Queen's UNIVERSITY | FACULTY OF ENGINEERING AND APPLIED SCIENCE

Lab 1: Linear Regression

The objective of this week's laboratory is to give you some experience writing MATLAB functions and using MATLAB to solve linear regression problems. It will also give you experience in testing the functions you write, as well as how to output figures and plots with MATLAB.

Before You Start

MATLAB can only use code that is saved in its 'Current Folder'. To set the Current Folder in MATLAB, press the 'Browse for Folder' button (). To keep your code organized, it is strongly suggested that you **create a new folder for each lab**. Create a folder for Laboratory 1, and select it as your Current Folder. Save all files to this folder for this lab, and make sure you upload the correct files to onQ after finishing the lab. Each lab should be saved in its own folder in future weeks to ensure the proper files are submitted.

Part A – Due at end of the lab period

The objective of Part A is to write code that will use a weighted linear regression to calculate the intercept (`b`) and slope (`m`) of the straight line y = mx + b, that best fits a set of x-y pairs taking into account the uncertainties in the y-values (`dy`). The code should also calculate the uncertainties in the intercept (`db`) and slope (`dm`). Expressions for the intercept, slope and uncertainties in the intercept and slope are given below:

$$b = [\sum(wx^2)\sum(wy) - \sum(wx)\sum(wxy)]/D$$

$$m = [\sum(w)\sum(wxy) - \sum(wx)\sum(wy)]/D$$

$$db = \sqrt{\sum(wx^2)/D}$$

$$dm = \sqrt{\sum w/D}$$

where $D = \sum(w)\sum(wx^2) - [\sum(wx)]^2$,

and $w = 1/dy^2$

Task A1

Write a function called `addUp` that will take a vector as input and return the sum of the elements. MATLAB has a built in function called `sum` that does this operation, but to give you some practice you should write your own.

For Part A, the basic file format and some code documentation has been done for you. Download the **addUp.m** file from onQ and save it in your Current Folder.

Replace the `% YOUR CODE GOES HERE` comment with your own code to produce the sum of the vector. Make sure your code is easily understandable to you **and others** by including documentation. This includes choosing good variable names and using comments to explain confusing sections of code, such as `for` loops.

Once your code is complete you need to test it. Code can be tested by hand, but this can take a long time, so we can write programs to test our code. For this lab, programs to test your different functions have been given. Download the **test_addUp.m** and **checkEqual.m** files from onQ and save them in your Current Folder. Then open the **test_addUp.m** and press the green "run" triangle to test whether your function works or not. If all three tests pass, you're done Task A1!

Task A2

Write a function called `wRegression` that takes vectors containing the x and y data points and the y uncertainties as input, and returns the intercept, slope, and uncertainties in the regression line.

Again, the basic file structure is given for you on onQ. Download the **wRegression.m** file and save it in your Current Folder. Use your `addUp` function from Task 1 to easily do the summations needed. Since your **addUp.m** file is the same folder as your **wRegression.m** file, the `addUp` function can be used in your `wRegression` function. If they're in different folders, your `addUp` function would be unavailable.

CAUTION: Use the " `.^` ", " `./` " and " `.*` " commands (powers, division, and multiplication) when you want to operate on individual elements of the vectors. If you don't include the " `.` " MATLAB will assume that you want to perform a matrix operation.

To test whether your code works, download the **test_wRegression.m** file from onQ and save it in your Current Folder. Open it up and press the green "run" arrow to see whether all the tests pass.

Task A3

Write a function called `uwRegression` that takes vectors containing `x` and `y` data points as input, and returns the intercept, slope, and uncertainties in the regression line. Your code should perform an **unweighted** regression to handle cases when the uncertainty in the `y` data does not vary significantly between points, or can't be measured accurately.

To perform an unweighted regression, you will need to modify the expressions for the intercept, slope and uncertainties in the intercept and slope given in Task A1. **Note:** you cannot simply remove all sums containing the weight vector `w`, there should be factors of `N`, the number of data points in the `x` and `y` vectors, in your final expressions. For the uncertainties in the intercept (`db`) and slope (`dm`), use the following expressions:

$db = dy * \sqrt{\sum(x^2)/D}$

$dm = dy * \sqrt{N/D}$

where $D = N\sum(x^2) - [\sum(x)]^2$,

$dy = \sqrt{\sum h^2/(N-2)}$,

and $h = y - (mx + b)$

Note: you should be able to explain where these expressions come from.

Use the same basic file structure as for `wRegression` by copying the file, changing the filename, and changing the function name in the file.

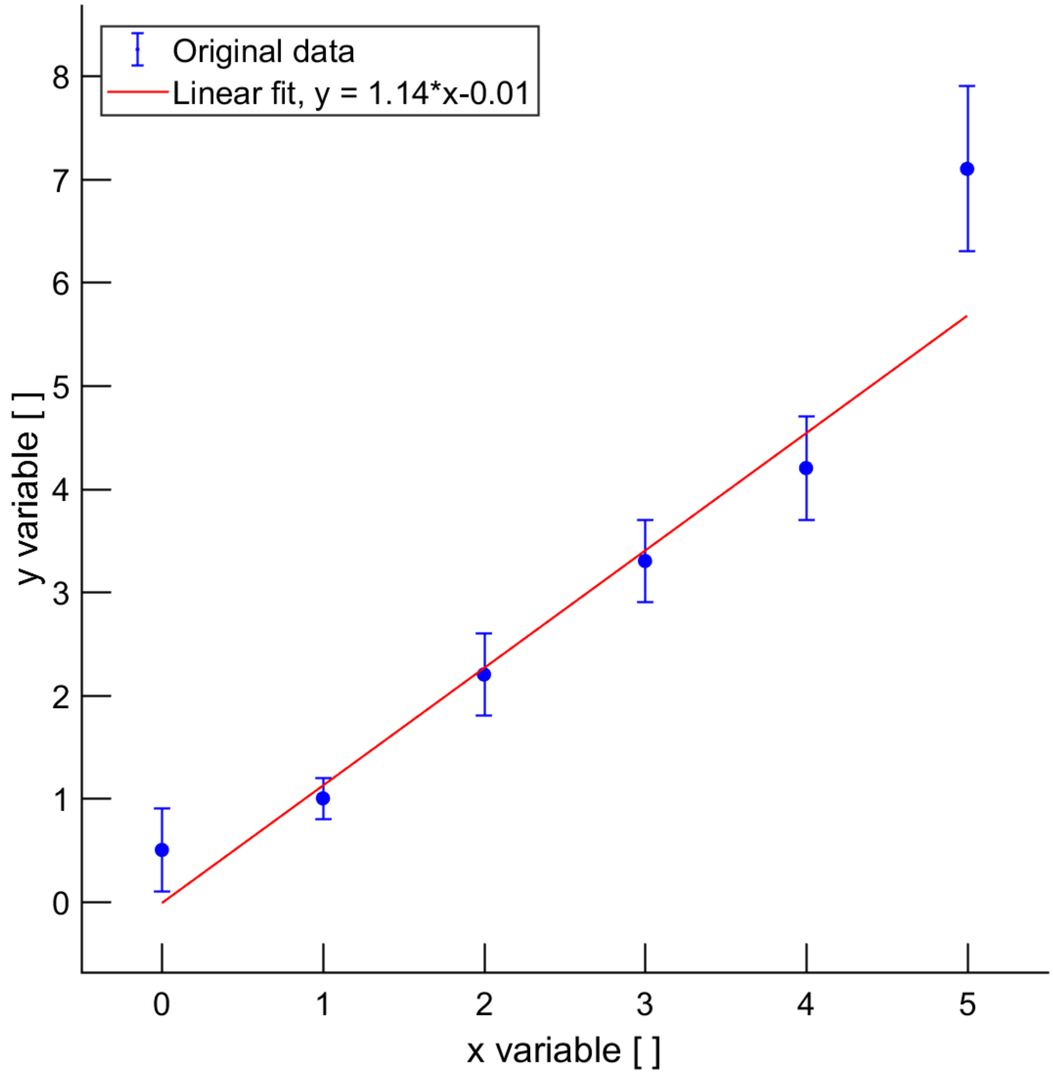
To test whether your code works, download the **test_uwRegression.m** file from onQ and save it in your Current Folder. Open it up and press the green "run" arrow to see whether all the tests pass.

Task A4

Write a function called `regressionPlot` that takes a set of `x`, `y` and `dy` values, and by using `wRegression`, plots the line of best fit for the data points given. The graph should also display the original data points with error bars and should include axis labels (make something up). To plot the original data as points, use the command `plot(x, y, '*')`. You should also figure out how to change the error bar (e.g. color, cap width, linewidth, etc.), data point (color, marker, size), axes (font, font size, tick direction, tick length, etc.) and plot (plot window size) properties. Download the **regressionPlot.m** file from onQ and save it in your Current Folder to give yourself a starting point.

You will have to figure out how to make MATLAB plot your data and regression line. MATLAB provides two useful commands for finding documentation on a function and any you've written: `help` and `lookfor`. If you know the name of the function, or can guess the name, then entering the command `help` followed by the name of the function will display the documentation on the function. For example, the command `help errorbar` will tell you how to generate a graph with error bars. Similarly, `help hold` will tell you how to plot more than one graph in the same window (so that you can plot the experimental data and the regression line on the same graph). If you are unsure of the name of the function, then entering the command `lookfor` followed by a key word will search and display documentation containing that word. For example, the command `lookfor label` would display a list of all the documentation that includes the word "label". All of the MATLAB documentation is also online, so searching for functions in Google can also be a great way to find useful tips and tricks.

Testing this function will be harder, as it outputs a figure instead of testable values. Provided is a function (download '**test_regressionPlot.m**' from onQ) that should generate a similar (but not identical) figure to the one here, which is for a slightly different dataset, if your `regressionPlot` function is working correctly.



Part A: Submission List

When complete, please submit the following files to the **Lab 1: Part A** Dropbox on onQ for marking:

- addUp.m
- wRegression.m
- uwRegression.m
- regressionPlot.m
- Any other functions you wrote that the above files use

Part B: Due Sunday @ 9PM

Task B1: Error Analysis for the ENPH 253 Impedance Experiment

In this experiment you will use weighted regression analysis to calculate the inductance and resistance of a circuit element. The details of the experiment can be found in your ENPH 253 laboratory manual. In brief, the magnitude of the experimental impedance of the circuit element is obtained by applying a sinusoidal voltage across the circuit element and recording the RMS Voltage (V) and Current (I). The magnitude of the impedance is given by the expression: $|Z|=V/I$. For an inductor it can be shown that $|Z|^2=R^2 + \omega^2L^2$, where R is series resistance of the inductor, L is the inductance, and ω is the **radian** frequency. Thus, by plotting $|Z|^2$ as a function of ω^2, and drawing the regression line, the value of L^2 can be determined from the slope and the value of R^2 from the intercept.

Write a function `findInductanceAndResistance` to determine the value of L (in H), R (in Ω) and the uncertainty in L and R for an inductor. Write another function `plotInductanceAndResistanceData` to plot the data points with error bars and the show the regression line. The starting files and testing files are available for download on onQ.

Note: To do this task you will have to use your `wRegression` function from Part A. To use it you will need to determine the uncertainty in $|Z|^2$ based on the uncertainties in V and I. An easy way to estimate the uncertainty in a function of several variables is to add the errors introduced by each variable in quadrature (see the ENPH 253 lab manual). For the example, if you defined a helper function under your `findInductanceAndResistance` called `Z2` (see the given **findInductanceAndResistance.m** file) that calculated $|Z|^2=(V/I)^2$, then the following command would generate a vector containing the uncertainty in $|Z|^2$ due to the uncertainty in the voltage (dV) and current (dI).

```
dZSq=((Z2(V+dV,I)-Z2(V,I)).^2+(Z2(V,I+dI)-Z2(V,I)).^2).^0.5;
```

Remember, you will also need to propagate the uncertainties when taking the square-root of the slope and intercept to find the inductance and resistance. For example, if A is the intercept and dA is the uncertainty in the intercept, then $R=A^{1/2}$, and using the approximate method described above $dR=|(A+dA)^{1/2}-(A)^{1/2}|$.

Task B2

Repeat Task B1 for a capacitor using $|Z|^2=R^2 + 1/(\omega^2C^2)$, writing the function `findCapacitanceAndResistance`. Again starting code and testing code is available for download from onQ.

Caution: This time you will be plotting $|Z|^2$ as a function of $1/(\omega^2)$.

Part B: Submission List

When complete, please submit the following files to onQ for marking:

- findInductanceAndResistance.m
- plotInductanceAndResistanceData.m
- findCapacitanceAndResistance.m

Any other functions you wrote that the above files use