# BasicProblems

May 25, 2017

**Strang Matrix Problem** Use Julia's array and control flow syntax in order to define the NxN Strang matrix:

$$\begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & -2 \end{bmatrix}$$

i.e. a matrix with `-2` on the diagonal, 1 on the off-diagonals, and 0 elsewere.

```
In [1]: #### Prepare Data For Regression Problem

        X = rand(1000, 3)                # feature matrix
        a0 = rand(3)                     # ground truths
        y = X * a0 + 0.1 * randn(1000);  # generate response

        # Data For Regression Problem Part 2
        X = rand(100);
        y = 2X  + 0.1 * randn(100);
```

**Regression Problem** Given an Nx3 array of data (`randn(N,3)`) and a Nx1 array of outcomes, produce the data matrix `X` which appends a column of 1's to the front of the data matrix, and solve for the 4x1 array $\beta$ via $\beta$`X = b` using `qrfact`, or `\`, or the definition of the OLS estimator. (Note: This is linear regression).

Compare your results to that of using `llsq` from `MultivariateStats.jl` (note: you need to go find the documentation to find out how to use this!). Compare your results to that of using ordinary least squares regression from `GLM.jl`.

**Regression Problem Part 2** Using your OLS estimator or one of the aforementioned packages, solve for the regression line using the (X,y) data above. Plot the (X,y) scatter plot using `scatter!` from Plots.jl. Add the regression line using `abline!`. Add a title saying "Regression Plot on Fake Data", and label the x and y axis.

**Logistic Map Problem** The logistic difference equation is defined by the recursion

$$b_{n+1} = r * b_n(1 - b_n)$$

where $b_n$ is the number of bunnies at time $n$. Starting with $b_0 = .25$, by around $400$ iterations this will reach a steady state. This steady state (or steady periodic state) is dependent on $r$. Write a function which plots the steady state attractor. This is done as follows:

1) Solve for the steady state(s) for each given $r$ (i.e. iterate the relation 400 times).

2) Calculate "every state" in the steady state attractor. This means, at steady state (after the first 400 iterations), save the next 150 values. Call this set of values $y_s(r)$.

3) Do steps (1) and (2) with $r \in (2.9, 4)$, `dr=.00005`. Plot $r$ x-axis vs $y_s(r)$=value seen in the attractor) using Plots.jl. Your result should be the Logistic equation bifurcation diagram.