Probability and Statistics Project 1

Distributions

January 12, 2022

[15 Marks] Task 1: Random Variables and Histograms

A **random variable** *X is a function $X : \Omega \to R$ that maps an outcome $\xi \in \Omega$ to a number $X(\xi)$ on the real line.*

There are 26 English letters. We can define random variable X as a letter we randomly draw from an English text. Consequently, we can think of X as an object with 26 different states. The mapping associated with the random variable is simple, $X(\text{“}A\text{”}) = 1$, $X(\text{“}B\text{”}) = 2$, etc. The probability of landing on a particular state approximately follows a histogram as shown below.

Task: Plot the frequency histrogram of the 26 English Alphabets from a given text. For this purpose, you will be implementing the function `plot_frequency_histogram` which takes as parameter the filename.

[15 Marks] Task 2: Probability Mass Functions

The probability mass function (PMF) of a random variable X is a function which specifies the probability of obtaining a number $X(\xi) = x$. We denote PMF as

$$p_X(x) = P[X = x]$$

The set of possible states of X is denoted as $X(\Omega)$

Roll a fair dice once. The sample space is $\Omega = \{1, 2, 3, 4, 5, 6\}$. We can assign a random variable X = number rolled. The random variable X takes six states : 1, 2, 3, 4, 5, 6. The PMF is therefore

$$p_X(1) = P[X = 1] = \frac{1}{6}$$

$$p_X(2) = P[X = 2] = \frac{1}{6}$$

$$p_X(3) = P[X = 3] = \frac{1}{6}$$

$$p_X(4) = P[X = 4] = \frac{1}{6}$$

$$p_X(5) = P[X = 5] = \frac{1}{6}$$

$$p_X(6) = P[X = 6] = \frac{1}{6}$$

Task: In this task, you will simulate the throwing of a fair die n times. You are required to experiment with reasonable values of n and for each value you are required to comment how the shape of the each obtained histogram compares with the ideal PMF.

[20 Marks] Task 3: Expectation, Variance & Moments

The expectation of a random variable X is

$$E[X] = \sum_{x \in X(\Omega)} x p_X(x)$$

The kth moment of a random variable X is

$$E[X^k] = \sum_{x} x^k p_X(x)$$

The variance of a random variable X is

$$Var[X] = E[(X - \mu)^2]$$

where $\mu = E[X]$ where is the expectation of X

Let X be a random variable with PMF $p_X(0) = \frac{1}{4}, p_X(1) = \frac{1}{2}$ and $p_X(2) = \frac{1}{4}$

$$E[X] = 0 \cdot \frac{1}{4} + 1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} = 1$$

$$E[X^2] = 0^2 \cdot \frac{1}{4} + 1^2 \cdot \frac{1}{2} + 2^2 \cdot \frac{1}{4} = 1.5$$

$$Var[X] = \left((0-1)^2 \cdot \frac{1}{4}\right) + \left((1-1)^2 \cdot \frac{1}{2}\right) + \left((2-1)^2 \cdot \frac{1}{4}\right) = \frac{1}{2}$$

Task: In Task 2, you plotted simulated the throwing of a fair die n times. In Task 3, you are required to compute the expectation, variance and moment for each value of n and compare with the actual expectation.
For this purpose, you will be implementing the function `compute_expectation_variance_moment`

Hint : Use `np.unique` to get frequency counts in a `numpy` array

[30 Marks] Task 4 : Common Discrete Random Variables

Task: In Task 1, you were introducted to Random Variables. In Task 2, you were introduced to PMF's. In Task 3, you were introduced to computing Expectation, Variance and Moments. In Task 4, you will be introduced to Common DRVs that you have already discussed in class. The aim of Task 4 is to introduce you the simulation of these DRVs in Python. In this task, you are required to vary parameters for each distribution while keeping others constant and comment on how does this affect the shape of the distribution.
The distributions that you are supposed to work on are:

- Bernoulli Random Variable
- Binomial Random Variable
- Geometric Random Variable
- Poisson Random Variable

[20 Marks] Task 5: The Birthday Paradox

One of the benefits of using code to compute probability is that the time-consuming calculations can be skipped. This can be seen by the Birthday's Paradox Problem.
Problem Statement: Given there are n students in the class, assuming that none are not born in a leap year, find the probability that at least 2 people share a birthday.

The complement of this event i.e. n people don't share the same birthday is easier to compute and hence our desired probability can be written in terms of its complement.

$$P(\text{at least 2 people share}) = 1 - P(\text{none share})$$

The pattern becomes apparent when we see it for some n For $n = 1$

$$\frac{365}{365}$$

For $n = 2$

$$\frac{365}{365} \times \frac{364}{365}$$

For $n = 3$

$$\frac{365}{365} \times \frac{364}{365} \times \frac{363}{365}$$

and so on...
Generalizing this, for $0 \leq n \leq 366$

$$P(\text{none share}) = \prod_{k=1}^{k=n} \frac{365 - k + 1}{365}$$

Let the empty product be 1. Therefore the desired probability can be computed.

$$P(\text{at least 2 people share}) = 1 - \prod_{k=1}^{k=n} \frac{365 - k + 1}{365}$$

The code snippet in the notebook shows how it can be plotted in Python.
Task:

1. The graph is incomplete. It doesn't have any labels, add the labels and feel free to make it aesthetic.

2. When solving any problems regarding probability, we should be vary of our assumptions. In this problem, we made two assumptions (which is not stated) for computing the probability. State those assumptions in the notebook. (Hint for one of the assumptions: Our logic breaks in the case of twins)

3. If $n = 400$, $k = 380$, in the product will give us a negative value. Does it mean that this algorithm will fail for $k = 380$?. See whether it does. If it does, then fix the error in the code. If it doesn't state why and whether the computation for $k \geq 367$ is representative of something or not, if it isn't update the code to reflect this? After making the respective changes, draw the graph for $n = 500$

Ungraded: How to simulate an entire sample space

In the above scenario, we saw that if we know the formula for computing the probability, we can easily code it. This is useful in some situations, but this is not the only thing that can be done to make our lives easier. We can compute the entire sample space for combinatorial problems (of reasonable size).

As an illustration, we can use it on a friendship graph. Suppose that we know that 3 of the links have broken apart and each of the possible outcome is equally likely.

For example take the case of a network.

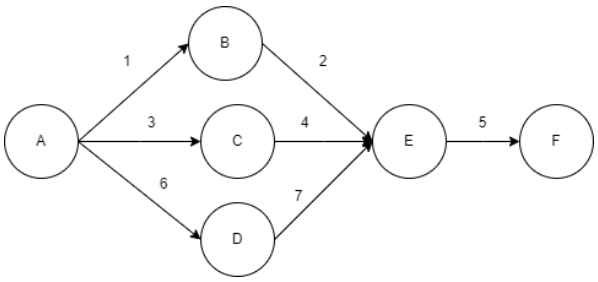

Figure 1: Network

Given a network of nodes and edges, if 3 connections break what is the probability that a signal can go from A to F. One thing we can do is generate all the possible combinations of lines that can be broken and assuming that each possible combination is **equally likely**and hence, compute which ones allow us to send the signal. This problem can be phrased in a different way: find the probability that there is still a path from A to F.

We can see from the diagram, that there are 3 paths to F

1. 1,2,5
2. 3,4,5
3. 6,7,5

If any of this path exists in the graph, then the element in our sample space is valid. The code in the notebook does this computation and prints the desired probability.
Note: We can probably implement Breadth First Search or Depth First Search to solve this problem. That would require us to store the vertices of the graph as well. That approach is better, since that can work on any arbitrary graph but for the purposes of illustrating the concept, this is the simpler approach.
Task: If you want, try making this problem work any arbitrary graph.