

## Make a Fair Coin from a Biased Coin

January 3rd, 2018.

If we have a biased coin, i.e. a coin that comes up heads with a probability not equal to  $\frac{1}{2}$ , how can we simulate a fair coin? The naive way would be throwing the coin 100 times and if the coin came up heads 60 times, the bias would be 0.6. We could then conduct an experiment to cope with this bias.

A much better solution was introduced by John von Neumann. We are going to consider the outcomes of tossing the biased coin twice. Let  $p$  be the probability of the coin landing heads and  $q$  be the probability of the coin landing tails - where  $q = 1 - p$ .

Imagine, we have a coin with  $p = 0.6$ . If we toss the coin twice and the coins faces are different, the probability would be either  $P(HH) = P(H) \cdot P(T) = 0.36$  or  $P(TT) = P(T) \cdot P(T) = 0.16$ , which does not reveal anything. But if the tosses are different, the probabilities are the same, that is  $P(HT) = P(H) \cdot P(T) = 0.24$  and  $P(TH) = P(T) \cdot P(H) = 0.24$ :

	Heads	Tails
Heads	$P(HH) = 0.36$	$P(TH) = 0.24$
Tails	$P(HT) = 0.24$	$P(TT) = 0.16$

That means when the two tosses are different, we can use the outcome of the first coin and throw away the second. If the two tosses are the same, we disregard them and start over until we find two different tosses. Since we do not consider all cases where the outcomes are the same, it doesn't change the result. So even if we have  $P(HT) = P(TH) = 0.24$ , it does not matter. The outcome has an equivalent probability, even if we have to wait for it a little longer. Von Neumann described this procedure like this:

1. Toss the coin twice.
2. If the outcome of both coins is the same (HH or TT), start over and disregard the current toss.
3. If the outcome of both coins is different (HT or TH), take the first coin as the result and forget the second.

It is important to note, that we did not have to know anything about the coins bias, other than the results are independent and identically distributed (IID) - which means it must stay the same coin - and there is a chance to win at all. The remaining question is, how many tosses are necessary on average until we get the desired outcome?

## Expected number of Coin Tosses

Lets call the probability of seeing two different outcomes in a biased coin flip  $P(\text{different}) = 2pq$  and the expected number of tosses until that event happens  $E_t$ . If we now succeed in the first round, we use exactly 2 flips. If we do not, we have flipped the coin two times and because it is as though we have to start over from the beginning again, where the expected remaining number of flips is still  $E_t$ . Hence

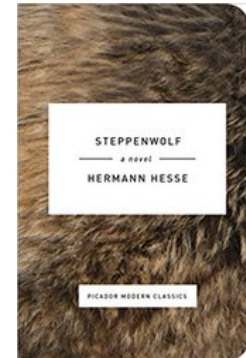
$$\begin{aligned}
 E_t &= 2P(\text{different}) + P(\text{equal}) \cdot (2 + E_t) \\
 &= 4pq + (1 - 2pq) \cdot (2 + E_t) \\
 &= 4pq + E_t + 2 - 2pqE_t - 4pq \\
 &= \frac{2}{2pq} \\
 &= \frac{1}{p(1-p)}
 \end{aligned}$$

You should Follow me!

- [Facebook](#)
- [Github](#)
- [Twitter](#)
- [RSS Feed](#)

A book a week challenge

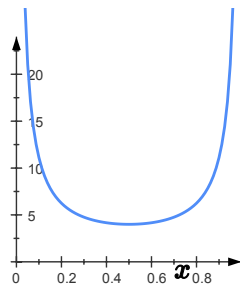
This week:



My Rating: ★★★★★

See [previous weeks](#)

When we plot this result as a function on the interval  $[0, 1]$ , we see that moving from an unbiased coin ( $p = 0.5$ ), the number of required trials increases dramatically:



## Implementation and Simulation

This whole idea can be implemented quite easily. Lets say we have a function  $\text{coin}(p)$ , which bias towards probability  $p$ . We can then make a fair coin of it with the following snippet:

```
var coin = p => () => Math.random() < p;

function fairCoin(coin) {
  do {
    var a = coin();
    var b = coin();
  } while (a == b);
  return a;
}

fairCoin(coin(0.6));
```

When we simulate the toss, we will see an outcome of 50:50:

```
var heads = 0, tails = 0;
for (var i = 0; i < 10000; i++) {
  if (fairCoin(coin(0.6))) heads++;
  else tails++;
}
console.log(heads / (heads + tails)); // 0.5005
```

## Related work

M. Mitzenmacher and D. Kozen went a little further by discussing strategies to simulate a fair coin from a biased coin that are optimal in the expected number of coin flips [1][2], like for example, tossing 4 times per round, which can be used in a similar way to the method of von Neumann. Q. Stout et al. uses tree algorithms to make a fair coin out of a biased coin [3] and P. Diaconis et al analyzes the dynamics of a coin toss [4].

- [1] M. Mitzenmacher (2009) [Tossing a Biased Coin](#)
- [2] D. Kozen (2014) [Optimal Coin Flipping](#)
- [3] Q. Stout, B. Warren (1984) [Tree Algorithms for Unbiased Coin Tossing with a Biased Coin](#)
- [4] P. Diaconis, S. Holmes, R. Montgomery (2004) [Dynamical Bias in the Coin Toss](#)

## You might also be interested in the following

- [How to plot a covariance error ellipse](#)
- [Calculate the sum of divisors](#)
- [Calculate the intersection point of two Lines](#)
- [Cracking Apple Interview Question](#)

Sorry, comments are closed for this article. Contact me if you want to leave a note.