1 Random Variables

Mathematically, a *random variable* is a function $X: \Omega \to E, \omega \mapsto X(\omega)$ assigning to each outcome in the event space Ω some value in E. The *probability* that the random variable takes on a value in the set $A \subseteq E$ is defined as

$$P(X \in A) = \mathbb{P}(\{\omega \in \Omega : X(\omega) \in A\})$$

where $\mathbb{P}$ is a so called *probability measure*. The probability measure $\mathbb{P}(S)$ tells you what proportion of events in Ω are in S. For example, let X be a random variable with $X: [0,1] \to [0,5], \omega \mapsto 5\omega$. Now we ask what is the probability that X takes on a value in the interval $[4,5]$, so, $A = [4,5]$. Then

$$\{\omega \in \Omega : X(\omega) \in A\} = \{\omega \in [0,1] : 5\,\omega \in [4,5]\} = [0.8, 1]$$

To calculate $\mathbb{P}([0.8, 1])$, we ask what proportion of events from the interval $[0,1]$ are in $[0.8, 1]$. As you might guess, the answer is 20%.

There are multiple equivalent ways of conceptualising this idea. If you like, you can think of the probability measure of a set S as measuring what proportion of the volume of the event space will be mapped into the set S. Or, you can think of throwing darts at your event space and asking what proportion of darts land in your set S. Or, if you know about integrals, you can imagine integrating over the set S and dividing by the integral over the whole event space. It's worth thinking of the event space as a source of randomness and a random variable as a function which takes the source of randomness and maps it onto values which occur with a certain probability.

How do we implement the concept of a random variable on a computer? Most languages have implementations of *random number generators (RNG)* built in. RNGs allow you to generate random numbers in the interval $[0,1]$ – to throw a dart, if you like. To get samples from other random variables you'll need to take this source of randomness and transform it in such a way as to yield a distribution of values you want. For example, say you wanted a random variable which has a 60% chance of returning 1 and a 40% chance of returning 0. Then you would define a function $X: [0,1] \to \{0,1\}$ as follows

$$X(\omega) = \begin{cases} 1 & \text{if } \omega \in [0, 0.6] \\ 0 & \text{if } \omega \in (0.6, 1] \end{cases}$$

This way, 60% of the time the random number generator lands in $\omega \in [0, 0.6]$ and so $X(\omega) = 1$ 60% of the time. 40% of the time the RNG yields $\omega \in (0.6, 1]$ and so $X(\omega) = 0$ 40% of the time.

Exercise 1. Implement a function bernoulli(ω, p) which takes a value $\omega \in [0,1)$ and a probability p as inputs and returns *true* with probability p and *false* with probability $1 - p$.

Exercise 2. Implement a function uniform(ω, a, b) which returns a value between a and b, given a random number $\omega \in [0,1)$. All values in $[a, b]$ should be equally probable.

Exercise 3. Experiment with making other random variables and plot their probability distributions. Or combine ones you've already made by adding, multiplying, etc. Another direction you could go with this is to try to make it as hard as possible to guess which ω your random variable took as an input to generate a given output.

Exercise 4. If you add two random variables what probability distribution do you get for the sum? You probably don't have the mathematical tools to work this out precisely in general, but maybe you can get a qualitative answer (or try to think of some very simple random variables where you can work it out precisely).

For the two simple cases above it is possible to essentially guess what function will yield the random variable you want. Let us now try to understand a more complex case. Say you want to sample a random number in the interval $[0, \infty)$, such that the *probability density (pdf)* at each point is given by

$$\text{pdf}(x) = e^{-x}$$

where the probability density represents the amount of probability in an infinitessimal volume a the point x. Since, we are dealing with just a number, volumes are lengths on the real line. To figure out how to sample from such a distribution, we need to convert the probability density to a *cumulative probability density (cdf)*

$$\text{cdf}(x) = P(X \leqslant x) = \int\limits_{-\infty}^{x} \text{pdf}(x)\, dx$$

If you don't know what the funny sign on the right means, no need to worry. You don't need to know. In the case of the exponential distribution above, we have

$$\text{cdf}(x) = \int\limits_{0}^{x} e^{-x}\, dx = 1 - e^{-x}$$

Exercise 5. If you know about integration, calculate this integral yourself. If you don't know about integrals, but you know about derivatives, check that $\frac{d}{dx}\text{cdf}(x) = \text{pdf}(x)$ for the exponential distribution.

The trick to sampling from the exponential distribution is to invert the cdf function and use it as our random variable, so we define

$$X\colon [0,1] \to [0, \infty), \omega \mapsto \text{cdf}^{-1}(\omega)$$

with the inverse function cdf^{-1} defined such that $\text{cdf}(\text{cdf}^{-1}(\omega)) = \omega$ and $\text{cdf}^{-1}(\text{cdf}(x)) = x$.

Exercise 6. Work out the inverse of the cdf for the exponential distribution and implement the random variable. Plot the histogram and see if it looks right. Try plotting the exponential pdf on the same plot as the histogram.

Exercise 7. Why does this method work? How can we know that each value will be sampled with the right probability density? Does this method work for any random variable you want to sample?

Note that the implementation of the built in randexp function works differently, since there are more performant ways to sample from an exponential distribution. See here.

Finally, let me provide a method that works even for very complicated distributions which are hard to integrate or invert. The basic idea is to sample from a distribution you know how to sample from and then reject some samples, such that the proprtion of samples at a given value matches the probability density there. As an example, let's take the distribution

$$\text{pdf}(x) = \frac{2}{\pi}\sqrt{1 - x^2} \quad \text{for} \quad -1 \leqslant x \leqslant 1$$

Exercise 8. If you know how to integrate, check that this distribution is normalised, so $P(X \in [-1, 1]) = \int_{-1}^{1} \text{pdf}(x)\, dx = 1$ (Hint: substitute $u = \sin(\theta)$ and use $\sin^2(\theta) + \cos^2(\theta) = 1$ and the fact that the integral over one period of $\sin^2(\theta)$ is equal to the integral over $\cos^2(\theta)$). If you do not know how to integrate, you can still try to see what this function represents geometrically and convince yourself on geometric grounds that $P(X \in [-1, 1]) = 1$.

The algorithm to sample from this distribution works like so

1. Sample a value x uniformly between -1 and 1.

2. Sample a value y uniformly between 0 and 1.

3. If $y > \text{pdf}(x)$, reject the sample x and go back to step 1. Else, accept the sample and return x.

This algorithm is called rejection sampling and you'll see it show up again in this course a few times.

Exercise 9. Implement the algorithm. Plot a histogram of samples to check if it works.

Exercise 10. Why does this algorithm work?

Exercise 11. How might you use the algorithm for this particular distribution to numerically estimate the value of π?

2 Monte Carlo Simulation of Random Walks

We begin with an ensemble of intial positions x_0. Then, for each position, we update the position by a random rule. For example,

$$x_{n+1} = x_n + \begin{cases} 1 & \text{with probability } \frac{1}{2} \\ -1 & \text{with probability } \frac{1}{2} \end{cases}$$

2.1 The Central Limit Theorem

Theorem 1. (Central Limit Theorem) *Let $\{X_1, \ldots, X_n, \ldots\}$ be independent and identically distributed random variables with finite mean μ and finite variance σ^2. Then, the distribution of*

$$Z_n = \frac{X_1 + \cdots + X_n}{n}$$

converges to a gaussian distribution with mean μ and variance $\frac{\sigma^2}{n}$.

2.2 Discrete Random Walk

2.3 Diffusion and Superdiffusion

2.4 Drift and Diffusion in a Box, in a Well and in a Periodic Well

2.5 Optimal Betting Strategy – The Kelly Criterion

3 Markov Chains and a bit of Linear Algebra

4 Nash Equilibrium and Poker