Probability Puzzle #1: The Unfair Coin

Problem Description

Summary

Using an unfair coin, and without knowing the actual probabilities of the coin landing heads-up or tails-up, is it possible to simulate flips of a perfectly fair coin?

Definitions

- A *fair coin* is one which has equal probabilities of landing heads-up and tails-up when flipped. Ignoring the extremely small probability of landing on an edge, we have

$$
\begin{aligned}
p(\text{heads}) &= p(\text{tails}) \\
p(\text{heads}) + p(\text{tails}) &= 1 \\
p(\text{heads}) + p(\text{heads}) &= 1 \\
2\,p(\text{heads}) &= 1 \\
p(\text{heads}) &= 1/2 \\
p(\text{tails}) &= 1/2
\end{aligned}
$$

  An *unfair coin* is one which has unequal probabilities of landing heads-up and tails-up when flipped.

- A *Bernoulli trial* is a random experiment with 2 possible outcomes, generally designated as *success* and *failure*, or as the corresponding numeric values 1 and 0. Depending on the underlying phenomena, the probability of success can be any value between 0 and 1, inclusive. Symbolically, we express this as $0 \le p(\text{success}) \le 1$.

  One simple example of a Bernoulli trial is a single coin flip, where we might arbitrarily designate the coin landing heads-up as success, and landing tails-up as failure. Whether the coin is fair or unfair, each flip is a Bernoulli trial, with $p(\text{success}) = p(\text{heads})$.

- A *Bernoulli process* is a sequence of *independent* Bernoulli trials with a constant (though not necessarily known) probability of success. By *independent*, we simply mean that the outcome of any given trial has no influence on the outcomes of any other trial.

  The output of a Bernoulli process is the sequence of outcomes of the individual trials, in the same order. Note that recording this sequence of individual Bernoulli trial outcomes is not the same as simply tallying the number of successes, or – equivalently – summing the numeric (0 or 1) values of the trials.

Scenario

You've been given a coin that you're told is not fair: The probability that the coin lands heads-up when flipped is not equal to 0.5; however, you don't know what the actual probability is.

You must use this unfair coin (and no other) to implement a Bernoulli process of length 100 – i.e. to conduct a sequence of 100 Bernoulli trials. Complicating matters, the individual trials in your Bernoulli process must have $p(\text{success}) = 0.5$.

How will you go about completing your task? How can you simulate a fair coin with an unfair one – and do so without knowing how unfair the coin is? Is this even possible?

Guidelines

Restrictions and assumptions

- Even with the unfair coin, each coin flip is independent of all others.
- There's no change in the physical properties of the coin or the surrounding environment over time. Thus, the probability of landing heads-up on a single flip remains constant, no matter how much time passes – and no matter how many times you flip the coin.
- You can't employ any special techniques or tricks to bias the outcome of coin flips.
- The coin is the only source of randomness that you're allowed to use.

  The probability of success in each of your Bernoulli trials must be *exactly* 0.5.

Hints

- Attempting to estimate the actual probability of the unfair coin landing heads-up (e.g. by flipping the coin many times and tallying the results) won't be of much use in solving the problem. In any event, no matter how many times you flip the coin, an estimate of the probability of success based on that sample would still be just that: an estimate.
- When using coin flips to illustrate or implement Bernoulli trials and processes, we typically define a trial as a single coin flip; however, for some purposes, we might instead treat multiple coin flips as a single trial.
- Sometimes, when transforming from a source of random events or values (e.g. a Bernoulli process with $p(\text{success}) \neq 0.5$) to another (e.g. a Bernoulli process with $p(\text{success}) = 0.5$), we might consider some outcomes (or combinations of outcomes) from the first to fall outside the bounds of usable input for the transformation; in that case, we simply transform the usable outcomes, and discard the rest. This technique is called *rejection sampling* or the *acceptance-rejection method*.