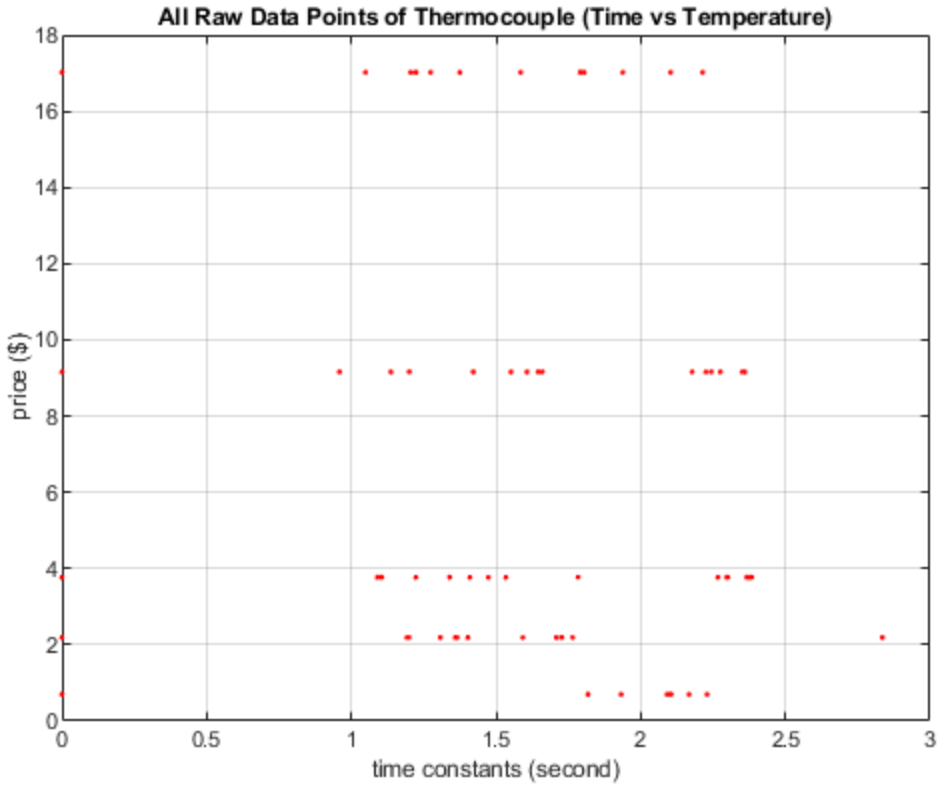
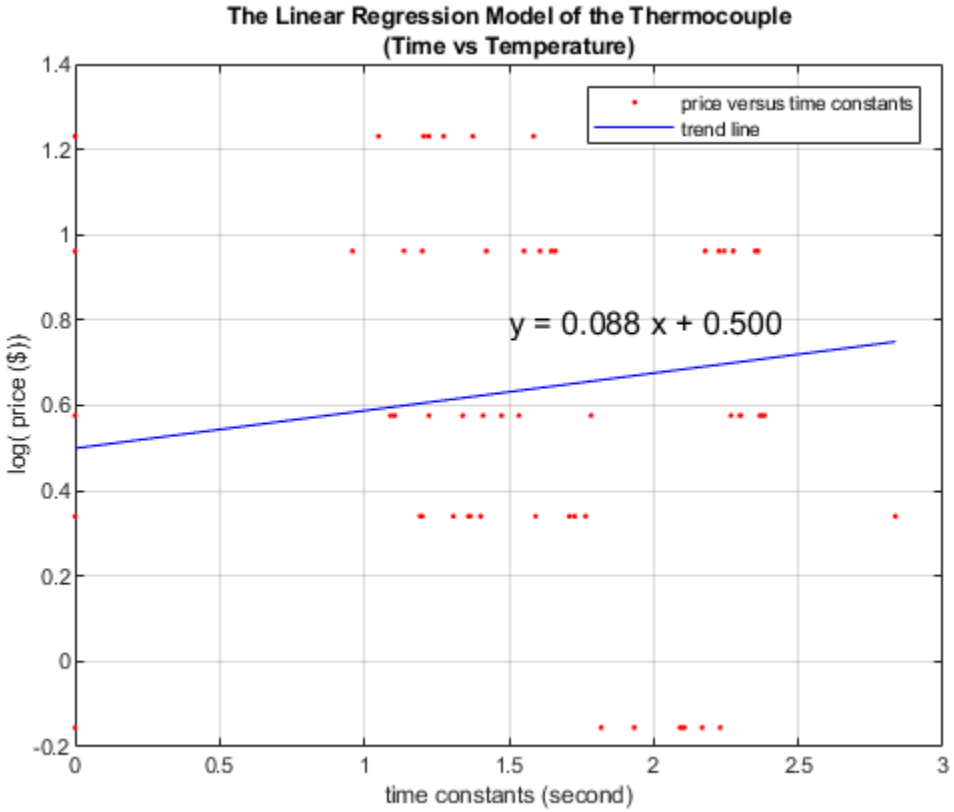
```
function Project_M4Regression_002_08(tau)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENGR 132 FINAL PROJECT
%
% Program Description
% This user-defined function can make a regression plot of data and
 calculate sse, sst, and
% rsquare
%
% Function Call
% function Project_M4Regression_002_08(tau)
%
% Input Arguments
% 1. tau: vector include all tau value
%
% Output Arguments
% none
% Assignment Information
%   Assignment:        M4
%   Author:             Tomoki Koike, koike@purdue.edu,
%                       Yi Zhou   zhou823@purdue.edu
%                       Eu JIn Lee  lee2219@purdue.edu
%                       Ian Pitman  ipitman@purdue.edu
%   Team ID:            002-08
%   Contributor:   no contributor
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%We give different model number corresponding price
fos1 = 17.02;
fos2 = 9.16;
fos3 = 3.77;
fos4 = 2.19;
fos5 = 0.7;

%We create a matrix vec and each element in vec repeat 10 times.
vec = [fos1 fos2 fos3 fos4 fos5 fos1 fos2 fos3 fos4 fos5];
pricevec = repelem(vec,10);

% this figure is showing all raw data points of thermocouple.
figure
plot(tau, pricevec,'.r');
xlabel('time constants (second)');
ylabel('price ($)');
title('All Raw Data Points of Thermocouple (Time vs Temperature)');
set(gca,'fontsize',8);
grid;

%linearization of data--------------------------------
%We already find that function type best models this relationship is
%exponential, next we will do linear regression.

%this figure is showing linear regression model of thermocouple
```

```
figure
logpricevec = log10(pricevec);
plot(tau,logpricevec,'.r');
xlabel('time constants (second)');
ylabel('log( price ($))');
title({'The Linear Regression Model of the Thermocouple','(Time vs
 Temperature)'});
set(gca,'fontsize',8);
grid on;

%this is going to get the trendline
pfittau = polyfit(tau,logpricevec,1);
tauslope = pfittau(1);
tauintercept = pfittau(2);
predicty = tauslope .* tau + tauintercept;
hold on
plot(tau,predicty,"-b")
legend("price versus time constants","trend
 line",'Location','northeast')
theString = sprintf('y = %.3f x + %.3f', tauslope, tauintercept);
  %dispaly fit equation on plot
text(1.5, 0.8, theString, 'FontSize', 12)
hold off

%calculate the SSE, SST, and r^2 of regression model
errorterm = pricevec - predicty;      %the error of y data and
 predicted y data
SSE = sum(errorterm);            %get SSE by adding all error square
 together
meanY = mean(pricevec);               %get average y value
ydatadeviation = pricevec - meanY;    %y data deviation
ydatadeviationsq = ydatadeviation.^2;   %square of y data deviation
SST = sum(ydatadeviationsq);      %get SST by adding square of
 deviation together
rsquare = 1 - (SSE/SST);
```


All Raw Data Points of Thermocouple (Time vs Temperature)
price ($)
time constants (second)
0
2
4
6
8
10
12
14
16
18
0
0.5
1
1.5
2
2.5
3


The Linear Regression Model of the Thermocouple
(Time vs Temperature)
price versus time constants
trend line
y = 0.088 x + 0.500
log(price ($))
time constants (second)
-0.2
0
0.2
0.4
0.6
0.8
1
1.2
1.4
0
0.5
1
1.5
2
2.5
3

Published with MATLAB® R2018a