HW07: Least Squares Regression

Date Due: 2015-03-25

1. **(10Points) Do Problem 12.4 in Siauw and Bayen.**

2. **(15 points)** Sometimes it is necessary to estimate values that you can't measure easily from values that you can. For example, bone traits can usually only be measured accurately using non-invasive techniques like x-ray, CT scan, or MRI. This is expensive and time consuming, but if you can measure a large sample of people using these techniques, you can regress the data to develop an equation which relates (for example) a person's height to their tibia length.

   Two files have been uploaded: MaleTibiaData.txt and FemaleTibiaData.txt.
   They contain actual subject data from men and women. The first column is each person's height, h (in *cm*), the second column is their tibia length (in *mm*) measured from a CT scan.

   Write a function with header **`[tibL] = myTibiaLength(height, sex),`** where **`height`** is a user's height in cm and **`sex`** is either "m" or "f". The output argument will be a 2x1 vector where

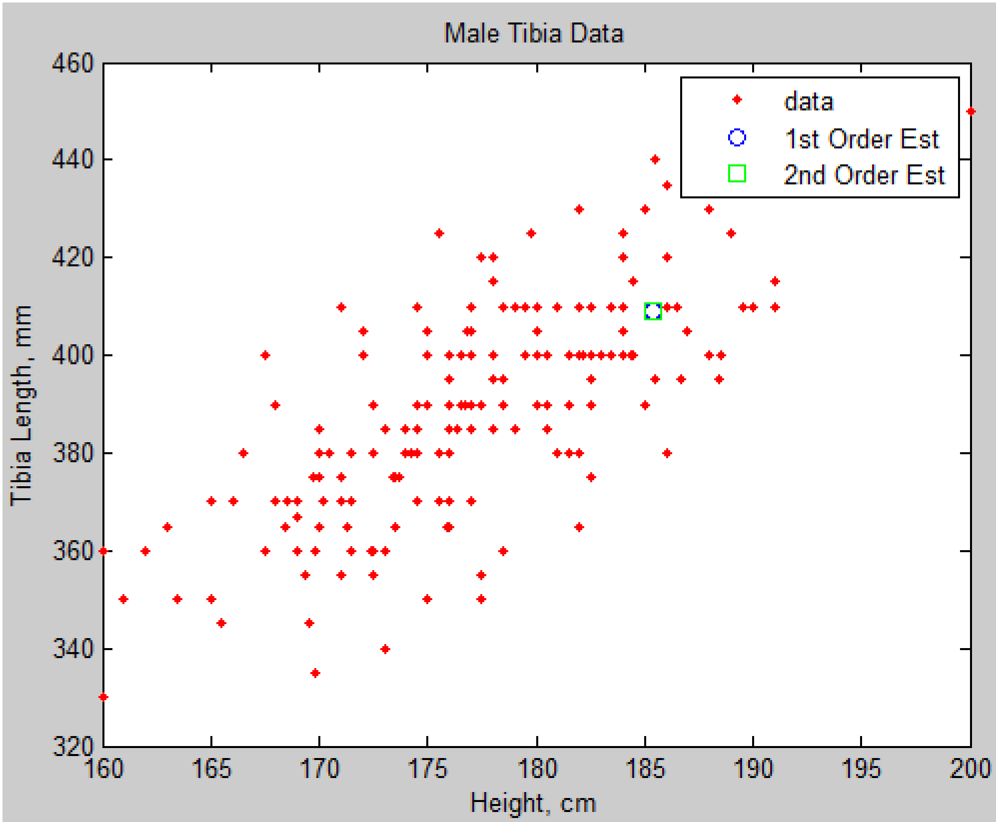
$$tibL(1) = a_1 + a_2 h$$
$$tibL(2) = a_1 + a_2 h + a_3 h^2$$

Notes:

a) Estimates for a male should be made from regressions of male data, and female estimates should be made from regressions of female data.
b) To load the data, use this syntax like this:
   `data = load('FileYouWantToLoad.txt');`

c) Use **`polyfit`** and **`polyval`** in your function.

d) The function should also generate a plot (below) with the tibial estimates plotted among the appropriate dataset (male or female) and an appropriate title:



3. **(15 Points) Do Problem 13.5 (myLinRegression) in Siauw and Bayen**

Note that you need to define a matrix A which will have the same number of columns as you have basis functions, f . (E.g., 2 columns if `f = {@sin, @cos}`, 3 columns if `f ={@sin, @cos, @tan}`, etc., etc.,) and the same number of rows as there are data points (e.g., `length(x)`). The columns of A will represent the evaluation of x by each basis function. After you have filled A, Beta can be solved for using the equation in the middle of page 204 and as demonstrated in the "Try It!" on page 206.

You might also want to try the test case of

```
>> x = linspace(0,2*pi,1000)';
>> y = 3*sin(x) - 2*cos(x) + randn(size(x));
>> f1 = @(x) 1;
>> f2 = @(x) x;
>> beta = myLinRegression(f, x, y)
>> plot(x,y,'b.',x,beta(1)*f{1}(x) + beta(2)*f{2}(x),'r','LineWidth',3)
```

to see how linear regression fits the data

Deliverables: Upload to Blackboard the following m-files:
myMakeLinInd.m
myTibiaLength.m
myLinRegression.m