CS 3753 & 5163 Data Science Fall 2020

Homework 4 (100 + 10 extra)

Submission:

1. Submit a single python script (abc123_hw#.ipynb or abc123_hw#.py) through Blackboard learn. All the results are outputted from your Python code.
2. You should have the instruction of running your code at the beginning of your code. It should run successfully either in the basic Python3 environment or in Jupyter Notebook.
3. If your code cannot run, we assume your code can run, then we will check whether your code is correct logically. If so, half points will be deducted. Otherwise, more points will be deducted if your code is wrong or there is no code.
4. Do not compress your source code and data files. Make sure all your files are in the same folder when you run the code. So, after the graders download your homework, they do not need to set the path for the data file. They can run your code successfully.
5. If there is any plagiarism, you will lose all points on the questions at first time. In next, you will lose all points in the whole homework.
6. You can submit your homework 3 times before the deadline. The late submission will lose 15% of the total points in the assignment. The submission is unacceptable if it is more than 24 hours late. The compressed files will receive a warning at the first time and will lose 10% points later.

Questions

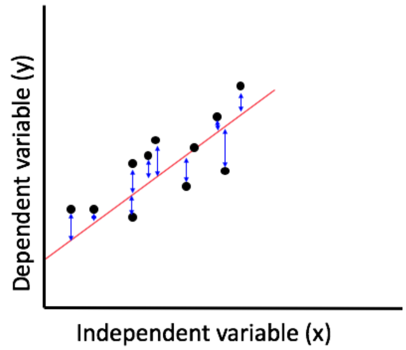
1. **Linear Regression (20 points)**

If the plot of n pairs of data (x , y) for an experiment appear to indicate a "linear relationship" between y and x, then the method of least squares may be used to write a linear relationship between x and y.

The least squares regression line is the line that minimizes the sum of the squares of the errors (SSE) from each data point to the line (see figure below). The least square regression line for the set of n data points is given by $y = \beta x + \alpha$, where α and β are given by

$$\beta = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\alpha = \bar{y} - \beta * \bar{x}$$



The sales of a company (in million dollars) for each year are shown in the table. Output your results by the print() function in Python.

x(year)	2005	2006	2007	2008	2009
y(sales)	12	19	29	37	45

a. Write a Python code to find the least square regression line $y = \beta x + \alpha$. You either can write a Python code to calculate α and β based on the equations above or use the linear regression function in the model sklearn (described in the lecture) to get the results directly. (15 pts)
b. Use the regression line as a model to estimate the sales of the company in 2012 (in million dollars). (5 pts)

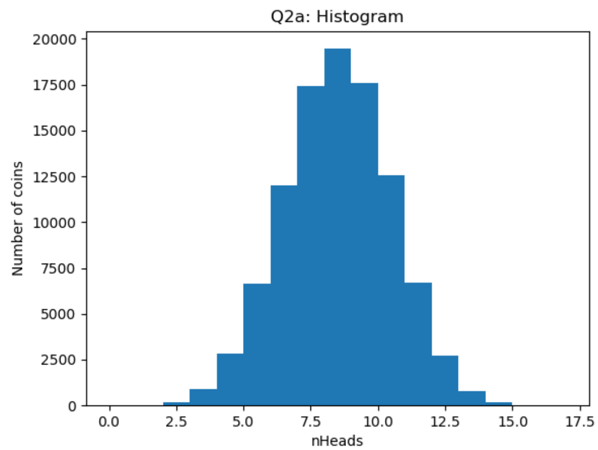
2. **Hypothesis testing (20 points).** If you flip the coin 15 times, and observed 12 heads, how likely this is a fair coin? A loaded coin tends to give head or tail more frequently. We set the p-value to be 0.05. We have the following statistical hypothesis testing
   H0 (null hypothesis): the coin is fair
   H1 (alternative hypothesis): the coin is loaded (i.e., biased towards head or tail)
   a. (15 pts) We assume H0 is correct, then we can calculate the probability P(at least 12 heads or 12 tails | coin is fair) =
   ____
   ____
   (hint: You do not need to calculate the final value. Just write the formula using the binomial PMF function *binom.pmf(k, n, p)* with actual values).

   b. (5 pts) If the value is smaller than the p-value 0.05, do you accept or reject H0?

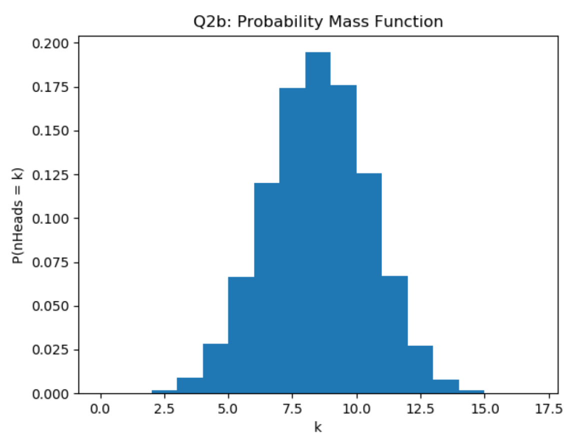
3. **Statistics (70 points)**
   Write a program to simulate an experiment of tossing a fair coin 16 times and counting the number of heads. Repeat this experiment 10^5 times to obtain the number of heads for every 16 tosses; save the number of heads in a vector of size 10^5 (nHeads). You should be able to do this in just a few lines. (Use np.random.uniform to generate a 2d array of 10^5 * 16 random numbers between 0

and 1; a value that is greater than 0.5 is considered a “head”.) Complete the following questions.

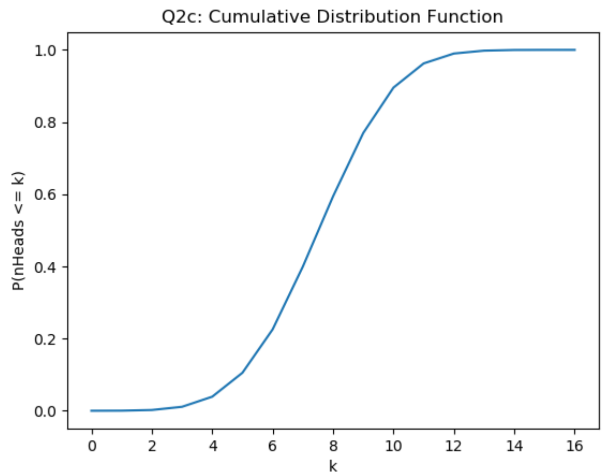
a. Plot the histogram of nHeads using plt.hist, with parameter bins = range(18). Label your plots clearly **(15pts)**.



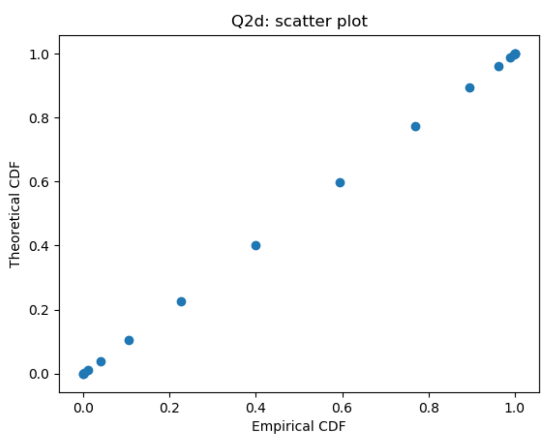
b. Plot the PMF using plt.hist with parameter bins and density. Label your plots clearly **(15pts).**

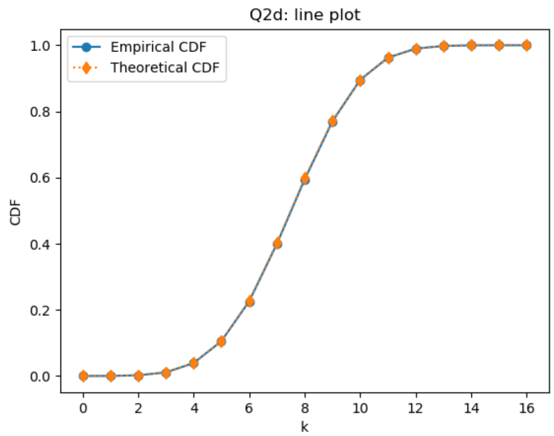


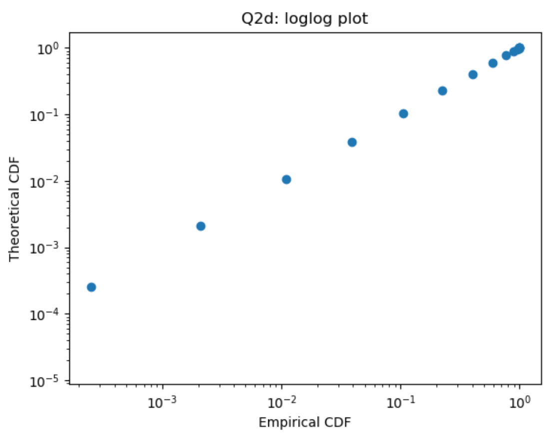
c. Calculate the probability of having NO MORE THAN k heads out of 16 tosses, where k = 0, 1, 2, … 15, 16. Plot this as a CDF (it accumulates the probabilities <= k). You can calculate the probabilities again or using values returned from a or b.) Label your plots clearly **(15pts).**



d. Use the binomial distribution CDF (use scipy.stats.binom.cdf) to compute the probability of having NO MORE THAN k heads out of 16 tosses, where k = 0, 1, 2, …, 15, 16 and compare these probabilities with the probabilities you obtained in 2c. (Plot the probabilities you obtained from the simulation results in 2c (empirical calculation) against the probabilities from your calculation here (theoretical calculation), as **a scatter plot and line graph**. Plot in **loglog scale** to visualize small probabilities.) **(25pts)**




Q2d: line plot
Empirical CDF
Theoretical CDF
1.0
0.8
0.6
0.4
0.2
0.0
CDF
0
2
4
6
8
10
12
14
16
k


Q2d: loglog plot
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
Theoretical CDF
10^{-3}
10^{-2}
10^{-1}
10^0
Empirical CDF