Table of Contents


```
function [SSE_final, SST] = M3_Regression_001_30

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENGR 132
% Program Description
% The program finds a model for estimating price of enzyme according
 to
% their Km value and output the estimate price.
%
% Function Call
% function [price] = price_estimate(Km)
%
% Input Arguments
% Km: the Michaelis Constant of the enzyme
%
% Output Arguments
% price_estimated: the estimate price for the enzyme
%
% Assignment Information
%   Assignment:     M3
%   Team Mmeber:    Luming Lin, lin971@purdue.edu
%                   Surya Manikhandan, smanikha@purdue.edu
%                   Julius Mesa, jmesa@purdue.edu
%                   Alex Norkus, anorkus@purdue.edu
%   Team ID:        001-30
%   Academic Integrity:
%     [] We worked with one or more peers but our collaboration
%        maintained academic integrity.
%     Peers we worked with: Name, login@purdue [repeat for each]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

INITIALIZATION

```
clc;clearvars;
data = readmatrix('Data_NovelEnzymes_priceCatalog');
price_measured = data(:,2);
Km_measured = data(:,1);
```

CALCULATIONS

```
SST = sum((price_measured - mean(price_measured)).^2);
```

for linear

```
coe1 = polyfit(Km_measured,price_measured,1);
slope_m1 = coe1(1);
b1 = coe1(2);
predict1 = polyval(coe1,Km_measured);
SSE1 = sum((price_measured - predict1).^2);
r2_1 = 1-(SSE1/SST);
```

for power

```
coe2 = polyfit(log10(Km_measured),log10(price_measured),1);
m2 = coe2(1);
b2 = 10 ^ coe2(2);
predict2 = b2 .* (Km_measured) .^ m2;
SSE2 = sum((price_measured - predict2).^2);
r2_2 = 1-(SSE2/SST);
```

for exponential

```
coe3 = polyfit(Km_measured,log10(price_measured),1);
m3 = coe3(1);
b3 = 10 ^ coe3(2);
predict3 = b3 .* 10 .^ (Km_measured .* m3);
SSE3 = sum((price_measured - predict3).^2);
r2_3 = 1-(SSE3/SST);
```

for Logarithmic

```
coe4 = polyfit(log(Km_measured),price_measured,1);
m4 = coe4(1);
b4 = coe4(2);
predict4 = m4 .* log10(Km_measured) + b4;
SSE4 = sum((price_measured - predict4).^2);
```

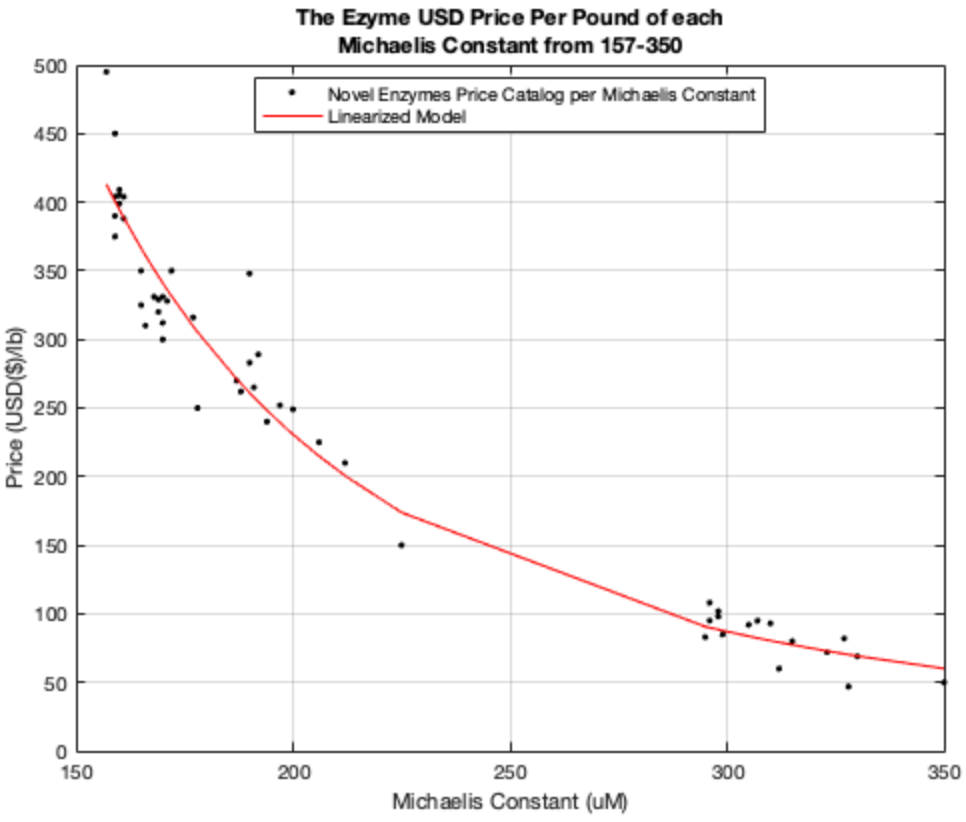
```
r2_4 = 1-(SSE4/SST);
```

FORMATTED TEXT/FIGURE DISPLAYS

```
SSE_value = [SSE1,SSE2,SSE3,SSE4];
SSE_final = min(SSE_value);
choose = find(SSE_value == SSE_final);

if choose == 1
r2 = r2_1;
plot(Km_measured, price_measured,'k.');
hold on;
plot(Km_measured, predict1,'r-');
title({'The Ezyme USD Price Per Pound of each';'Michaelis Constant
 from 157-350'})
xlabel('Michaelis Constant (uM)');
ylabel('Price (USD($)/lb)')
grid on;

elseif choose == 2
r2 = r2_2;
plot(Km_measured, price_measured,'k.')
hold on;
plot(Km_measured, predict2,'r-');
title({'The Ezyme USD Price Per Pound of each';'Michaelis Constant
 from 157-350'})
xlabel('Michaelis Constant (uM)');
ylabel('Price (USD($)/lb)')
legend('Novel Enzymes Price Catalog per Michaelis
 Constant', 'Linearized Model', 'location', 'best')
grid on


elseif choose == 3
    r2 = r2_3;
plot(Km_measured, price_measured,'k.')
hold on;
plot(Km_measured, predict3,'r-');
title({'The Ezyme USD Price Per Pound of each';'Michaelis Constant
 from 157-350'})
xlabel('Michaelis Constant (uM)');
ylabel('Price (USD($)/lb)')
grid on


elseif choose == 4
    r2 = r2_4;
plot(Km_measured, price_measured,'k.')
hold on;
plot(Km_measured, predict4,'r-');
```

```
title({'The Ezyme USD Price Per Pound of each';'Michaelis Constant
 from 157-350'})
xlabel('Michaelis Constant (uM)');
ylabel('Price (USD($)/lb)')
grid on
sgtitle('Data on Various Scaled Plots')

end
```

```
ans =

   3.5496e+04
```



COMMAND WINDOW OUTPUT

ACADEMIC INTEGRITY STATEMENT

We have not used source code obtained from any other unauthorized source, either modified or unmodified. Neither have we provided access to my code to another. The function we are submitting is our own original work.

Published with MATLAB® R2019a