AINT351 MACHINE LEARNING 2016
P1.2: LINEAR REGRESSION

IMPORTANT NOTICE

The final lab journal should contain all your individual lab reports
Submit to DLE: 12th January 2017 (4pm)
You will receive feedback within 20 working days

Please only submit your overall report in PDF format. You **must** include your student number in a header on every page of the report and **also include** your student number **in the document filename**!

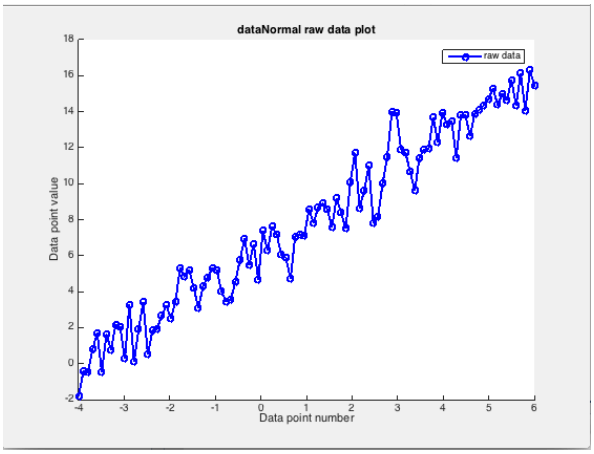
Each laboratory report should contain a few pages of explanation (although it can be more than this is necessary) and a set of images showing the plots requested by the individual practical exercises, as well as embedded Matlab code that you developed to implement your solutions.

The report must be a **stand-alone document**. Please embed images in-line in the report. In order to show video demonstrations if you have any, please use Internet links to videos that you have been uploaded to YouTube. Please do not submit more than the single PDF file.

AINT351 MACHINE LEARNING 2016
P1.2: LINEAR REGRESSION

1. Generate a noisy line

- Generate 100 sample points using the equation for a 1D line:
  y = mx + C
  Use the following parameters:
  x-axis starting point = -4;
  x-axis ending point = 6;
  m = 1.6;
  C = 6;
- HINT: you can use the Matlab function linspace to sample x
- Use the randn function to add Gaussian noise to the data values with a mean of 0 and standard deviation of 1
- Plot the data values against sample number to generate a plot like that shown below:
- Include your commented Matlab solution code embedded in the report document

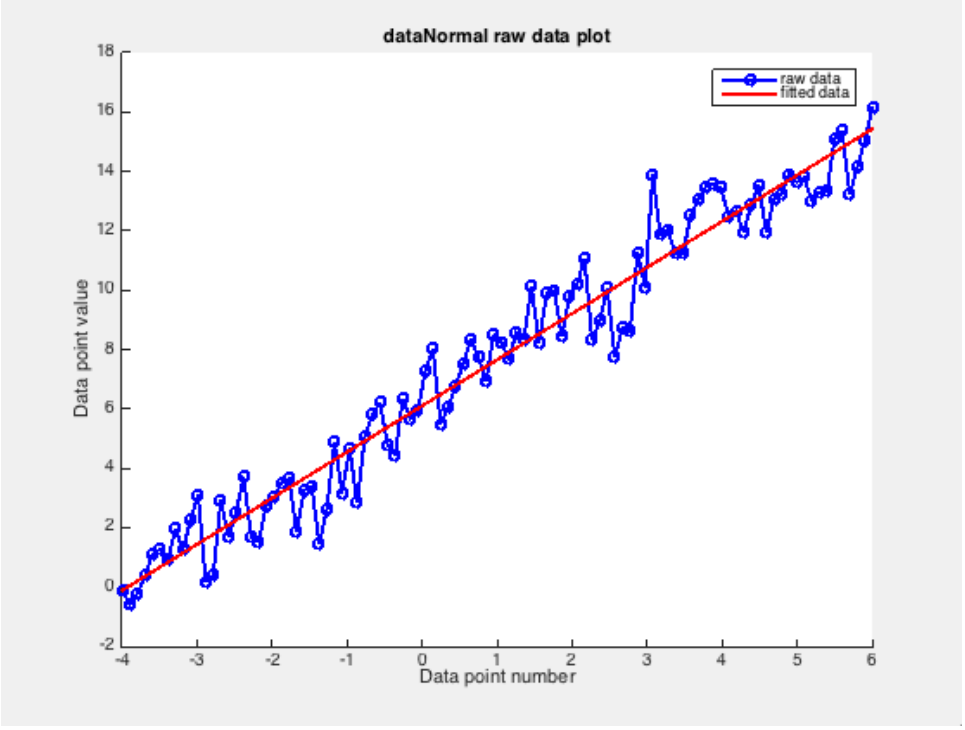


2. Implement linear regression from first principles

- Implement linear regression from first principles in Matlab
- DO NOT use a Matlab function to do this regression task!
- Include your commented Matlab solution code embedded in the report document

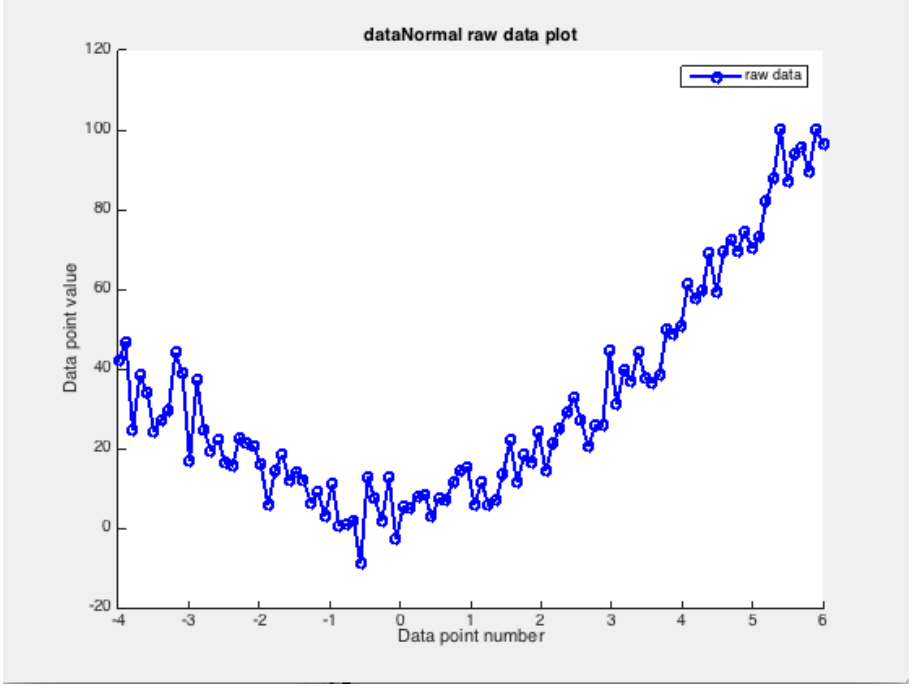
3. Fit the test line using your linear regression function

- Fit the test data
- Plot the raw data and fitted line. You should get a plot like that below:
- Include your commented Matlab solution code embedded in the report document



4. Generate a noisy quadratic curve

- Generate 100 sample points along the equation for a 1D quadratic line
  $y = Ax^2 + Box + C$
  Using the quadratic coefficients
  A = 1.6;
  B = 2.5;
  C = 6;
  X-axis starting point = -4;
  x-axis ending point = 6;
- HINT: you can use the Matlab function linspace to sample x
- Using the randn function, add Gaussian noise to the data values with a mean of 0 and standard deviation of 5
- Plot the data values against sample number to generate a plot like that shown below:
- Include your commented Matlab solution code embedded in the report document



5. Fit the quadratic curve using linear regression

- Fit the test data with a straight line using Matlab linear regression function regress
- HINT: generate a linear basis for the regress function
  XLin = [X; ones(1,samples);];

- Also fit the test data with a quadratic line using linear regression function regress
- HINT: generate a quadratic basis for the regress function
  Quad =[X .* X; X; ones(1,samples);];

- Plot the raw data and fitted lines
- Include your commented Matlab solution code embedded in the report document