NumPy

Re-submit Assignment

Due Sep 20 by 11:59pm **Points** 100 **Submitting** a file upload
File Types pdf and html

Create a Jupyter Notebook that solves the following problems, using NumPy.

1) (20 points) Assume you are creating a 3D-Breakout game where a ball must break through a 3-dimensional structure of blocks. Create a function that returns a random integer number of hits that each block must take in order to be destroyed. The first line of the function should be:

def blockmodel(x,y,z,mu,sigma):

where x, y, z is the size of the whole structure and mu, sigma are the mean and standard deviation of the Gaussian distribution that models the number of hits it takes to destroy a block. The output of the function should be a NumPy array of size x by y by z, where each element is the number of hits it takes to destroy each block, each a random draw from the Gaussian distribution.

Demonstrate your code by running the following:

blockmodel(2,3,4,5,0.5)

2) Download the **hurricanes data file**.

(5 points) Using matplotlib, create a plot of the number hurricanes versus time. Make this plot look nice.

(20 points) Create a function called running_mean which smooths the data by replacing each element in the data with the mean of the surrounding data

def running_mean(data, length)

where data is the input array and length is the number of elements to average around each datum. Hint: use numpy.cumsum. Plot the smoothed data on top of the original data to show smoothing with different length arguments.

(15 points) Make a histogram of the number of tropical storms that aren't hurricanes. Are the data normally distributed? Plot a Gaussian distribution on top of the histogram by computing the mean and standard deviation of the data and plotting the Guassian distribution with that mean and standard deviation. Answer the question: Are the data normally distributed?

3) Implement **Conway's Game of Life** (https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life) . Use the function templates in **this file** to get you started.

(30 points) Demonstrate your Life simulator with a 20x20 grid. Show an image of the beginning board and then the image of 20 generations. Make this look nice in your notebook. Use subplots to show multiple images per row, for example. 4 images per row, with 5 rows would be nice.

(10 points) Edit your code so that at each generation, not only is the grid shown, but also the number of neighbors for each grid square shown. Use different colors for different numbers of neighbors. Demonstrate your visualization with a 20x20 grid over 20 generations.

Submit your Jupyter notebook as a PDF or HTML file.