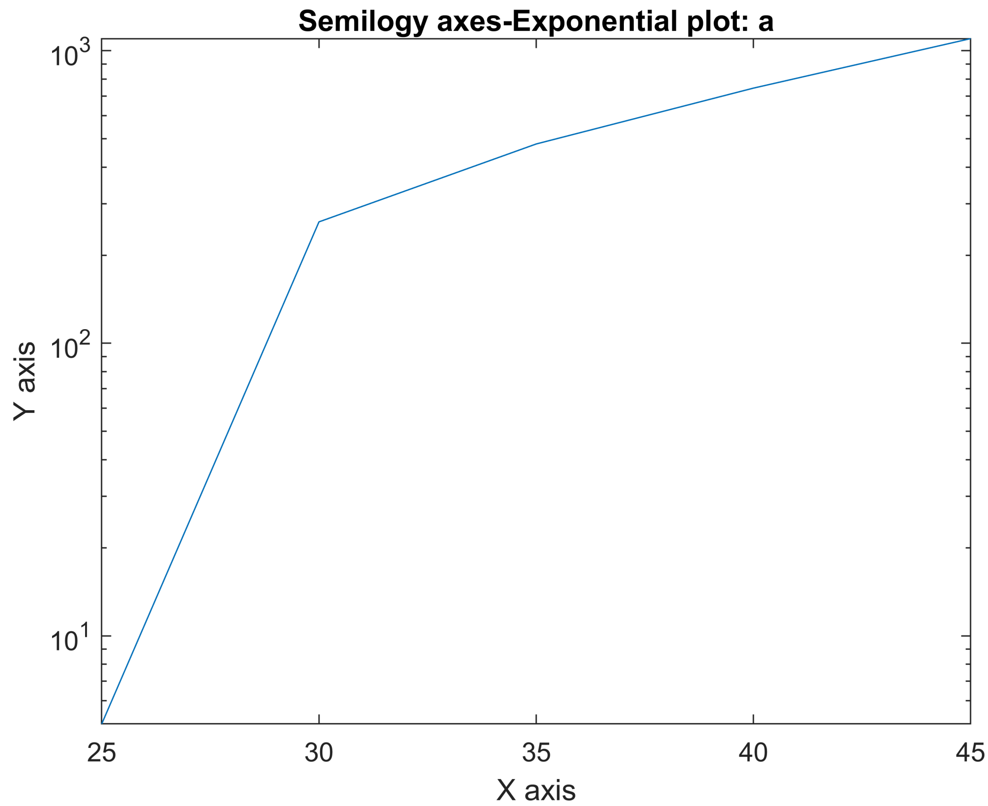
```
% Plot the x and y on rectilinear, loglog and semilogy axes

figure
plot(x1,y1)
title("Rectilenar axes-Linear plot: a")
xlabel("X axis")
ylabel("Y axis")
```




```
figure
loglog(x1,y1)
title("Loglog axes-Power plot: a")
xlabel("X axis")
ylabel("Y axis")
```




```
figure
semilogy(x1,y1)
title("Semilogy axes-Exponential plot: a")
xlabel("X axis")
ylabel("Y axis")
```




```
% Analyze the plots to see which plot gives a straight line or close to a
% straight line

a1=polyfit(x1,y1,1)
```

```
a1 = 1×2
10^3 ×
    0.0535   -1.3545
```

```
% b

% Define x and y
x2=[2.5 3 3.5 4 4.5 5 5.5 6 7 8 9 10]
```

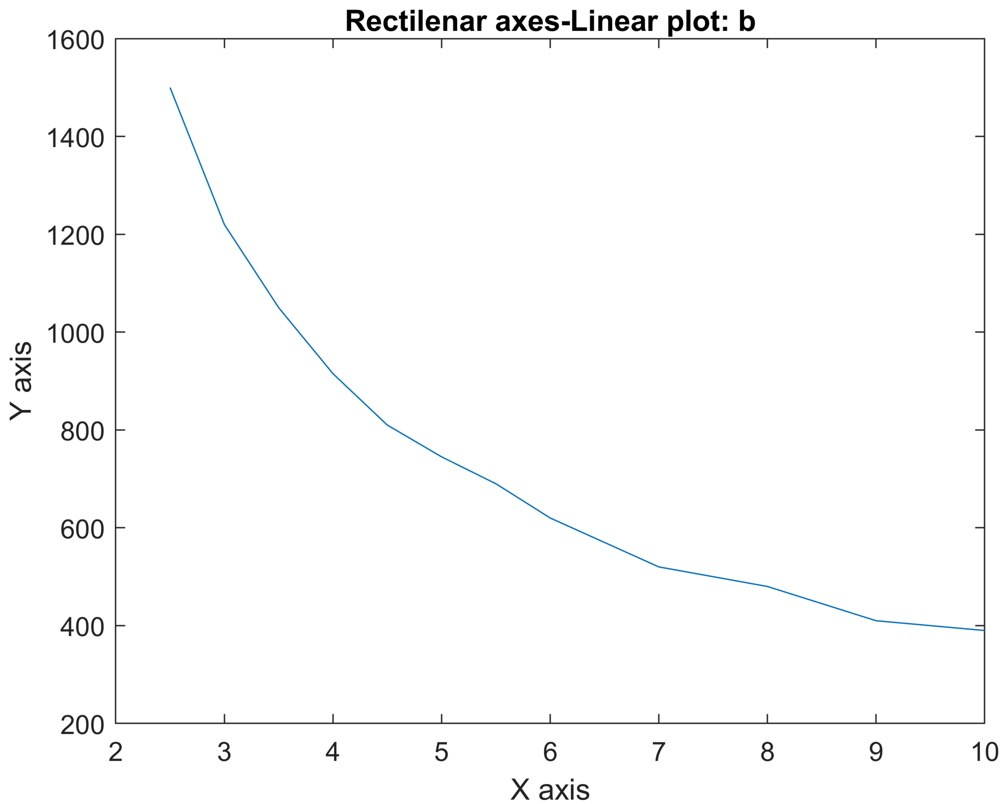
```
x2 = 1×12
    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000    5.5000    6.0000 ⋯
```

```
y2=[1500 1220 1050 915 810 745 690 620 520 480 410 390]
```

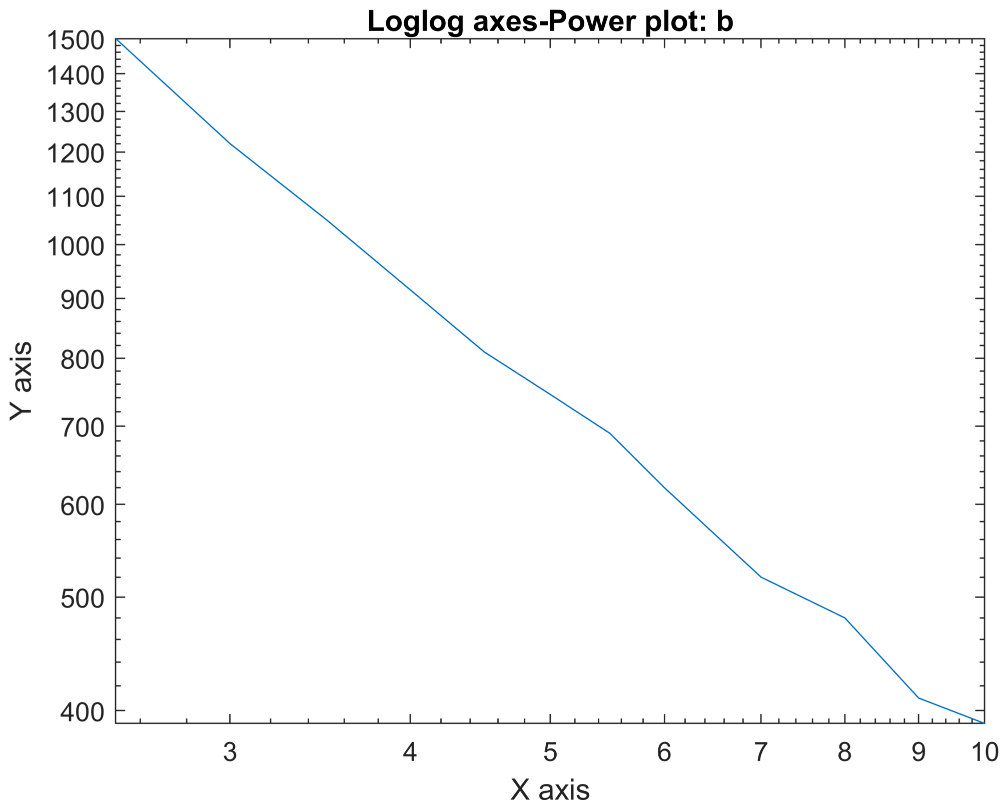
```
y2 = 1×12
        1500        1220        1050         915         810         745 ⋯
```

```
% Plot the x and y on rectilinear, loglog and semilogy axes

figure
plot(x2,y2)
```

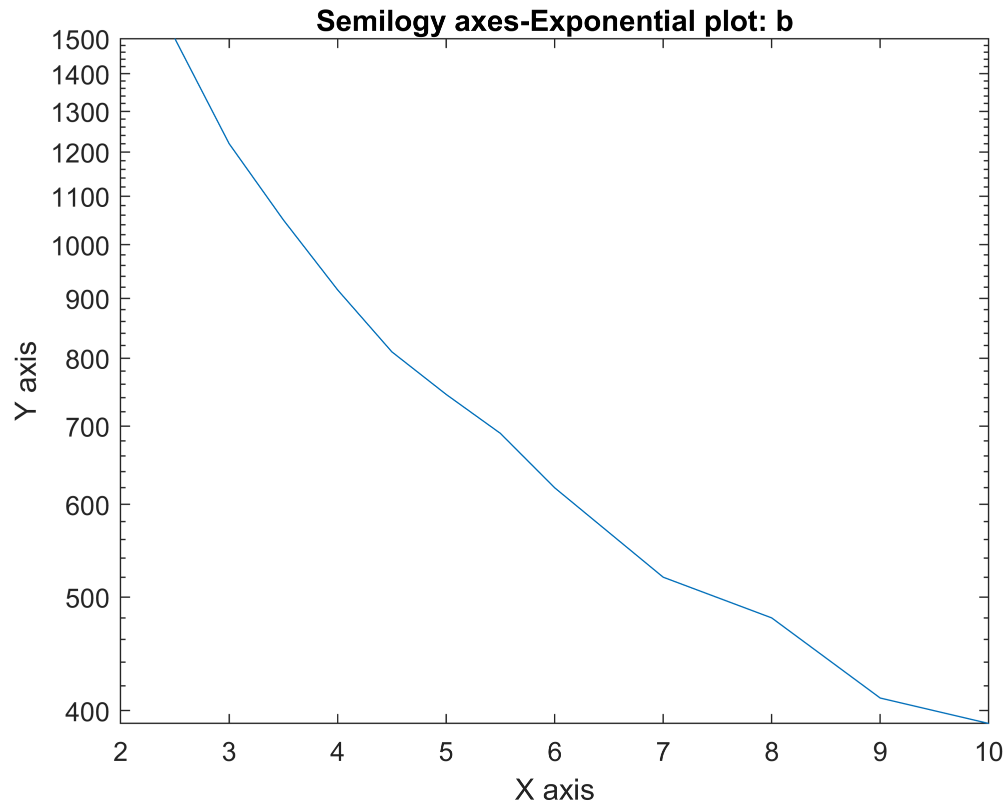
```
title("Rectilenar axes-Linear plot: b")
xlabel("X axis")
ylabel("Y axis")
```




```
figure
loglog(x2,y2)
title("Loglog axes-Power plot: b")
xlabel("X axis")
ylabel("Y axis")
```




```
figure
semilogy(x2,y2)
title("Semilogy axes-Exponential plot: b")
xlabel("X axis")
ylabel("Y axis")
```



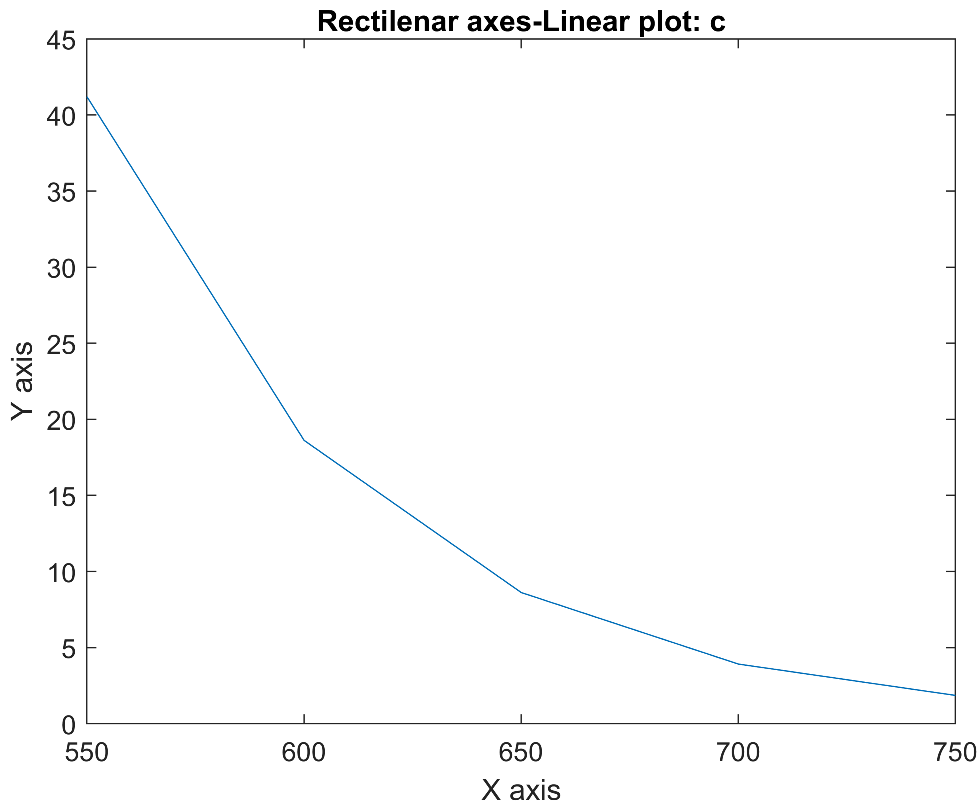

```
% Analyze the plots to see which plot gives a straight line or close to a
% straight line

% c

% Define x and y
x3=[550 600 650 700 750]
```

```
x3 = 1×5
   550   600   650   700   750
```

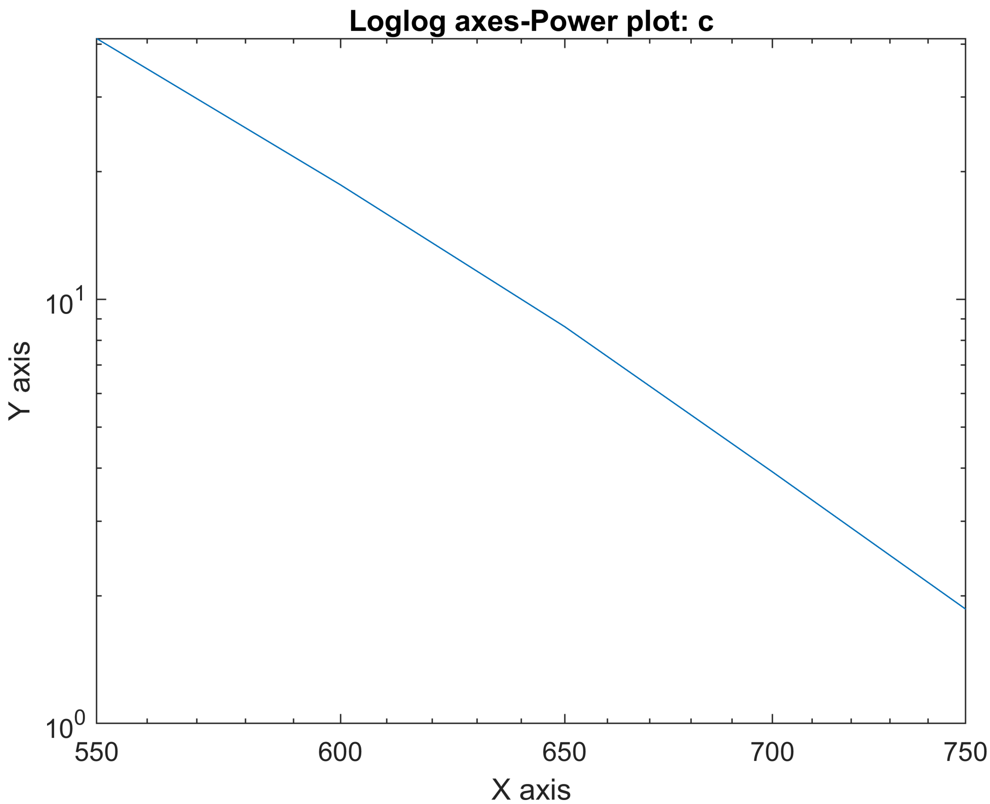
```
y3=[41.2 18.62 8.62 3.92 1.86]
```

```
y3 = 1×5
   41.2000   18.6200    8.6200    3.9200    1.8600
```

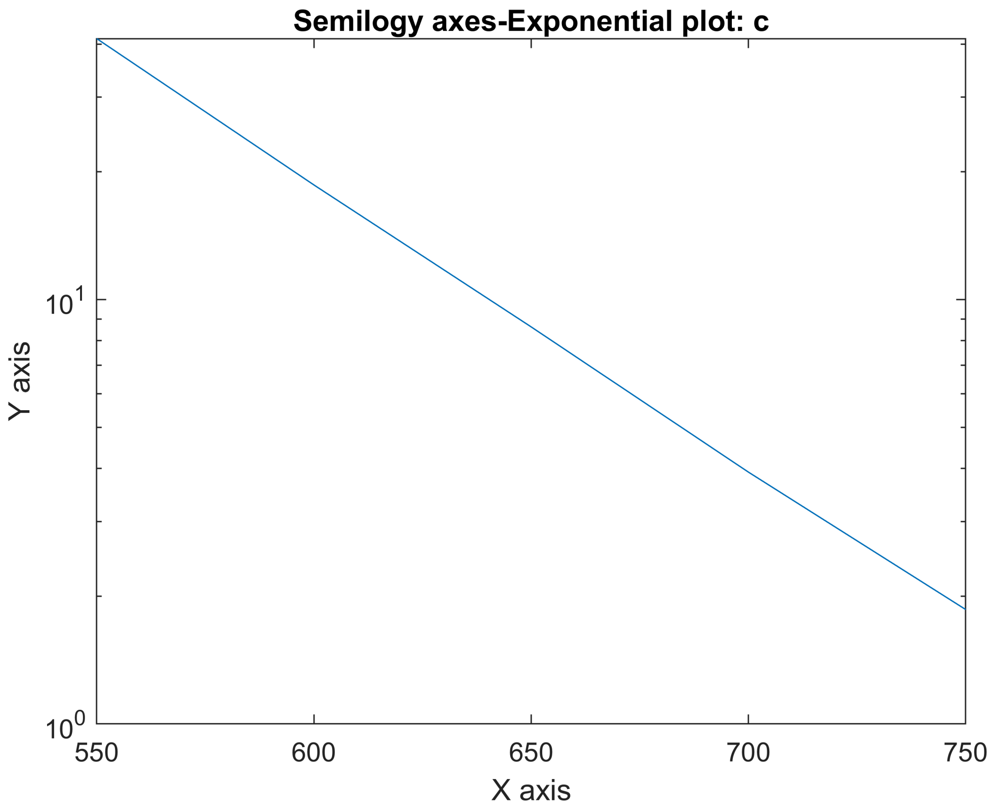
```
% Plot the x and y on rectilinear, loglog and semilogy axes

figure
plot(x3,y3)
title("Rectilenar axes-Linear plot: c")
xlabel("X axis")
ylabel("Y axis")
```




```
figure
loglog(x3,y3)
title("Loglog axes-Power plot: c")
xlabel("X axis")
ylabel("Y axis")
```




```
figure
semilogy(x3,y3)
title("Semilogy axes-Exponential plot: c")
xlabel("X axis")
ylabel("Y axis")
```




```
% Analyze the plots to see which plot gives a straight line or close to a
% straight line


a=['The data set a fits a rectilinear plot since it forms a straight line and the following lin
```

```
a =
'The data set a fits linear plot and the equation is y=53.5*x-1354.5'
```

```
disp(a)
```

```
The data set a fits linear plot and the equation is y=53.5*x-1354.5
```

```
disp("The data set b fits loglog plot because it forms a straight line and the power function r
```

```
The data set b fits power plot
```

```
disp("The data set c fits semilogy and loglog plots since it forms straight line on those plots
```

```
The data set c fits exponential plots
```

```
% Question 2 (chapter 6 question 9)
% Delete all previous information
clear
clc
close all
```

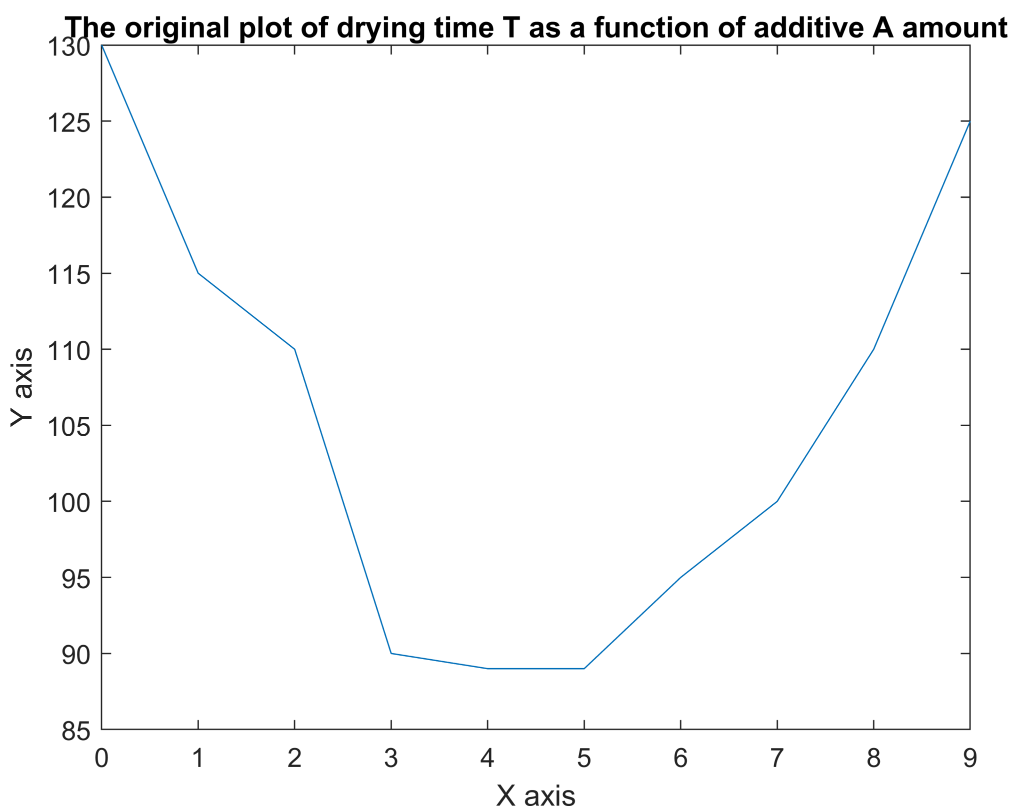
```
% Description: the following data give the drying time T of a certain pint
% as a function of the amount of a certain additive A


%a: Find the first, second, third and fourth degree polynomials that fit
%the data and plot each polynomial with the data. Determine the quality of
%the curve t for each by computing J, S, and r^2

%Define x and y

A=[0 1 2 3 4 5 6 7 8 9] % oz
```

```
A = 1×10
     0     1     2     3     4     5     6     7     8     9
```

```
T=[130 115 110 90 89 89 95 100 110 125] %min
```

```
T = 1×10
   130   115   110    90    89    89    95   100   110   125
```

```
% Plot to observe the original plot of the data

plot(A,T)
title("The original plot of drying time T as a function of additive A amount")
xlabel("X axis")
ylabel("Y axis")
```




```
% first-degree polynomial
```

```
% use polyfit to obtain the first degree function coefficients
p1=polyfit(A,T,1)
```

```
p1 = 1×2
   -0.6970  108.4364
```

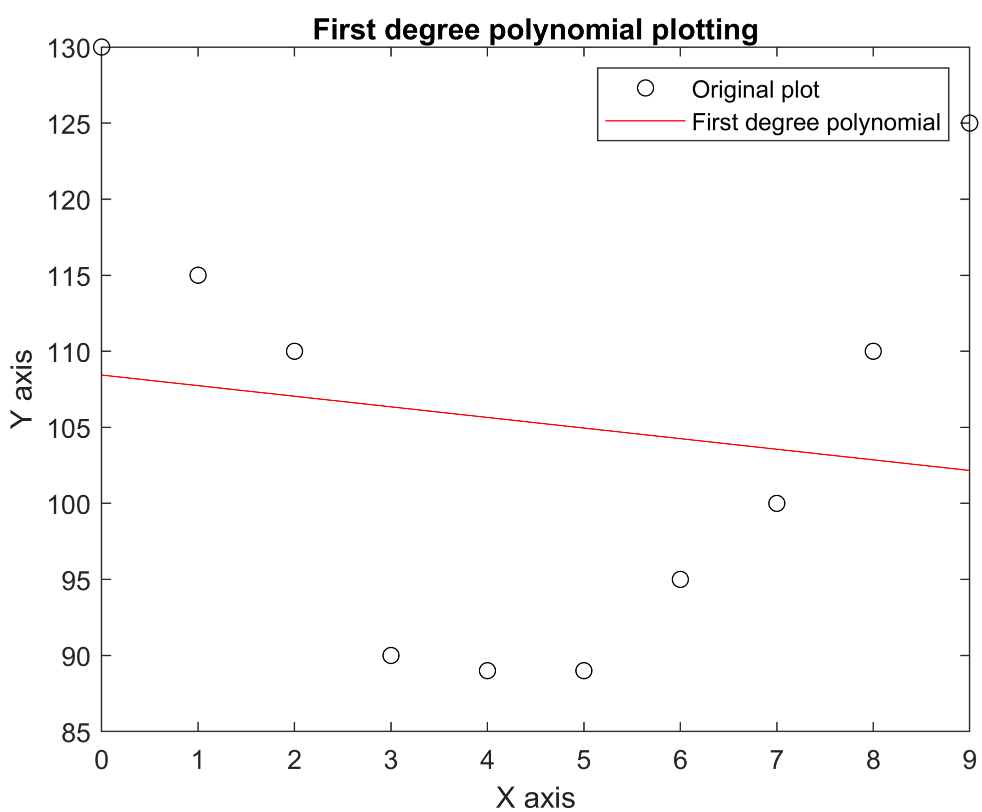
```
% use polyval to obtain y values when inputting A data to p1 first degree
% polynomial function
x1=linspace(min(A),max(A),100)
```

```
x1 = 1×100
         0    0.0909    0.1818    0.2727    0.3636    0.4545    0.5455    0.6364 ⋯
```

```
y1=polyval(p1,x1)
```

```
y1 = 1×100
  108.4364  108.3730  108.3096  108.2463  108.1829  108.1196  108.0562  107.9928 ⋯
```

```
%plot the original and the first degree polynomial data to analyze the
%plots
figure
plot(A,T,'ko')
hold on
plot(x1,y1,'r')
title("First degree polynomial plotting")
xlabel("X axis")
ylabel("Y axis")
legend("Original plot","First degree polynomial")
```



```
% calculate J, S, and r^2 using below formulas

for k=1:4;
    coeff1=polyfit(A,T,1);
    J1(k)=sum((polyval(coeff1,A)-T).^2);
    end

mu1=mean(T);
for k=1:4;
    S1(k)=sum((T-mu1).^2);
    r2_1(k)=1-J1(k)/S1(k);
end
disp("The J value for the first degree polynomial fitting is:")
```

```
The J value for the first degree polynomial fitting is:
```

```
disp(J1)
```

```
   1.0e+03 *

    1.9960    1.9960    1.9960    1.9960
```

```
disp("The S value for the first degree polynomial fitting is:")
```

```
The S value for the first degree polynomial fitting is:
```

```
disp(S1)
```

```
   1.0e+03 *

    2.0361    2.0361    2.0361    2.0361
```

```
disp("The r^2 value for the first degree polynomial fitting is:")
```

```
The r^2 value for the first degree polynomial fitting is:
```

```
disp(r2_1)
```

```
    0.0197    0.0197    0.0197    0.0197
```

```


% second-degree polynomial
% use polyfit to obtain the second degree function coefficients
p2=polyfit(A,T,2)
```

```
p2 = 1×3
    1.9053  -17.8447  131.3000
```

```
% use polyval to obtain y values when inputting A data to p2 second degree
% polynomial function
x2=linspace(min(A),max(A),100)
```

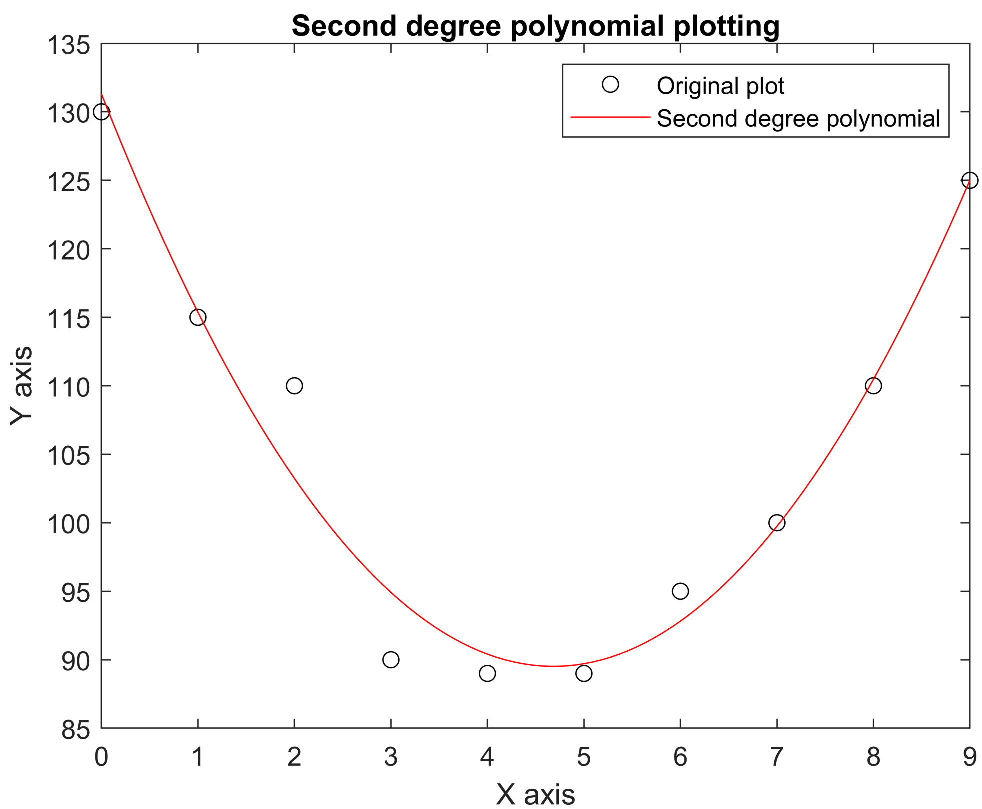
```
x2 = 1×100
         0    0.0909    0.1818    0.2727    0.3636    0.4545    0.5455    0.6364 ···
```

```
y2=polyval(p2,x2)
```

```
y2 = 1×100
```

```
  131.3000  129.6935  128.1185  126.5750  125.0630  123.5824  122.1334  120.7159 ···
```

```
%plot the original and the second degree polynomial data to analyze the
%plots
figure
plot(A,T,'ko')
hold on
plot(x2,y2,'r')
title("Second degree polynomial plotting")
xlabel("X axis")
ylabel("Y axis")
legend("Original plot","Second degree polynomial")
```




```
% calculate J, S, and r^2 using below formulas

for k=1:4;
    coeff2=polyfit(A,T,2);
    J2(k)=sum((polyval(coeff2,A)-T).^2);
    end

mu2=mean(T);
for k=1:4;
    S2(k)=sum((T-mu2).^2);
    r2_2(k)=1-J2(k)/S2(k);
end
disp("The J value for the second degree polynomial fitting is:")
```

```
The J value for the second degree polynomial fitting is:
```

```
disp(J2)
```

```
   79.2894   79.2894   79.2894   79.2894
```

```
disp("The S value for the second degree polynomial fitting is:")
```

```
The S value for the second degree polynomial fitting is:
```

```
disp(S2)
```

```
   1.0e+03 *

    2.0361    2.0361    2.0361    2.0361
```

```
disp("The r^2 value for the second degree polynomial fitting is:")
```

```
The r^2 value for the second degree polynomial fitting is:
```

```
disp(r2_2)
```

```
    0.9611    0.9611    0.9611    0.9611
```

```
% third-degree polynomial
% use polyfit to obtain the third degree function coefficients
p3=polyfit(A,T,3)
```

```
p3 = 1×4
    0.0107    1.7611  -17.3522  131.0308
```

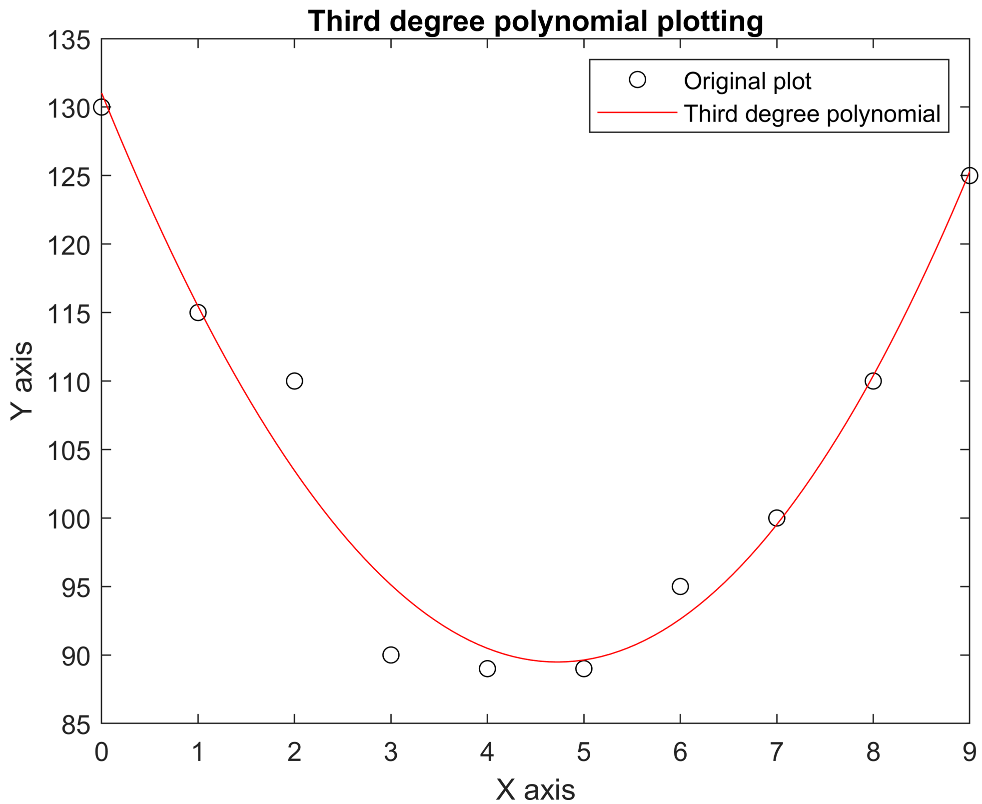
```
% use polyval to obtain y values when inputting A data to p3 third degree
% polynomial function
x3=linspace(min(A),max(A),100)
```

```
x3 = 1×100
         0    0.0909    0.1818    0.2727    0.3636    0.4545    0.5455    0.6364 ···
```

```
y3=polyval(p3,x3)
```

```
y3 = 1×100
  131.0308  129.4679  127.9341  126.4296  124.9543  123.5083  122.0916  120.7044 ···
```

```
%plot the original and the third degree polynomial data to analyze the
%plots
figure
plot(A,T,'ko')
hold on
plot(x3,y3,'r')
title("Third degree polynomial plotting")
xlabel("X axis")
ylabel("Y axis")
legend("Original plot","Third degree polynomial")
```




```
% calculate J, S, and r^2 using below formulas

for k=1:4;
    coeff3=polyfit(A,T,3);
    J3(k)=sum((polyval(coeff3,A)-T).^2);
    end

mu3=mean(T);
for k=1:4;
    S3(k)=sum((T-mu3).^2);
    r2_3(k)=1-J3(k)/S3(k);
end
disp("The J value for the third degree polynomial fitting is:")
```

```
The J value for the third degree polynomial fitting is:
```

```
disp(J3)
```

```
   78.9368   78.9368   78.9368   78.9368
```

```
disp("The S value for the third degree polynomial fitting is:")
```

```
The S value for the third degree polynomial fitting is:
```

```
disp(S3)
```

```
   1.0e+03 *
```

```
    2.0361    2.0361    2.0361    2.0361
```

```
disp("The r^2 value for the third degree polynomial fitting is:")
```

```
The r^2 value for the third degree polynomial fitting is:
```

```
disp(r2_3)
```

```
    0.9612    0.9612    0.9612    0.9612
```

```

% fourth-degree polynomial
% use polyfit to obtain the fourth degree function coefficients
p4=polyfit(A,T,4)
```

```
p4 = 1×5
   -0.0249    0.4591   -0.7551  -12.8679  129.9545
```

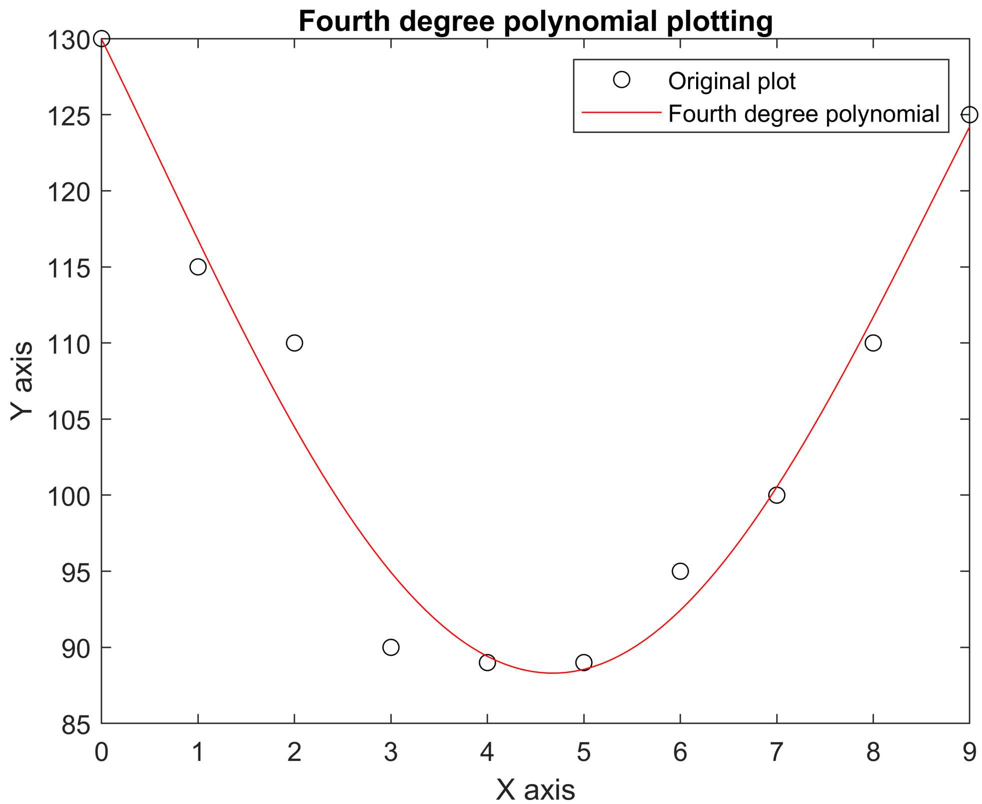
```
% use polyval to obtain y values when inputting A data to p4 fourth degree
% polynomial function
x4=linspace(min(A),max(A),100)
```

```
x4 = 1×100
         0    0.0909    0.1818    0.2727    0.3636    0.4545    0.5455    0.6364 ⋯
```

```
y4=polyval(p4,x4)
```

```
y4 = 1×100
  129.9545  128.7788  127.5927  126.3981  125.1971  123.9915  122.7833  121.5743 ⋯
```

```

%plot the original and the fourth degree polynomial data to analyze the
%plots
figure
plot(A,T,'ko')
hold on
plot(x4,y4,'r')
title("Fourth degree polynomial plotting")
xlabel("X axis")
ylabel("Y axis")
legend("Original plot","Fourth degree polynomial")
```




```
% calculate J, S, and r^2 using below formulas

for k=1:4;
    coeff4=polyfit(A,T,4);
    J4(k)=sum((polyval(coeff4,A)-T).^2);
    end

mu4=mean(T);
for k=1:4;
    S4(k)=sum((T-mu4).^2);
    r2_4(k)=1-J4(k)/S4(k);
end
disp("The J value for the fourth degree polynomial fitting is:")
```

```
The J value for the fourth degree polynomial fitting is:
```

```
disp(J4)
```

```
   68.7127   68.7127   68.7127   68.7127
```

```
disp("The S value for the fourth degree polynomial fitting is:")
```

```
The S value for the fourth degree polynomial fitting is:
```

```
disp(S4)
```

```
   1.0e+03 *
```

```
    2.0361    2.0361    2.0361    2.0361
```

```
disp("The r^2 value for the fourth degree polynomial fitting is:")
```

```
The r^2 value for the fourth degree polynomial fitting is:
```

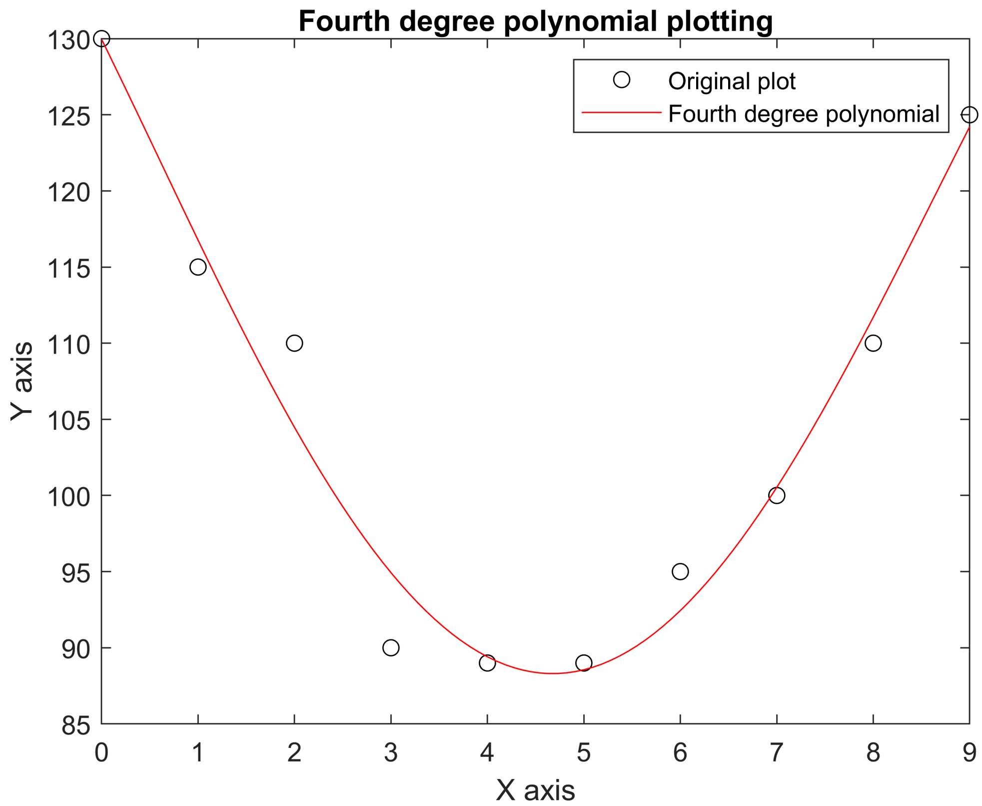
```
disp(r2_4)
```

```
    0.9663    0.9663    0.9663    0.9663
```

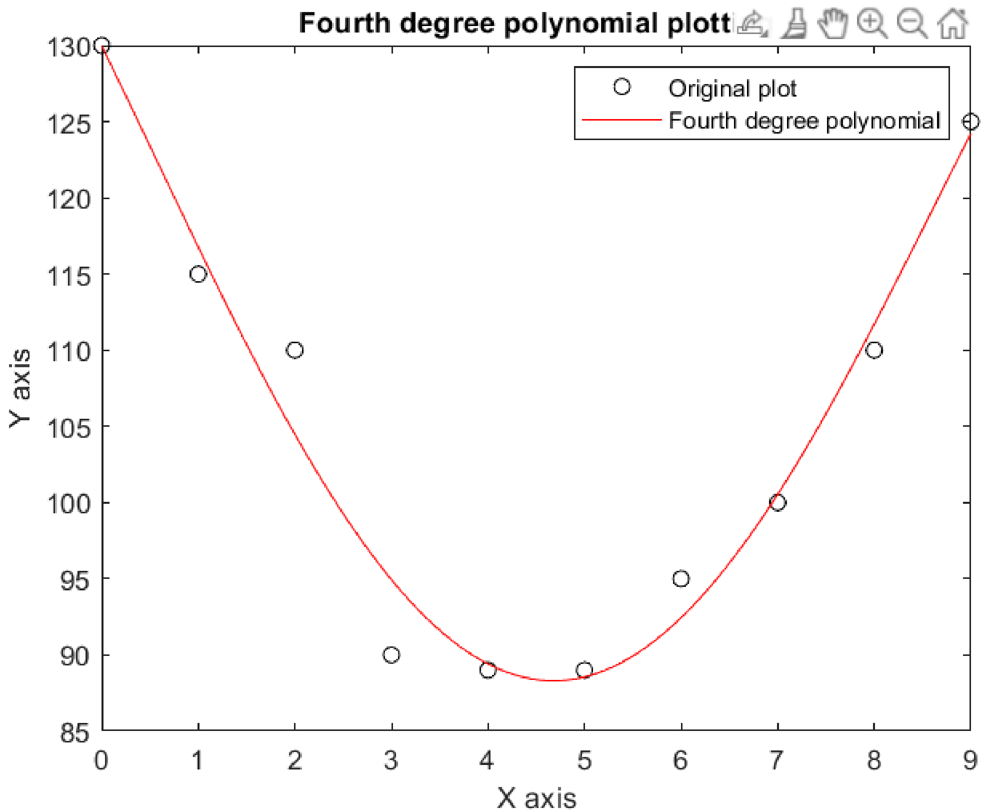
```
disp("After analyzing the r^2 values of first, second, third, and fourth degree polynomials, th
```

```
After analyzing the r^2 values of first, second, third, and fourth degree polynomials, the r^2 value of fourth polyn
```

```
%b: use the polynomial giving the best fit to estimate the amount of
%additive that minimizes the drying time

%plot the original and the fourth degree polynomial data to analyze the
%plots
figure
plot(A,T,'ko')
hold on
plot(x4,y4,'r')
title("Fourth degree polynomial plotting")
xlabel("X axis")
ylabel("Y axis")
legend("Original plot","Fourth degree polynomial")
```



```
%obtain the minimum additive A to minimize time T

[minA minT]=ginput(1)
```




```
minA = 4.6492
minT = 88.3119
```

```
% Question 3 (chapter 6 question 12)

% Delete all previous information
clear
clc
close all

% Description:The following representss pressure samplesm in pounds per
% square inch (psi), taken in a fuel line once every second for 10 sec.

% a : fit a first, second, third degree polynomial to these data, Plot the
% curve fits along with the data points.

% define x and y which are t and p, respectively
t=[1 2 3 4 5 6 7 8 9 10]
```

```
t = 1×10
     1     2     3     4     5     6     7     8     9    10
```

```
p=[26.1 27.0 28.2 29.0 29.8 30.6 31.1 31.3 31.0 30.5]
```

```
p = 1×10
   26.1000   27.0000   28.2000   29.0000   29.8000   30.6000   31.1000   31.3000 ⋯
```

```
% first-degree polynomial
% use polyfit to obtain the first degree function coefficients
p1=polyfit(t,p,1)
```

```
p1 = 1×2
    0.5467   26.4533
```

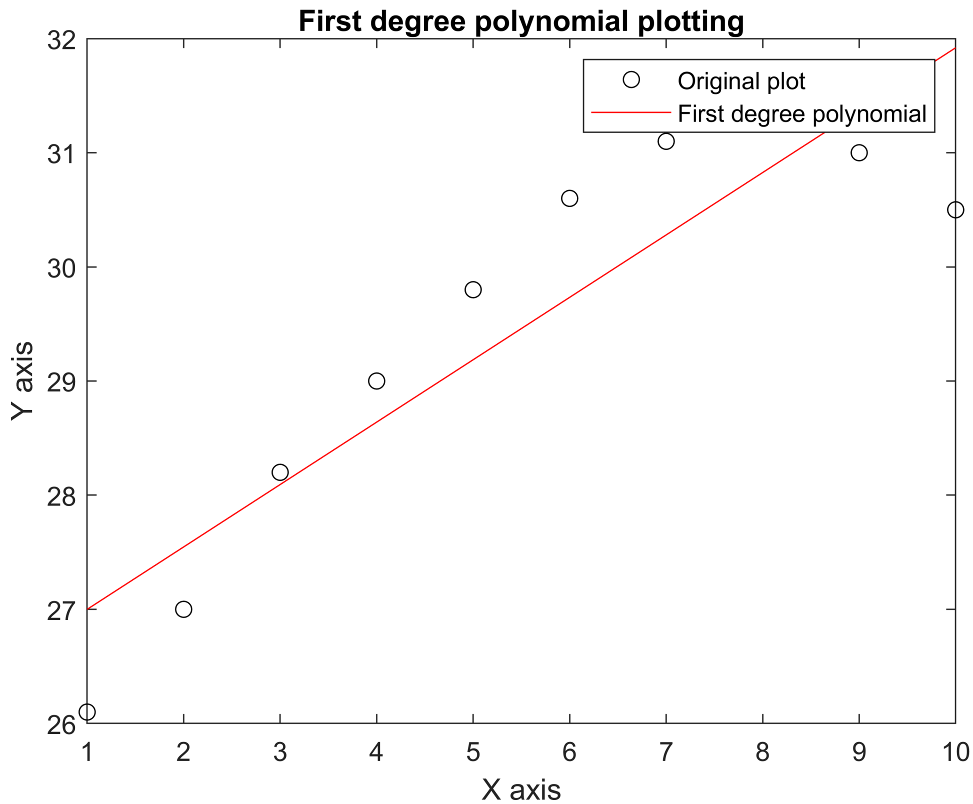
```
% use polyval to obtain y values when inputting A data to p1 first degree
% polynomial function
x1=linspace(min(t),max(t),100)
```

```
x1 = 1×100
    1.0000    1.0909    1.1818    1.2727    1.3636    1.4545    1.5455    1.6364 ⋯
```

```
y1=polyval(p1,x1)
```

```
y1 = 1×100
   27.0000   27.0497   27.0994   27.1491   27.1988   27.2485   27.2982   27.3479 ⋯
```

```
%plot the original and the first degree polynomial data to analyze the
%plots
figure
plot(t,p,'ko')
hold on
plot(x1,y1,'r')
title("First degree polynomial plotting")
xlabel("X axis")
ylabel("Y axis")
legend("Original plot","First degree polynomial")
```




```
% calculate J, S, and r^2 using below formulas

for k=1:4;
    coeff1=polyfit(t,p,1);
    J1(k)=sum((polyval(coeff1,t)-p).^2);
    end

mu1=mean(p);
for k=1:4;
    S1(k)=sum((p-mu1).^2);
    r2_1(k)=1-J1(k)/S1(k);
end
disp("The J value for the first degree polynomial fitting is:")
```

```
The J value for the first degree polynomial fitting is:
```

```
disp(J1)
```

```
    5.4293    5.4293    5.4293    5.4293
```

```
disp("The S value for the first degree polynomial fitting is:")
```

```
The S value for the first degree polynomial fitting is:
```

```
disp(S1)
```

```
   30.0840   30.0840   30.0840   30.0840
```

```
disp("The r^2 value for the first degree polynomial fitting is:")
```

```
The r^2 value for the first degree polynomial fitting is:
```

```
disp(r2_1)
```

```
    0.8195    0.8195    0.8195    0.8195
```

```



% second-degree polynomial
% use polyfit to obtain the second degree function coefficients
p2=polyfit(t,p,2)
```

```
p2 = 1×3
   -0.0977    1.6217   24.3033
```

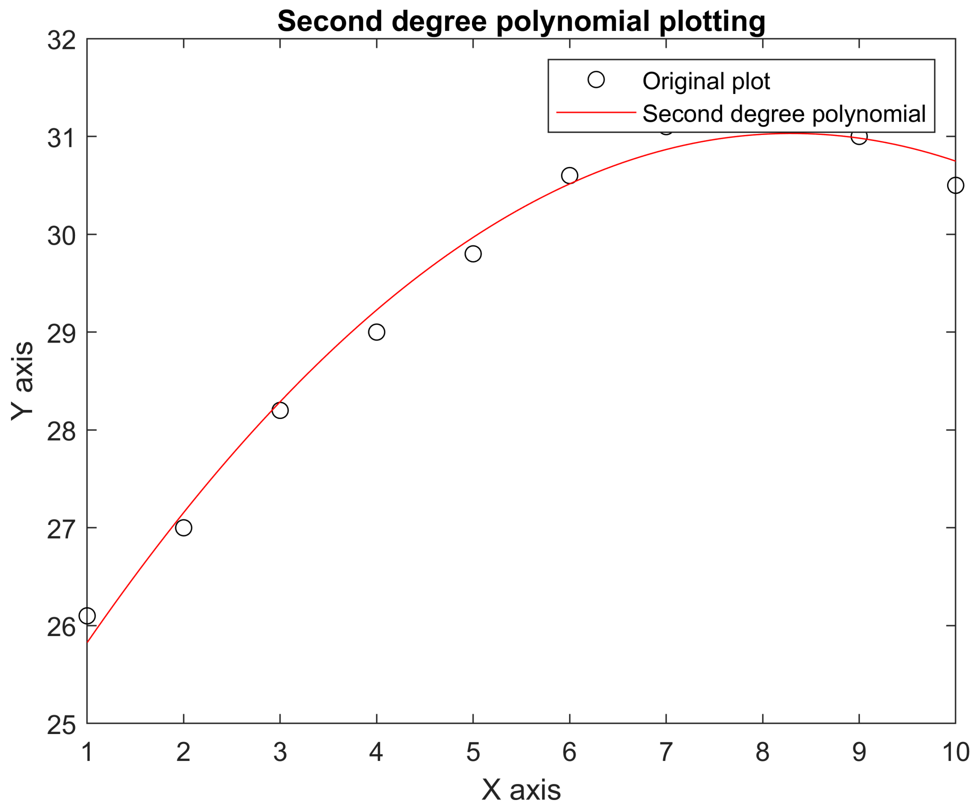
```
% use polyval to obtain y values when inputting A data to p2 second degree
% polynomial function
x2=linspace(min(t),max(t),100)
```

```
x2 = 1×100
    1.0000    1.0909    1.1818    1.2727    1.3636    1.4545    1.5455    1.6364 ···
```

```
y2=polyval(p2,x2)
```

```
y2 = 1×100
   25.8273   25.9561   26.0834   26.2090   26.3330   26.4554   26.5761   26.6953 ···
```

```

%plot the original and the second degree polynomial data to analyze the
%plots
figure
plot(t,p,'ko')
hold on
plot(x2,y2,'r')
title("Second degree polynomial plotting")
xlabel("X axis")
ylabel("Y axis")
legend("Original plot","Second degree polynomial")
```




```
% calculate J, S, and r^2 using below formulas

for k=1:4;
    coeff2=polyfit(t,p,2);
    J2(k)=sum((polyval(coeff2,t)-p).^2);
    end

mu2=mean(p);
for k=1:4;
    S2(k)=sum((p-mu2).^2);
    r2_2(k)=1-J2(k)/S2(k);
end
disp("The J value for the second degree polynomial fitting is:")
```

```
The J value for the second degree polynomial fitting is:
```

```
disp(J2)
```

```
    0.3866    0.3866    0.3866    0.3866
```

```
disp("The S value for the second degree polynomial fitting is:")
```

```
The S value for the second degree polynomial fitting is:
```

```
disp(S2)
```

```
   30.0840   30.0840   30.0840   30.0840
```

```
disp("The r^2 value for the second degree polynomial fitting is:")
```

```
The r^2 value for the second degree polynomial fitting is:
```

```
disp(r2_2)
```

```
    0.9871    0.9871    0.9871    0.9871
```

```
% third-degree polynomial
% use polyfit to obtain the third degree function coefficients
p3=polyfit(t,p,3)
```

```
p3 = 1×4
   -0.0106    0.0766    0.8175   25.2100
```

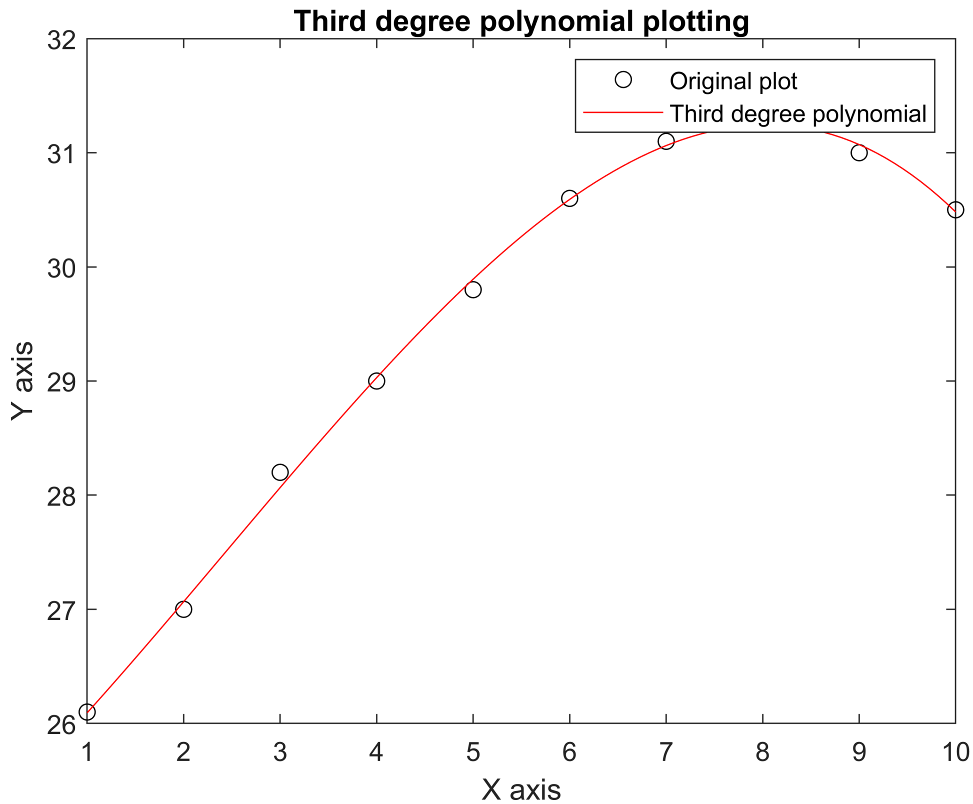
```
% use polyval to obtain y values when inputting A data to p3 third degree
% polynomial function
x3=linspace(min(t),max(t),100)
```

```
x3 = 1×100
    1.0000    1.0909    1.1818    1.2727    1.3636    1.4545    1.5455    1.6364 ⋯
```

```
y3=polyval(p3,x3)
```

```
y3 = 1×100
   26.0936   26.1793   26.2657   26.3528   26.4405   26.5287   26.6174   26.7066 ⋯
```

```
%plot the original and the first degree polynomial data to analyze the
%plots
figure
plot(t,p,'ko')
hold on
plot(x3,y3,'r')
title("Third degree polynomial plotting")
xlabel("X axis")
ylabel("Y axis")
legend("Original plot","Third degree polynomial")
```




```
% calculate J, S, and r^2 using below formulas

for k=1:4;
    coeff3=polyfit(t,p,3);
    J3(k)=sum((polyval(coeff3,t)-p).^2);
    end

mu3=mean(p);
for k=1:4;
    S3(k)=sum((p-mu3).^2);
    r2_3(k)=1-J3(k)/S3(k);
end
disp("The J value for the third degree polynomial fitting is:")
```

```
The J value for the third degree polynomial fitting is:
```

```
disp(J3)
```

```
    0.0417    0.0417    0.0417    0.0417
```

```
disp("The S value for the third degree polynomial fitting is:")
```

```
The S value for the third degree polynomial fitting is:
```

```
disp(S3)
```

```
   30.0840   30.0840   30.0840   30.0840
```

```
disp("The r^2 value for the third degree polynomial fitting is:")
```

```
The r^2 value for the third degree polynomial fitting is:
```

```
disp(r2_3)
```

```
    0.9986    0.9986    0.9986    0.9986
```

```


% b: use the results from part a to predict the pressure at t=11 sec.
% Explain which curve fit gives the most reliable prediction. Consider the
% coefficients of determination and the residuals for each fit in making
% your decision.

disp("The coefficient of determination of the first, second, and third degree polynomial are:")
```

```
The coefficient of determination of the first, second, and third degree polynomial are:
```

```
disp(r2_1)
```

```
    0.8195    0.8195    0.8195    0.8195
```

```
disp(r2_2)
```

```
    0.9871    0.9871    0.9871    0.9871
```

```
disp(r2_3)
```

```
    0.9986    0.9986    0.9986    0.9986
```

```

disp("As can be observed from the above coefficients of determination, third degree polynomials

% To find t=11, use polyval function with p3 third degree polynimal fitted
% function

t_11=polyval(p3,11)

disp("The pressure when t=11 sec is :")
disp(t_11)
```

```
% Question 4 (chapter 6 question 14)

% Description: the solubility of salt in water is a funciton of the water
% temperature. Let S represent the solubility of NaCL as grams of salt in
% 100 g of water. Let T be temperature in C. Use the following data to
% obtain a curve for S as a function of T. Usee the fit to estimate S when
% T=25 C

% delete all previous information and entries
clear
clc
close all
```

```
% Define x and y which are T and S, respectively
x=[10 20 30 40 50 60 70 80 90]
```

```
x = 1×9
    10    20    30    40    50    60    70    80    90
```

```
y=[35 35.6 36.25 36.9 37.5 38.1 38.8 39.4 40]
```

```
y = 1×9
   35.0000   35.6000   36.2500   36.9000   37.5000   38.1000   38.8000   39.4000 ···
```

```
% the following code will be used to figure out which degree polynomial
% fits the data best

% polyfit is used to find coefficients of first, second, third, fourth, and
% fifth degree polynomials that fits the original data
p1=polyfit(x,y,1)
```

```
p1 = 1×2
    0.0628   34.3639
```

```
p2=polyfit(x,y,2)
```

```
p2 = 1×3
   -0.0000    0.0633   34.3560
```

```
p3=polyfit(x,y,3)
```

```
p3 = 1×4
   -0.0000    0.0001    0.0603   34.3865
```

```
p4=polyfit(x,y,4)
```

```
p4 = 1×5
   -0.0000    0.0000   -0.0001    0.0657   34.3472
```

```
p5=polyfit(x,y,5)
```

```
p5 = 1×6
   -0.0000    0.0000   -0.0001    0.0023    0.0231   34.5917
```

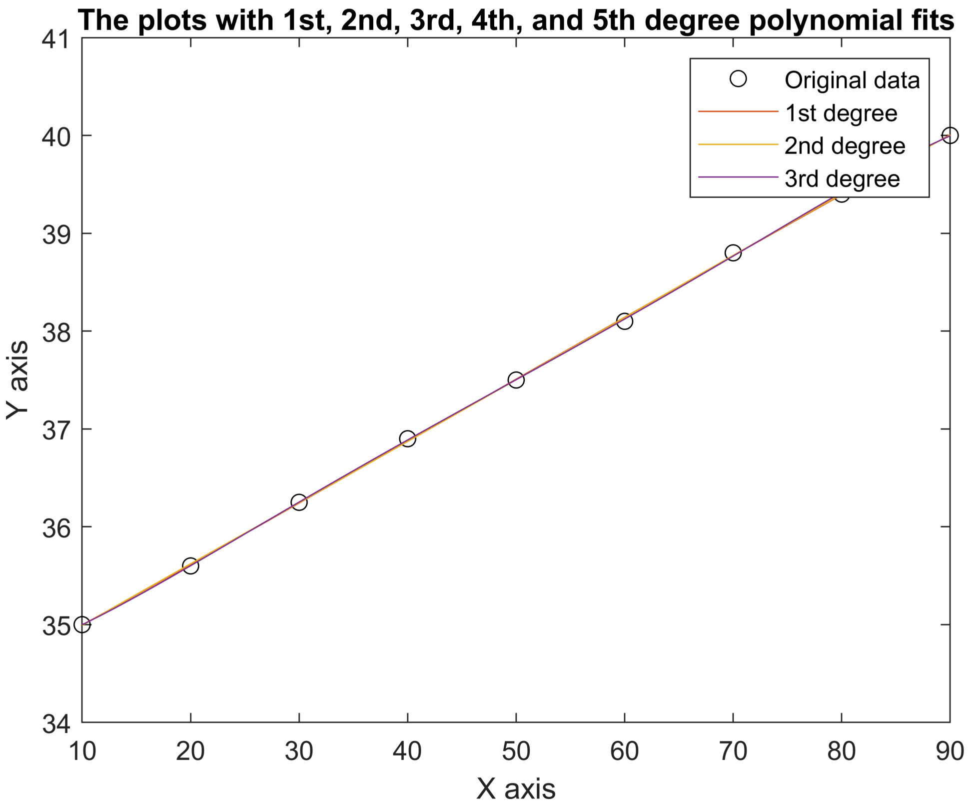
```
x1=linspace(min(x),max(x),100)
```

```
x1 = 1×100
   10.0000   10.8081   11.6162   12.4242   13.2323   14.0404   14.8485   15.6566 ···
```

```
% polyval will be used to plot x values on those specific functions

figure
plot(x,y,'ko')
% hold on
% plot(x1,polyval(p1,x1))
% hold on
% plot(x1,polyval(p2,x1))
hold on
plot(x1,polyval(p3,x1))
```

```
hold on
plot(x1,polyval(p4,x1))
hold on
plot(x1,polyval(p5,x1))
title("The plots with 1st, 2nd, 3rd, 4th, and 5th degree polynomial fits")
xlabel("X axis")
ylabel("Y axis")
legend("Original data","1st degree","2nd degree","3rd degree","4th degree","5th degree")
```

```
Warning: Ignoring extra legend entries.
```




```
% Obtain an equation describing the polynomial fit
p1=polyfit(x,y,1)
```

```
p1 = 1×2
    0.0628   34.3639
```

```
m=p1(1)
```

```
m = 0.0628
```

```
b=p1(2)
```

```
b = 34.3639
```

```
ans=['After observing the plot of 5 polynomial fits, it is obvious that the data fits the first
```

```
ans =
'After observing the plot of 5 polynomial fits, it is obvious that the data fits the first degree polynomial and wil
```

```
disp(ans)
```

After observing the plot of 5 polynomial fits, it is obvious that the data fits the first degree polynomial and will

```
% find S when T=25 which is finding y when x=25
S=polyval(p1,25)

ansb=['The solubility of NaCl described by S when T is 25 C is ',num2str(S)]
disp(ansb)
```

```
% Question 5 (chapter 6 question 16)

% Description: The following function is linear in the parameters a1 and
% a2: y(x)=a1+a2*ln(x). Use the least squares regression with the following
% data to estimate the values of a1 and a2. Use the curve fit to estimate
% the values of y at x=2.5 and at x=11.

% delete all the previous data and entries
clear
clc
close all

% define x and y

x=[1 2 3 4 5 6 7 8 9 10]
```

```
x = 1×10
     1     2     3     4     5     6     7     8     9    10
```

```
y=[10 14 16 18 19 20 21 22 23 23]
```

```
y = 1×10
    10    14    16    18    19    20    21    22    23    23
```

```
% Least squares method from textbook. Important to remember that x=ln(x)

% create symbolic variables a and b
% a is a1 and b is a2
syms a b
% find the J function and its sum
J=(a+b*log(x)-y).^2
```

J =

$$\left((a-10)^2 \quad \left(a+\frac{6243314768165359\,b}{9007199254740992}-14\right)^2 \quad \left(a+\frac{2473854946935173\,b}{2251799813685248}-16\right)^2 \quad \left(a+\frac{62433147681}{4503599627}\right.\right.$$

```
J=sum(J)
```

J =

$$\left(a+\frac{2473854946935173\,b}{2251799813685248}-16\right)^2+(a-10)^2+\left(a+\frac{4682486076124019\,b}{2251799813685248}-22\right)^2+\left(a+\frac{49477098938}{22517998136}\right.$$

```
% find derivatives with respect to a1 and a2
a1=diff(J,a)
```

a1 =

$$20\,a+\frac{1360484536715062 49\,b}{4503599627370496}-372$$

```
a2=diff(J,b)
```

a2 =

$$\frac{13604845367150624 9\,a}{4503599627370496}+\frac{22432542821490860975088303308704 13\,b}{40564819207303340847894502572032}-\frac{1390518427306330813}{2251799813685248}$$

```
% set the integer value to 4 for each value
a1=vpa(a1,4)
```

a1 = $20.0\,a+30.21\,b-372.0$

```
a2=vpa(a2,4)
```

a2 = $30.21\,a+55.3\,b-617.5$

```
% solve the equations for a and b by setting them equal to 0
eqns=[a1==0, a2==0];
S=solve(eqns,[a b])
```

```
S = struct with fields:
    a: [1×1 sym]
    b: [1×1 sym]
```

```
% obtain a and b values which are a and b respectively
a=vpa(S.a,4)
```

a = 9.912

```
b=vpa(S.b,4)
```

b = 5.752

```
disp("The a1 was calculated to be: ")
```

The a1 was calculated to be:

```
disp(a)
```

9.912

```
disp("The a2 was calculated to be: ")
```

```
The a2 was calculated to be:
```

```
disp(b)
```

5.752

```
% Obtain y values when x=2.5 and x=11
% Important note: x=log(x)
x_25=2.5
```

```
x_25 = 2.5000
```

```
y1_25=a+b*log(x_25)
```

y1_25 = 15.182587127809158758257413511116

```
x_11=11
```

```
x_11 = 11
```

```
y1_11=a+b*log(x_11)
```

y1_11 = 23.704424966734827324262951439749

```
% Least squares regression method with polyfit which provides coefficients
% for first degree polynomial equation

p1=polyfit(log(x),y,1)
```

```
p1 = 1×2
    5.7518    9.9123
```

```
% Obtain values for the x=2.5 and x=11
y2_25=polyval(p1,log(2.5))
```

```
y2_25 = 15.1826
```

```
y2_11=polyval(p1,log(11))
```

```
y2_11 = 23.7044
```

```
% Compare the equations and the final y values when x=2.5 and x=11
x1=['Using the least squres method from the textbook, the following equation was obtained: y=',
```

```
x1 =
'Using the least squres method from the textbook, the following equation was obtained: y=9.9123+5.7518*ln(x). The y
```

```
disp(x1)
```

Using the least squres method from the textbook, the following equation was obtained: y=9.9123+5.7518*ln(x). The y v

```
x2=['Using polyfit which utilizes least square regression method, the following equation was ob
```

x2 =
'Using polyfit which utilizes least square regression method, the following equation was obtained: y=9.9123+5.7518*l

```
disp(x2)
```

Using polyfit which utilizes least square regression method, the following equation was obtained: y=9.9123+5.7518*ln

Hisffshjfhjdhfkjshdf dsfjshdjfhsjlks dshfjfsl 9.91 or 5.752