CS 3753 & 5163 Data Science

Homework 4 (100 points)

Submission: It is the same as previous homework.

Questions

1. **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) =

   _____

   _____

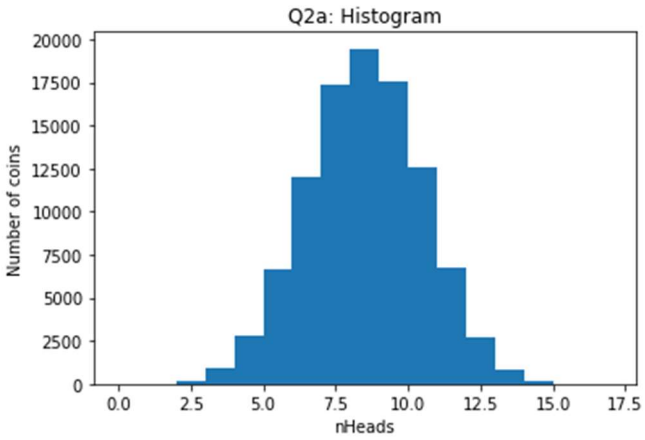
   (hint: You do not need to calculate the final value. Just write the formula using the binomial PMF function *binom.pmf(k, n, p)* with actual values).

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

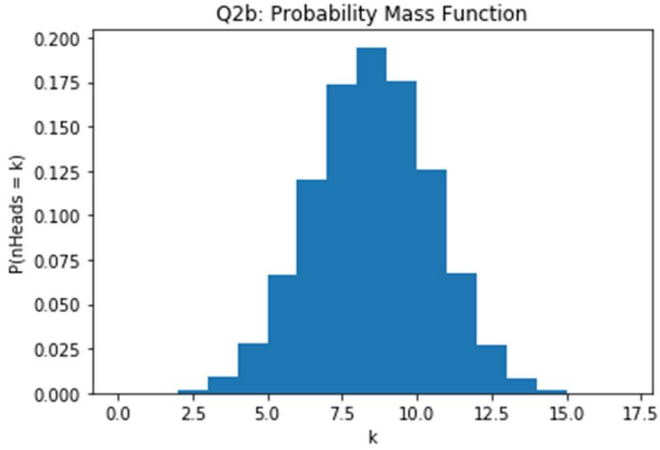
2. **Statistics (60 points)**

   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.

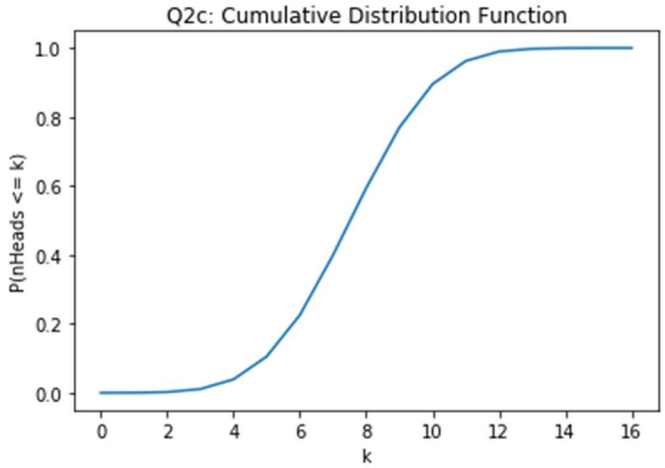
   a. Plot the histogram of nHeads using plt.hist, with parameter bins = range(18). Label your plots clearly **(15pts)**.



b. Plot the PMF using plt.hist with parameter bins and density. Label your plots clearly **(10pts).**



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).**

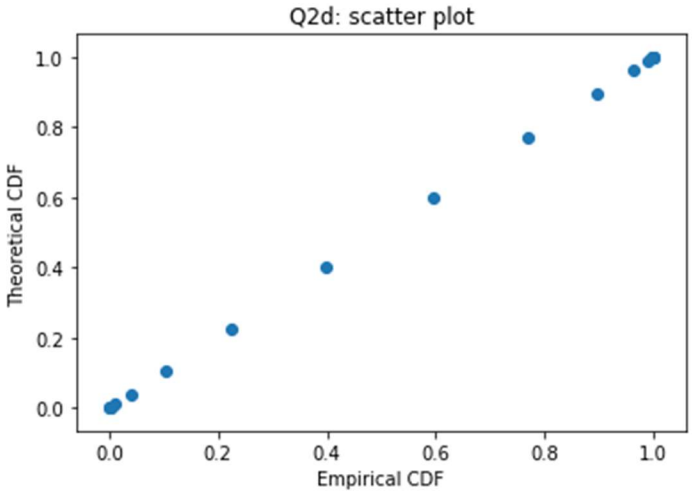


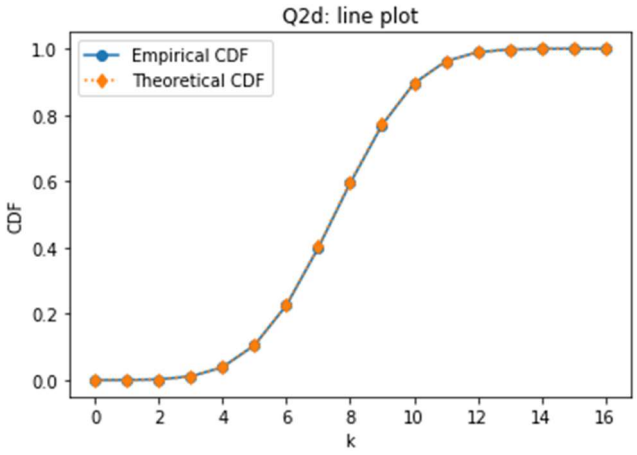
d. Use the binomial distribution CDF (use scipy.stats.binom.cdf) to compute the probability of having NO MORE THAN k heads out of 16 tosses, where k = 0, 1, 2, …, 15, 16 and compare these probabilities with the probabilities you obtained in subquestion c. (Plot the probabilities you obtained from the simulation results in subquestion c (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.) **(20pts)**

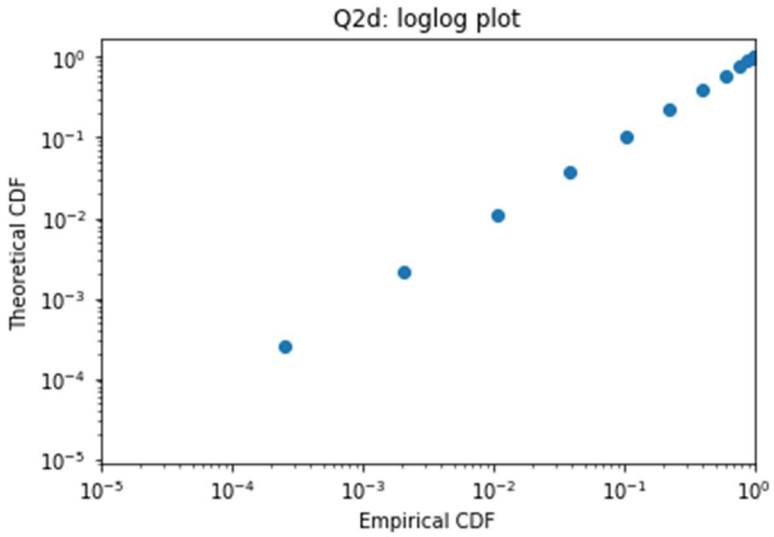
Remark: Please make sure the scales of x and y axis are correct in the loglog plot. Your plot should be exactly the same as the one in the document. Make sure the loglog plot have the values of x and y as

x: [0.0000e+00 2.5000e-04 2.1000e-03 1.0910e-02 3.8970e-02 1.0523e-01 2.2548e-01 3.9946e-01 5.9403e-01 7.6973e-01 8.9545e-01 9.6247e-01 9.8986e-01 9.9784e-01 9.9972e-01 9.9996e-01 1.0000e+00]

y: [1.52587891e-05 2.59399414e-04 2.09045410e-03 1.06353760e-02 3.84063721e-02 1.05056763e-01 2.27249146e-01 4.01809692e-01 5.98190308e-01 7.72750854e-01 8.94943237e-01 9.61593628e-01 9.89364624e-01 9.97909546e-01 9.99740601e-01 9.99984741e-01 1.00000000e+00]







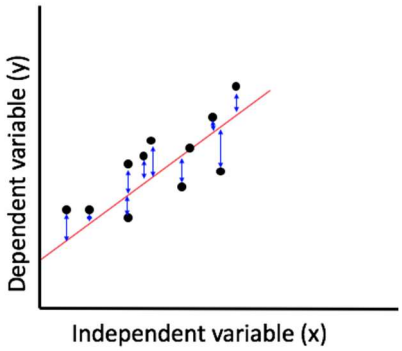
3. Linear Regression (20 points)

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

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

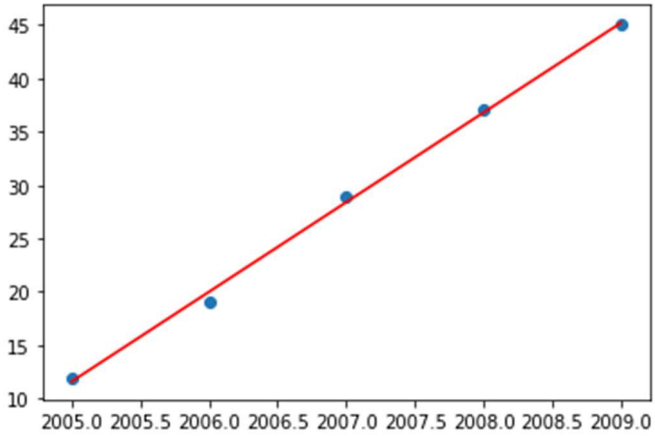
$$\alpha = \bar{y} - \beta * \bar{x}$$



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. (The line plot is optional plot) (15 pts)



b. Use the regression line as a model to estimate the sales of the company in 2012 (in million dollars). (5 pts)