```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%
% 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) Zak Reichenbach
% 2) Ella Mumolo
% 3) Bart Kubiak
% 4) Anna Z Miecznik
% 5) Andrew Miller
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%
```

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

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

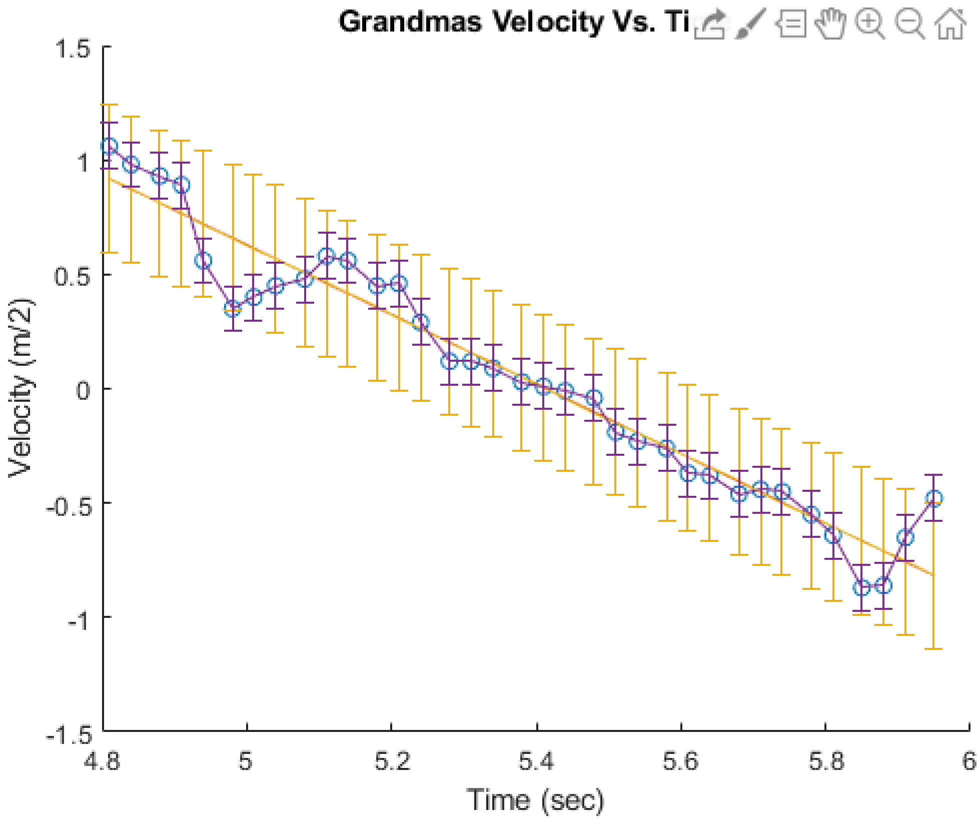
Load and extract the time and velocity vectors from the data

```
data = readtable('Challenge4_data.csv');
t = table2array(data(:,1));    % [s]
v = table2array(data(:,2));  % [m/s]
%
% %% Calculations
```

```
% % Find number of data points in the vectors
N = length(v);
%
% % Find linear best fit coefficients A and B
% % Create H matrix
% H =[ones(length(y))1,x]
H = [t,ones(length(v),1)];
%
% % Create y matrix
%
y = v;

Sigma_y = 0.1;
%
% % Create W matrix (hint: type <help diag> in command line)
 W = eye(length(y)).* (1/Sigma_y^2);
% W =
%
% % Solve for P matrix
 P =(H' * W *H)^-1;
%
% % Solve for x_hat matrix and extract A and B parameters
%X_hat = ((H' * H)^-1) * H' * Y
 x_hat = ((H' * H)^-1) * H' * y;
 A = x_hat(2);
 B = x_hat(1);
%

% % % % %
% % % % % % %  %deviation = y - (A + B.*x)
% % % % % % %  %Sigma_y = sqrt()1/length(y)-length(X_hat))) *
 sum(Deviation * Deviation)
% % % % % % %  %delta_y (1:length(y)) = Sigma_y;
% % % % % % %
% % % % % % % %  Diagonal =  ./ (delta_y .* delta_y);
% % % % % % % %  W = diag(Diagonal);
% % % % % % % %
% % % % % % % %  Error Covariance matrix
% % % % % % % %  P = (H' * W *H)^-1;
% % % % % % % %  Uncertainty in Parameter A
% % % % % % % %  Sigma_A = sqrt(P(1,1));
% % % % % % %  %
% % % % % % %
% % % % % % % %  X_hat_1 = ((H' * W * H)^-1) *H' * W * y;
% % % % % % % %
% % % % % % % %  y_fit = A +B.*x;
% % % % % % %
% % % % % %%Next two lines do the same thing
% % % % % % %  %Y_0 = A + B + 0; (Intercept of y)
% % % % %
% % % % % %%% y_0_new = [1 0 ] * X_hat;
% % % % %
% % % % % %% subset of H to look at the projection we are makeing
 (x=0)
```

```
% % % % %  %%% Sigma_Y0 = sqrt([1 0] * P * [1;0]);
% % % % %
% % extract uncertainty in A and uncertainty in B from P matrix
 A_error = sqrt(P(1,1));
 B_error = sqrt(P(2,2));
%
% %% Display acceleration with associated uncertainty and the intial
 velocity with associated uncertainty
% %  Make sure to use and display with CORRECT SIG FIGS
 fprintf('Our acceleration is %.2f and our uncertainty is %.2f.
\n',B,B_error);
 fprintf('Our initial velocity is %.2f and our uncertainty is %.2f
 \n',v_predicted(1),v_predicted_error(1))
%
%
% %% Find predicted velocity values using your linear fit equation
 v_predicted = A + B.*t;
%

x_values = H(:,1);
% %% Ploting and Error Calculations
% % On the same plot, do the following:
% % 1. plot the velocity data vs time as a scatter plot
scatter (x_values,y)
hold on
% % 2. plot predicted velocity vs time as a line
plot(x_values,v_predicted)
% % 3. title your plot so that it indicates what celestial body this
 data
% %    simulates
title('Grandmas Velocity Vs. Time')
xlabel('Time (sec)')
ylabel('Velocity (m/2)')
% % 4. Add measured velocity error bars and predicted velocity error
 bars to
% %    the plot (hint - this will involve error propagation
 calculations

v_err = ones(35,1).* Sigma_y;
v_predicted_error = ones(35,1).* (A_error + B_error);

errorbar(x_values,v_predicted,v_predicted_error)
errorbar(x_values,v,v_err)



%T&V uncertainty = 0.1 (W = 1/sigma_y^2)
```

```
Warning: Table variable names were modified to make them valid MATLAB
identifiers. The original names are saved in the VariableDescriptions
 property.
Our acceleration is -1.52 and our uncertainty is 0.27.
Our initial velocity is 0.92 and our uncertainty is 0.32
```



Published with MATLAB® R2019a