Homework 3

CSC-432

Due: 2/11/13

Question 1 (3 points)

Using the `random_walk_simulations` notebook that we covered in class, revise the code so that the "entity" walks North, South, East, or West with a probability of 20%, 30%, 45%, or 5%, respectively. Plot the path the entity takes, as we did in class. You do NOT have to do the animation part, just a static plot of the final path is fine.

Hint(s): You could still use `numpy.random.choice` for this, though you may want to combine this with the dictionary approach mentioned in the notebooks. You can do something like this

```
np.random.choice(["North", "South", "East", "West"])
```

You will want to investiage the `p` argument to choice for assigning the non-uniform probabilities.

In your dictionary, what would the value for the `key` "North" be? (0, 1) right?

Question 2 (2 points)

If A is a 80 x 5 matrix and B is a 5 x 1 matrix, what is the shape of the product AB? What is the shape of $B'A'$?

Question 3 (2 points)

Let

```
A = [[1, 3, 5],
     [2, 1, 6],
     [3, 0, 2],
     [4, 1, 2]]

B = [[1, 3, 5]]
```

What is the solution to AB'? You can do this with numpy using the `np.dot` function. However, I want to see you type out the summations in comments as well, so I know you understand what is going on under the hood. Python code should help you arrive at your intuitions not replace them.

Question 4 (2 points)

Suppose you have a system of equations

$$\begin{aligned} x + y + z &= 6 \\ 2x - y + 3z &= 9 \\ -x + 2y + z &= 9 \end{aligned}$$

Put this into two numpy arrays X and Y so that it is in the form XB = Y where B is the matrix of unknowns. Use `numpy.linalg.solve` to solve for B. Use `numpy.testing.assert_almost_equal` to ensure that $XB' == Y$

Question 5 (2 points)

Imagine that you have 4 observations of traffic congestion for the DC Metro area.

x = [9.5, 7, 7.8, 4]

You also have a measure of air quality for these same days.

y = [52.5, 48.1, 47.1, 41.9]

Assuming a linear relationship of the form

$$y = \beta_0 + \beta_1 x + \epsilon$$

or in matrix notation

$$y = X\beta' + \epsilon$$

Use `np.linalg.lstsq` to find the solution for β that minimizes the sum of squared errors. Print the result, clearly indicating which is the constant term and which is the coefficient on x. **Hint** be sure to attach a column of ones to your X array.

Question 6 (2 points)

Using the result from the previous question, calculate $\hat{y}$ where $\hat{y} = X\beta'$. Assign it to a variable called `yhat`. Use `yhat` to calculate the errors ϵ from this regression. Assign it to a variable named `residuals`. Calculate the sum of squared residuals and print it out. **Hint:** if the original equation states that $y = X\beta' + \epsilon$ and $\hat{y} = X\beta'$ how would we calculate ϵ? Make sure you use numpy broadcasting in your solution!