Contents

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENGR 133
% Program Description
% Plot a data set and find the line of best fit and r-squared value
%
% Assignment Information
%   Assignment:     Ma5_Task3
%   Team ID:        LC1-04
%       Contributor(s): Nolan Hays, haysn@purdue.edu
%                   Jackson Bitterolf, jbittero@purdue.edu
%                   Ayush Viswanathan, viswan11@purdue.edu
%                   Roshan Sundar, rmsundar@purdue.edu
%   My contributor(s) helped me:
%     [ ] understand the assignment expectations without
%         telling me how they will approach it.
%     [ ] understand different ways to think about a solution
%         without helping me plan my solution.
%     [ ] think through the meaning of a specific error or
%         bug present in my code without looking at my code.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

INITIALIZATION

```
% Import the data from the text file
data = load('extraction_data.txt');

% Separate the values into x and y arrays
x_values = data(:,2);
y_values = data(:,1);

% Sort the x and y values
x_sorted = sort(x_values);
y_sort = sort(y_values);

% Initialize SSE and SST
SSE = 0;
SST = 0;
```

CALCULATIONS

```
% Find the slope and y-intercept of the dataset
p = polyfit(x_values, y_values, 1);

% Calculate the predicted y-values for line of best fit
predicted_y = polyval(p, x_sorted);

% Find extraction efficiency at 35 minutes
ee = polyval(p, 35);

% Calculate SSE
for I = 1:length(y_values)
    num = (y_values(I) - polyval(p, x_values(I))) ^ 2;
    SSE = SSE + num;
end

% Calculate SST
for J = 1:length(y_values)
    num = (y_sort(J) - mean(y_values)) ^ 2;
    SST = SST + num;
end

% Find r^2 by taking 1 minus the product of SSE and SST
r_sqr = (1 - (SSE/SST));
```

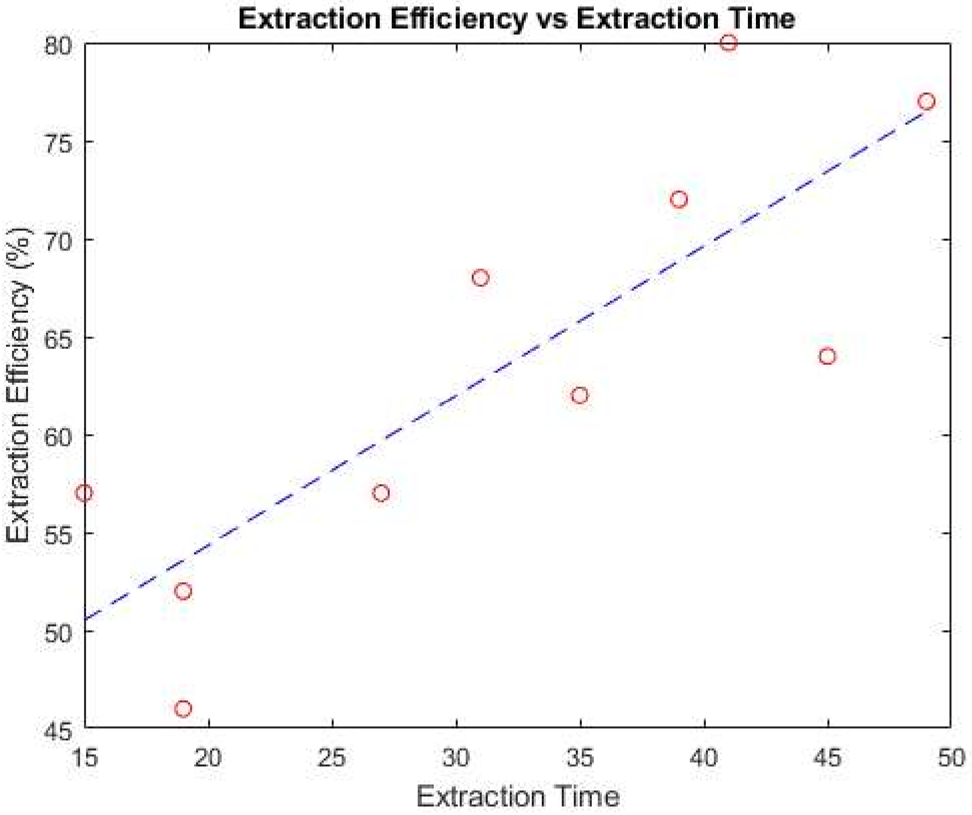
OUTPUTS

```
% Plot the original points
plot(x_values, y_values,'ro')
title('Extraction Efficiency vs Extraction Time')
xlabel('Extraction Time')
ylabel('Extraction Efficiency (%)')
hold on

% Plot the line of best fit
plot(x_sorted, predicted_y, 'b--')
hold off

% Print the results
fprintf('The best fit line is y = %fx + %f\n', p(1), p(2))
fprintf('The extraction efficiency is %f percent at 35 minutes.\n', ee)
fprintf('r^2 is %f\n', r_sqr)
```

```
The best fit line is y = 0.764000x + 39.052000
The extraction efficiency is 65.792000 percent at 35 minutes.
r^2 is 0.680298
```



ACADEMIC INTEGRITY STATEMENT

I have not used source code obtained from any other unauthorized source, either modified or unmodified. Neither have I provided access to my code to another. The project I am submitting is my own original work.

Published with MATLAB® R2020b