**CS 5163 HW2**

**Due: Sunday Sept 30, 11:59pm.**

**Submit your source code and writeups via blackboard.** Please include your solutions to all questions in **one single document**. Source code can be separated into multiple files if you want and source code for Q1 and Q2 are not required. Source code for Q3 and Q4 are required. Name/document your functions to the extent that it is obvious which question the function is solving.

1. **Numpy and vectorized computing (24 points)**

Let x be a numpy array with 4 rows and 4 columns:

x = numpy.array( [[ 1, 2, 3, 4],

[ 5, 6, 7, 8],

[ 9, 10, 11, 12],

[13, 14, 15, 16]])

What is the result of the following operations? (Please try to solve them without using a computer and then use python to validate your results.)

1. y = x[:, 2]; print (y)

[3, 7, 11, 15]

1. y = x[-1,:2]; print (y)

[13, 14]

1. y = x[:, [True, False, False, True]]; print(y)

[[1, 4],

[5, 8],

[9, 12],

[13, 16]]

1. y = x[0:2, 0:2]; print(y)

[[1, 2],

[5, 6]]

1. y = x[[0, 1, 2], [0, 1, 2]]; print(y)

[1, 6, 11]

1. y = x[0]\*\*2; print(y)

[1, 4, 9, 16]

1. y = x.max(axis=1); print(y)

[4, 8, 12, 16]

1. y = x[:2, :2]+x[:2,2:]; print(y)

[[4, 6],

[12, 14]]

1. y = x[:2, :3].T; print(y)

[[1, 5],

[2, 6],

[3, 7]]

1. y = x[:2, :3].reshape((3, 2)); print(y)

[[1, 2],

[3, 5],

[6, 7]]

1. y=x[:, :2].dot([1, 1]); print(y)

[3, 9, 11, 27]

1. y = x[:, :2].dot([[3, 0], [0, 2]]); print(y)

[[3, 4],

[15, 12],

[27, 20],

[39, 28]]

1. **Probability (16 points)**. Show the results and your calculation using python-style code (see 2a for an example).
2. Toss a fair coin 5 times, what is the probability of seeing 5 heads in a row?

P(HHHHH | fair coin) = (0.5)\*\*5 = 0.0313

1. Given a box that contains 90% fair coins and 10% loaded coins, (a loaded coin gives heads 90% of the time), what is the probability for a randomly drawn coin to give 5 heads in a row?

P(HHHHH | random coin from box of mixed coins) = ((0.5\*0.9)+(0.9\*0.1))\*\*5 = 0.0459

1. Given a coin randomly drawn from the box mentioned above in 2b, what is the probability to get exactly 9 heads in 10 tosses?

P(9 H and 1T | random coin from box of mixed coins) =

10\*((0.5\*0.9)+(0.9\*0.1))\*\*9((0.5\*0.9)+(0.1\*0.1))= 0.000036

1. If you randomly pick a coin from the box mentioned above in 2b, toss it 5 times and get all heads. What is the probability that this is a loaded coin?

P(loaded | HHHHH) = (0.9\*\*5\*0.1)/(0.9\*\*5\*0.1+(0.5)\*\*5\*0.9) = 0.6773

1. If you randomly pick a coin from the box mentioned above in 2b, toss it 10 times and get nine heads and one tail. What is the probability that this is a loaded coin?

P(loaded | 9H and 1T) =

(0.9\*\*9\*0.1)/(0.9\*\*9\*0.1+0.5\*\*9\*0.9) \* (0.1\*\*1\*0.9)/(0.1\*\*1\*0.9+0.5\*\*1\*0.1) = 0.61495

1. **Statistics (30 points)**
2. Write a program to simulate tossing a fair coin for 100 times and count the number of heads. Repeat this simulation 10\*\*5 times to obtain a distribution of the head count and plot the histogram as well as CDF. Label your plots clearly.
3. Use the binomial distribution CDF (use scipy.stats.binom.cdf) to estimate the probability of having NO MORE THAN k heads out of 100 tosses, where k = 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100. Do these probabilities agree with the numbers of head counts you obtained in 3a? (Plot the head counts you obtained from the simulation results in 3a against the probabilities from your theoretical calculation here. Plot in loglog scale is probably needed to visualize small probabilities.)
4. Make a normal probability plot (thinkstats ch 4.4) to show that this distribution is close to a normal distribution with mean 50 and standard deviation 5.
5. Use normal distribution approximation to calculate the cumulative probabilities that you were asked to calculate in 3b, and compare the two results using a loglog plot. (Hint: If head count follows a normal distribution with mean = 50 and std = 5, a head count of 40 is equivalent to z-score = -2, and the corresponding CDF can be calculated using scipy.stats.norm.cdf.)
6. **Data analysis (30 points + 10 extra points)**

In this exercise we will be analyzing the BRFSS weight vs height data. Download data and code skeleton from course website. The code skeleton contains code to load the data and generate a numpy array object. The five columns in the numpy array represent: age, current\_weight (kg), weight\_a\_year\_ago (kg), height (cm), and gender, where gender == 1 represents male and 2 represents female.

* 1. Use your code from HW1 to produce a summary statistics graph on current\_weight, weight\_a\_year\_ago, and height.
  2. Define weight\_change = (current\_weight – weight\_a\_year\_ago). Calculate correlation between weight\_change and the following variables, and determine which one is most correlated (regardless of sign of correlation) with weight\_change. Use scatter plot to support your conclusion.
     1. current\_weight
     2. weight\_a\_year\_ago
     3. age
  3. Calculate and compare the mean and SEM (standard error of the mean) for the weight\_change of male and female. Use t-test to test whether there is a significant difference between the weight\_change of male and female.
  4. Randomly split the subjects into two groups of roughly equal sizes, and use t-test to test whether there is a significant difference between the weight\_change of the two groups. Repeat the process 1000 times and plot the distribution of the -log10(p-value) of the t-test results. What can you say about the difference between male and female in terms of their weight\_change? (Consider both the p-value and the absolute differences between the two means.)
  5. Define weight\_height \_ratio as current\_weight/height. Use t-test to test whether there is a significant difference between the weight\_height\_ratio of male and female. Also, repeat the analysis you did in 4d, but replace weight\_change with weight\_height\_ratio in your analysis.
  6. (**Extra 10 points**) Propose and perform your own analysis that utilizes different skills taught in class (or reveal additional interesting insight from this data set).