```
% J Hundley
% Lab06
% October 9, 2011
%{
Linear Equation and Regression Correlation Coefficients
Measures the strength of the linear relationship between y and x.

  Edwin Hubble used the Mount Wilson Observatory telescopes to measure
features of nebulae outside the Milky Way. He found that there is a
relationship between a nebula's distance from earth and the velocity
with which it was traveling from the earth. Hubble's initial data on
24 nebula is presented in Table 1 in the scenario document.
  The relationship between distance and velocity led scientists to propose
that the universe came into being with a Big Bang, a long time ago.
If material scattered from the point of the Big Bang traveling at a
constant velocity, the distance traveled can be determined.

- Ask the user to enter several pairs of velocity and distance and save them
in two vectors using a counter to determine the index of the vector element
- Use least-squares to find the linear equation [y = mx + b] that fits
the velocity and distance data.

- Velocity will be the independent variable(x) variable and
distance dependent variable(y)
- Use fprintf() to print the the linear model equation
- Use linear equation to find the distance given a velocity, interactively
- Print the enter velocity and distance in a sentence with units,
two decimal places, and no extra spaces.
%}
clc
clear all

%*****INITIALIZE*****
% initialize counter and accumulators
sumX     = 0;  % sum of all x (velocity)
sumY     = 0;  % sum of all y (distance)
sumXX    = 0;  % sum of x^2   (velocity)
sumXY    = 0;  % sum of x*y   (velocity*distance)
nCounter = 1; % count number of pairs entered (n)

%*****INPUT*****
% ask user to enter a velocity and distance until velocity = 0
xVel(nCounter) = input('Enter the velocity of a nebula(enter 0 to stop): ');
while xVel(nCounter) ~= 0
    yDist(nCounter) = input('Enter the distance of a nebula: ');
    nCounter = nCounter + 1;
    xVel(nCounter)  = input('Enter the velocity of a nebula(enter 0 to stop): ');
end % end while
nCounter = nCounter - 1;  % don't count xVel(nCounter) = 0
```

```
%*****COMPUTE*****
% accumulate into sums data
for n = 1 : nCounter
    sumX  = sumX  + xVel(n);            % sum of all x (velocity)
    sumY  = sumY  + yDist(n);           % sum of all y (distance)
    sumXX = sumXX + xVel(n) ^ 2;        % sum of x^2   (velocity)
    sumXY = sumXY + xVel(n) * yDist(n); % sum of x*y   (vel*dist)
end
% compute SLOPE (m)-----------------------------------------
num    = nCounter * sumXY - sumX * sumY;
den    = nCounter * sumXX - sumX ^ 2;
mSlope = num / den;

% compute INTERCEPT (b)-----------------------------------------
num        = sumY * sumXX - sumX * sumXY;
yIntercept = num / den;

%*****OUTPUT*****
% print velocity and distance data
fprintf('\n   NEBULA INPUT DATA\n');
fprintf(' VELOCITY  DISTANCE\n');
fprintf('  km/sec  106 parsecs\n');

for n = 1 : nCounter
    fprintf('   %4.0f      %5.3f\n', xVel(n), yDist(n));
end

% print LINEAR EQUATION -- y = mx + b
fprintf('\nLINEAR EQUATION: distance = %.4f * velocity + %.3f \n\n', mSlope, yIntercept);

%*****USE LINEAR EQUATION*****
% ask user for velocity
xVelocity = input('Enter a velocity of a nebula from above: ');
% compute distance using linear equation
yDistance = mSlope * xVelocity + yIntercept;
% print velocity and distance
fprintf('For velocity = %.0f, distance = %.3f\n', xVelocity, yDistance);
```

```
---'Save As' this file using the name in the assignment instructions.
---Type you information.
---Submit the completed development plan via Blackboard with you other files.

NAME:        J Hundley
ASSIGNMENT:  Lab06
DATE:        October 9, 2011

PROBLEM SOLVING IN ENGINEERING AND SCIENCE
Always use a systematic problem-solving strategy.

1. STATE THE PROBLEM:
---Describe the problem to be solved for the assignment.
Given pairs of velocity and distance from the Hubble data.
Use least-squares to compute the slope and y-intercept of a
linear equation to estimate the distance for a velocity.

2. DESCRIBE THE INPUT AND OUTPUT REQUIREMENTS:
---List and describe the following as needed to solve the problem, as needed.
---Include units where needed.

   CONSTANTS (known values that don't change):
N/A

   INPUT (values needed to find the output):
pairs of
x - velocity  km/sec
y - distance  106 parsecs

   OUTPUT (unknowns)
slope
y-intercept

   Relevant formulas:
      (for complicated equations, it may be helpful to divide it into parts)
mSlope    =(nCounter*sumXY-sumX*sumY)/(nCounter*sumXX-sumX^2)
yIntercept=(sumY*sumXX-sumX*sumXY)   /(nCounter*sumXX-sumX^2)
y = mx + b linear equation

3. WORK HAND EXAMPLES
---Solve the problem with a few hand examples.
---Record the input values used and the results
 VELOCITY  DISTANCE
  km/sec  106 parsecs
    170      0.032
    290      0.034
   -130      0.214
    -70      0.263
   -185      0.275

slope       = -0.0006
y-intercept = 0.172
```

4. DEVELOP AN ALGORITHM:
---Think about the steps used to solve the problem to solve the problem by
   hand and list them here to create an algorithm.
---The algorithm steps should be used as comments in your program as a guide.

```
*****INITIALIZE*****
  initialize counter and accumulators

*****INPUT*****
  while velocity not = 0
  ask user to enter a velocity and distance
  add values to vectors
  increment counter

*****COMPUTE*****
  accumulate into sums data
  compute SLOPE (m)
  compute INTERCEPT (b)

*****OUTPUT*****
  print velocity and distance data in table
  print LINEAR EQUATION -- y = mx + b

*****USE LINEAR EQUATION*****
  ask user for velocity
  compute distance using linear equation
  print velocity and distance
```

5. SOLVE THE PROBLEM:
---This step represents your writing a computer program to solve the problem.
---NOTE: Do not type your program here. Submit it as a computer program file.
---Use steps in your algorithm as comments in your program
   to guide the development of you program.

6. TEST THE SOLUTION:
---Run your program using the values from #3 to check for correctness.
---If there is an error, correct your program code and run again.

COMP1200-MatLab - Lab 06
Due midnight – Thursday – October 20
Submit `devPlan06.txt` and `Lab06.m` **via Blackboard**

Before you start writing your program:

Read all of these instructions carefully. The **`devPlan06.txt`** file at the assignment link is an incomplete development plan. You are to save the file and edited it by adding your name and the date and by completing: 1. STATE THE PROBLEM, 2. DESCRIBE THE INPUT AND OUTPUT REQUIREMENTS, and 3. WORK HAND EXAMPLES. Use the development plan as a guide when writing the m-script file solution for the following problem. **This file must be saved as a .txt file.**

For 3. WORK HAND EXAMPLES, find the slope and y-intercept using at least 4 pairs of velocity(x) and distance(y). **The values in your hand example should NOT be the ones used in the sample input and output below.**

NOTE: You will see later that the spelling and casing of file names is very important in MATLAB. Your submitted file(s) MUST be spelled and cased as instructed.
[-5 points per file for not doing so.]

***Problem*:**

***Program*: Lab06.m**

Edwin Hubble used the Mount Wilson Observatory telescopes to measure features of nebulae outside the Milky Way. He found that there is a relationship between a nebula's distance from earth and the velocity with which it was traveling from the earth. Hubble's initial data on 24 nebula is presented in Table 1 in the problem scenario.

The relationship between distance and velocity led scientists to propose that the universe came into being with a Big Bang, a long time ago. If material scattered from the point of the Big Bang traveling at a constant velocity, the distance traveled can be determined.

***Problem Constants*:**
See instructions.

***Problem Inputs*:**
See instructions.

***Problem Outputs*:**
See instructions.

***Other variables*:**
See instructions.

***Relevant formulas*:**
See instructions.

Regression Definition:
A regression is a statistical analysis assessing the association between two variables. It is used to find the relationship between two variables.

Regression Formula:
Regression Equation y = mx + b

$$\text{Slope } (m) = \frac{n\,(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2}$$

$$\text{Intercept } (b) = \frac{(\sum y)(\sum x^2) - (\sum x)(\sum xy)}{n(\sum x^2) - (\sum x)^2}$$

n is the number of x,y pairs

Instructions:

- ☐ Insert comments at the top and throughout each file
  - Include the follow comments at the beginning of this (and ALL) files.
    - % your name
    - % assignment number
    - % date you completed the assignment
    - % a short narrative about what the file does
  - Use your development plan as a guide for comments throughout each file

-5 points per file for absence of any of these required comments at the top

- ☐ Use `clc` and `clear all` at the beginning of your program.
- ☐ Use descriptive variable names.
- ☐ Use Sample Input/Output as a guide.
- ☐ No extra output, i.e., use semicolons!

- ☐ Initialize the counters and accumulator.

- ☐ Loops:
  - Think carefully about what needs to be done before the loop, in the loop, and after the loop
  - Use a **sentinel loop** for entering velocities and distances.
  - Use **counting loops** when summing data and printing table.
- ☐ Printing:
  - Use **fprintf** for all output.
  - Decimal places:
    - velocity 0
    - distance 3
    - slope 4
    - y-intercept 3
  - Column numbers **right-justified**, i.e., right-aligned
  - No extra blank spaces in the other output.

New commands
`fprintf`
`input`

Revisit
using index with a vector, Ch.4
initialize counter and accumulators
sentinel loop
counting loop

***Other information*:**

- Ask the user to enter a velocity and distance pair **until zero(0) is entered for velocity**.
- Build vectors to store the velocities and distances using an **index** to assign the values in the elements.
- Count the number of pairs (n) of velocity and distance and use to control for loops later.
- Compute the sums needed to computer the slope and y-intercept.

- Print the contents of the velocity and distance vectors in a two column table with a title and column headings.
- Print slope and y-intercept in the form of a linear equation. Use the answers in your hand example to check slope and y-intercept.
- Ask the user to enter one of the velocities entered earlier and compute the distance using the linear equation that you create. Note: The distance may not be the exact value because of the limited amount of input.

Sample Input/Output:

```
Enter the velocity of a nebula(enter 0 to stop): 170
Enter the distance of a nebula: .032
Enter the velocity of a nebula(enter 0 to stop): 290
Enter the distance of a nebula: .034
Enter the velocity of a nebula(enter 0 to stop): -130
Enter the distance of a nebula: .214
Enter the velocity of a nebula(enter 0 to stop): -70
Enter the distance of a nebula: .263
Enter the velocity of a nebula(enter 0 to stop): -185
Enter the distance of a nebula: .275
Enter the velocity of a nebula(enter 0 to stop): 0

  NEBULA INPUT DATA
 VELOCITY  DISTANCE
  km/sec  106 parsecs
    170      0.032
    290      0.034
   -130      0.214
    -70      0.263
   -185      0.275

LINEAR EQUATION: distance = -0.0006 * velocity + 0.172

Enter a velocity of a nebula from above: -70
For velocity = -70, distance = 0.211
```

The values in your hand example should NOT be the ones used in the sample.

Submit via Blackboard:

`devPlan06.txt`	Software development method
`Lab06.m`	MATLAB script file