```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CODE CHALLENGE 4 - Linear Least-Squares Fit
%
% The purpose of this program is to calculate the equation of the best fit
% line for a data set using linear least-squares fitting.
%
% To complete the challenge, finish the code below to:
% 1) load data from csv file
% 2) find linear best fit coefficients and associated uncertainty
% 3) plot the original data along with the best fit line
% 4) add errorbars for fit uncertainty to this plot from the data and from
%    the linear regression parameters
%
% NOTE: DO NOT change any variable names already present in the code.
%
% Upload your team's script to Gradescope when complete.
%
% NAME YOUR FILE AS Challenge4_Sec{section number}_Group{group breakout #}.m
% ***Section numbers are 1 or 2***
% EX File Name: Challenge4_Sec1_Group15.m
%
% STUDENT TEAMMATES
% 1) Austin Sommars
% 2) Hannah Obolsky
% 3) Brendan Sheets
% 4) Issac Esquivel
% 5) Andrew Logue

% T & V uncertainty = .1 = sigma y
%{
Mars: 3.72076 [m/s^2]
Moon: 1.625   [m/s^2]
Mercury: 3.7  [m/s^2]
Uranus: 8.87  [m/s^2]
%}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Housekeeping (Please don't "clear all" or "clearvars", it makes grading difficult)

```
close all   % Close all open figure windows
clc           % Clear the command window
```

Load and extract the time and velocity vectors from the data

```
data = readtable('Challenge4_data.csv');
```

```
Warning: Column headers from the file were modified to make them valid MATLAB identifiers before creating
 variable names for the table. The original column headers are saved in the VariableDescriptions property.
Set 'PreserveVariableNames' to true to use the original column headers as table variable names.
```

```
t = data{:,1};   % [s]
```

```
v =data{:,2};    % [m/s]
```

Calculations

Find number of data points in the vectors

```
%Number of time intervals from table
N = length(t);

% Find linear best fit coefficients A and B
% Create H matrix
H = [(ones(length(v),1)),t];

% Create y matrix
y = [v]';
sigma_v = .1;

% Create W matrix (hint: type <help diag> in command line)
%delta_v is a vector of sigma_v values with a length of N
delta_v(1:N) = sigma_v;
Diagonal= 1 ./ (delta_v .* delta_v);
W = diag(Diagonal);

% Solve for P matrix
P = inv((H' * W * H));


% Solve for x_hat matrix and extract A and B parameters
x_hat = ((H' * H)^-1) * H' * v;
A = x_hat(1);
B = x_hat(2);

% extract uncertainty in A and uncertainty in B from P matrix
A_error = sqrt(P(1,1)); %Velocity
B_error = sqrt(P(2,2)); %Acceleration
```

Display acceleration with associated uncertainty and the intial velocity with associated uncertainty

Make sure to use and display with CORRECT SIG FIGS

```
Acceleration = round(B,2);
fprintf("Acceleration = %.2f ± %.2f \n", Acceleration, round(B_error, 1,'significant'))
```

```
Acceleration = -1.52 ± 0.05
```

```
InitialVel = round(A,2);
fprintf("Initial Velocity = %.2f ± %.2f \n", InitialVel, round(A_error, 1,'significant'))
```

```
Initial Velocity = 8.24 ± 0.30
```

Find predicted velocity values using your linear fit equation

```
%Estimated Velocity values over time
```

```
v_predicted =  A + B .* t;
```

Ploting and Error Calculations

On the same plot, do the following: 1. plot the velocity data vs time as a scatter plot 2. plot predicted velocity vs time as a line 3. title your plot so that it indicates what celestial body this data simulates 4. Add measured velocity error bars and predicted velocity error bars to the plot (hint - this will involve error propagation calculations)

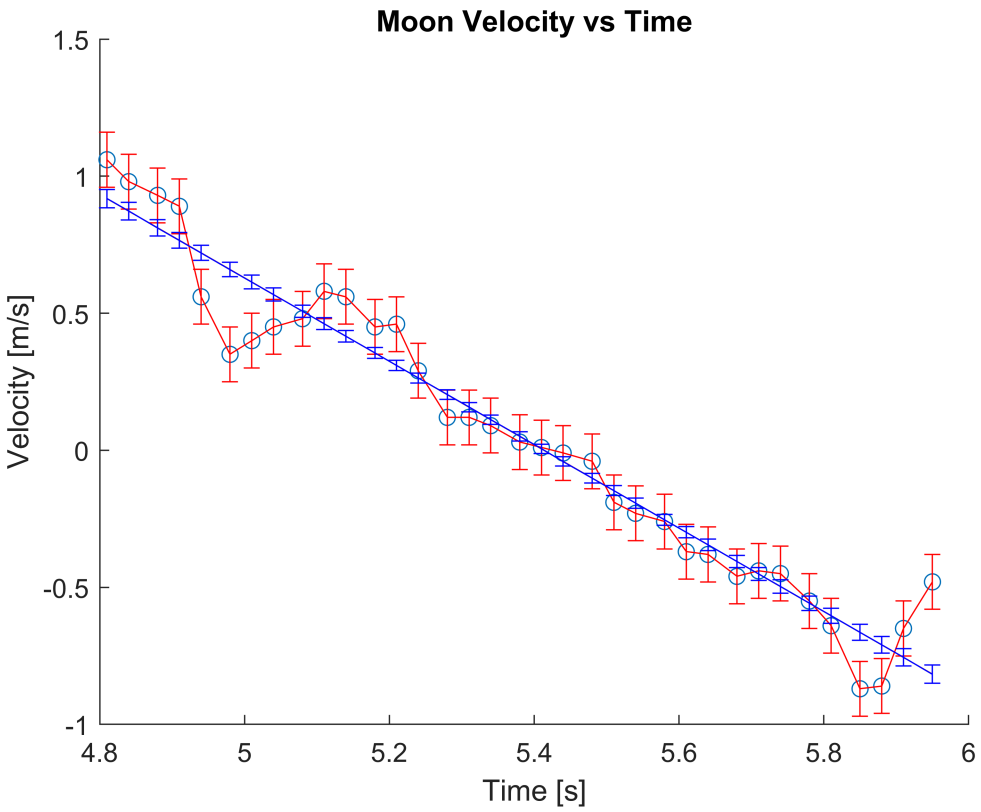
```
hold on
scatter(t,v)
plot(t,v_predicted)

%Our value for B is 1.5216 which matches the acceleration on the moon most
title('Moon Velocity vs Time')
xlabel('Time [s]')
ylabel('Velocity [m/s]')

v_err = .1; %Given uncertainty (Red error bars)
v_predicted_error = sqrt(diag(H*P*H')); %Predicted uncertainty from best fit line (Blue error b

e=(ones(length(v),1)).*v_err; %Error values vector of v_err values
e2=(ones(length(v),1)).*v_predicted_error; %Error values vector of v_predicted_error values
errorbar(t,v,e,"Color",'Red');
errorbar(t,v_predicted,e2,'color', 'Blue');

hold off
```


Moon Velocity vs Time
Velocity [m/s]
Time [s]
1.5
1
0.5
0
-0.5
-1
4.8
5
5.2
5.4
5.6
5.8
6