Week 10: Vectorization and Intro to Linear Regression

In this week, we begin studying MATLAB. The goal of this week and the next three meetings will be to implement linear regression to approximate data and make predictions.

Vectorization

Vectorization can make tedious calculations much more concise and easier to understand. Try to vectorize the following procedures without using any if-statements or for loops.

Problem 1: Mathematical Function

Let f(x) = $x^3 + 3x^2 - 7x + 2$. Given an array of x-values, x, apply the function f to all of these points.

Sample:
f([-2 , -1 , 0 , 1 , 2 , 3.5]) = [20.0000 , 11.0000 , 2.0000 , -1.0000 , 8.0000 , 57.1250]

Problem 2: Encryption

Here is a potential cipher you might use: given some lowercase plaintext that has been wrapped into a matrix of dimension n-by-m, we change each letter in the first column to the letter 1 before that, each letter in the second column to the letter 2 before that, …, each letter in the m-th column to the letter m before that, wrapping back around to Z if necessary. For example,

```
        a c                              z a
                       ->
        c d                              b b
```

Please implement this encryption function.

Sample:
encrypt(['matlab' ; 'isreal' ; 'lyfun!']) = [‘lyqhvv’ ; ‘hqoavf’ ; ‘kwcqii’]

Linear Regression Cost Function

Given some 2D data points, calculate the cost function for a potential line of best fit. Remember that the cost function is defined as

$$\frac{1}{2m}\sum_{i=1}^{m}\left(h\left(x^i\right)-y^i\right)^2$$

The data points will be given as an m-by-2 matrix where each row is a data point. You will also be given two parameters Θ_0 and Θ_1 which represent the equation

$$y = \Theta_0 + \Theta_1 x$$

and you will have to calculate the y values that the hypothesis function predicts and use the difference between the prediction and actual value to calculate the total cost.

Sample: cost_function([1 , 2 ; 3 , 4 ; 2 , 3 ; 1.3 , 2.4; 2.4   3.2 ; 3.3 , 3.7] , 0 , 1) = 0.4175