

# PHSX815: Computational Methods in Physical Sciences

## Project 2

### Project Paper

#### Title: Two Hypotheses for the Random Walk Experiment

Vishakha Lingadahally

May 16, 2021

## Introduction

In Science, we are often interested in a set of outcomes rather than a single event. Stochastic simulations allow us to generate and examine a series of simulations of a system in which the steps are governed by random choice. A simple algorithm for flipping of coin can be a useful tool to 'choose' if a step in a phenomenon will occur or not. Consider a drunkard trying to walk along the pavement in the middle of the night. In their drunk state of mind, they can only move sideways-either to the left or the right from their current position. Whether they take a step to the left or the right can be dictated by flipping a coin and checking if the outcome is 'Heads' or 'Tails'. The philosophy behind the seemingly simple thought experiment has a wide range of application in physics, chemistry as well as biology; to describe the diffusion of gas molecules, simulate bacterial motion and learn the pattern of genetic drift.

In Project 1, we explored 1D random walk with the hypothesis of equal probabilities of stepping on either side, i.e., the coin had equal probabilities of showing either heads or tails. In Project 2, we explore two different hypothesis:

- Equal probabilities of the drunkard stepping on either side of the pavement (in the code, `step_prob=0.5`)
- Unequal probabilities of the drunkard stepping on either side of the pavement, i.e., the probability of the drunkard steps on either the left or the right of the pavement is greater than the other scenario.

The likelihood of the occurrence of each of the hypotheses is computed using the likelihood for a binomial probability distribution, which allows us to calculate the likelihood ratio for the two cases. The likelihood of a binomial distribution is given by the following expression:

$$L(p|n) = \frac{n!}{k!(n-k)!} p^{(k)} q^{(n-k)}$$

where in our context,  $n$  is the total number of steps taken,  $k$  is the number to steps taken to the right,  $p$  is the probability of stepping to the right and  $q$  is that stepping to the left.

## Understanding the code

### Flipping the coin

The first section of the code consists of importing the necessary modules for computation and plotting. We then begin by flipping the coins and understanding their output. To see why this is relevant in random walk, think of a gas molecule that has equal probabilities of moving either to the left or the right. To dictate whether the molecule moves to the left or the right by flipping a coin. This will obviously change every time that we run the code cell. To convert this to a 'heads' and 'tails' readout, we can assume that this is a totally fair coin. This means that the probability of getting 'heads' to get  $P(H)$  is the same as flipping a 'tails'  $P(T)$  such that  $P(H)+P(T)=1$ . This means that for a fair coin,  $P(H)=P(T)=0.5$ . To convert our coin flips to 'heads' or 'tails', we simply have to test if the flip is above or below 0.5. If it is below 0.5, we say that the coin was flipped 'heads', otherwise, it is 'tails'.

### Testing if the coin-toss is fair

Now, in order to check if our coin-toss is fair, we toss it multiple times, say 1000 times. Let's flip a coin one thousand times and compute the probability of getting 'heads'. The simulated probability is very close to our expected  $P(H)=0.5$ , but not exactly. This is the nature of stochastic simulations-it's based on repetitions. If we were to continue to flip a coin more number of times, our simulated  $P(H)$  would get closer and closer to 0.5. This is why doing many repetitions of stochastic simulations is necessary to generate reliable statistics.

### The random walk

We start at position zero and flip a coin at each time step. If it is less than 0.5, we take a step left. Otherwise, we take a step to the right. At each time point, we keep track of our position and then plot our trajectory.

In order to compute the the number of steps taken to the right and that taken to the left, we solve the simultaneous equations  $n_l + n_r = 100$  and  $n_l - n_r = \text{final position}$ , where  $n_l$  is the number of steps taken to the left and  $n_r$  is the number of steps taken to the right. We plug these values in the likelihood for our binomial distribution, which can be stated as "The likelihood of the probability of stepping to the right, given that the drunkard takes  $n_l$  steps to the left and  $100 - n_l$  steps to the right. The same procedure is used in both of our hypotheses. We then compute the likelihood ratio of the two hypothesis and subsequently the p-value.

As our steps are based on the generation of random numbers, this trajectory will change every time the code is run. The power of stochastic simulation comes from doing them many times over. The next step would be to run the steps, say, one thousand times and plot all of the traces. We plot the mean at each time point as a blue line.

## Understanding the output

The user sees the following results as the output of the code:

- The results of the of three consecutive coin flips that our code simulates in the beginning is displayed. Then, these probability outputs are converted to 'heads' and 'tails'.
- Then, the predicted and simulated probabilities are then displayed. The predicted probability, naturally, is 0.5 and the simulated probability is the ratio of the number of heads to that of tails.
- The number of steps taken to the left and the number of steps taken to the right are computed for our first hypothesis.
- Next, the likelihood for our first hypothesis, i.e., the likelihood for the binomial distribution with equal probabilities of stepping to either to the left or the right is computed.
- The number of steps taken to the left and the number of steps taken to the right are computed for our second hypothesis.
- The likelihood for our second hypothesis, i.e., the likelihood for the binomial distribution with unequal probabilities for stepping on the left or the right is computed.
- Then, the likelihood ratio for our two hypotheses is computed.
- Finally, we compute the p-value.
- The plots of the tracks of the random walk for both the hypotheses are generated, along with the random walk simulated multiple times ( $n=1000$ ).

## Output interpretation and further scope

Let us take a look at the probability distribution curves for the two hypotheses. For the first hypothesis,  $p=q=0.5$ . A sample experiment for this hypothesis would produce a result of almost equal steps to the left and to the right.

A typical plot for an isolated experiment would look as follows:

When such an experiment is repeated a large number of times, say 1000 times, we obtain a set of tracks that would look something like this:

The mean is a straight line passing through the origin.

Let us take a look at an alternate hypothesis, say  $p=0.4$  and  $q=0.6$ , and plot an isolated track and the set of tracks for repetitions of the experiment.

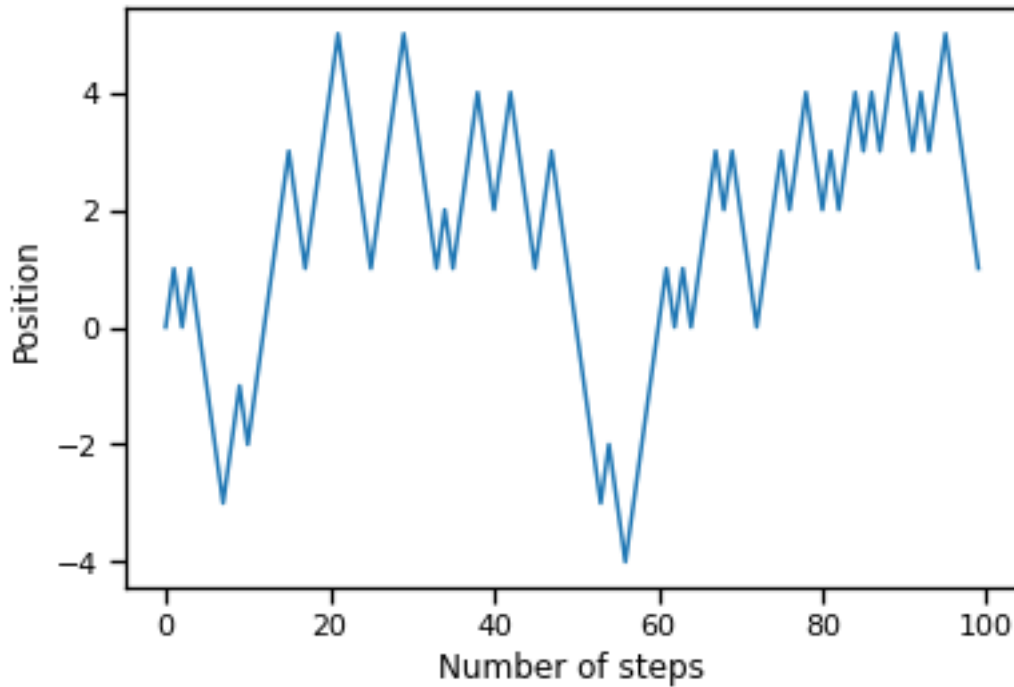


Figure 1: Track for  $p=0.5$

We can observe how the mean is shifting towards the negative part of the position axis, with steps.

Similarly, with varying value of  $p$ , we find the mean tilted either to the right or the left, depending the hypothesis. As far as the  $p$ -value is concerned, we find it to be almost 0 when the alternate hypothesis has the value of probability of stepping to the right of 0.1 or 0.9, i.e., at the extreme ends. The  $p$ -value is 1 when the alternate hypothesis considers the probabilities of stepping to the right and to the left to be equal. This indicates that the statistics with equal probabilities of stepping to the left or to the right is most likely.

## References:

- [github.io](https://github.io)
- [stackoverflow.com](https://stackoverflow.com)
- [medium.com](https://medium.com)
- [towardsdatascience.com](https://towardsdatascience.com)

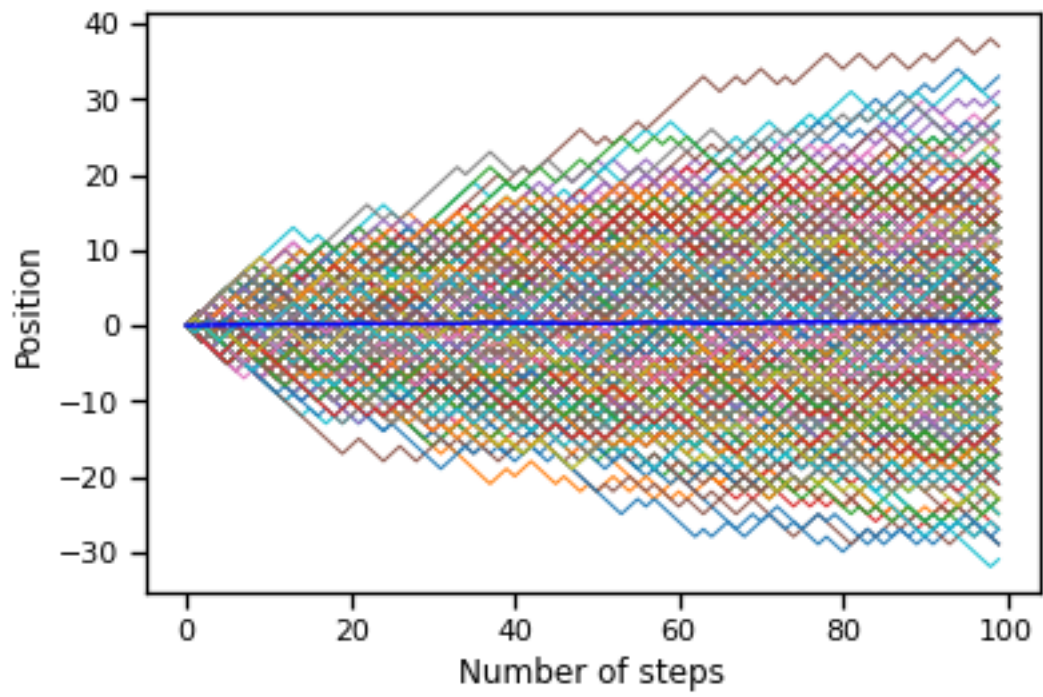


Figure 2: Set of tracks for  $p=0.5$

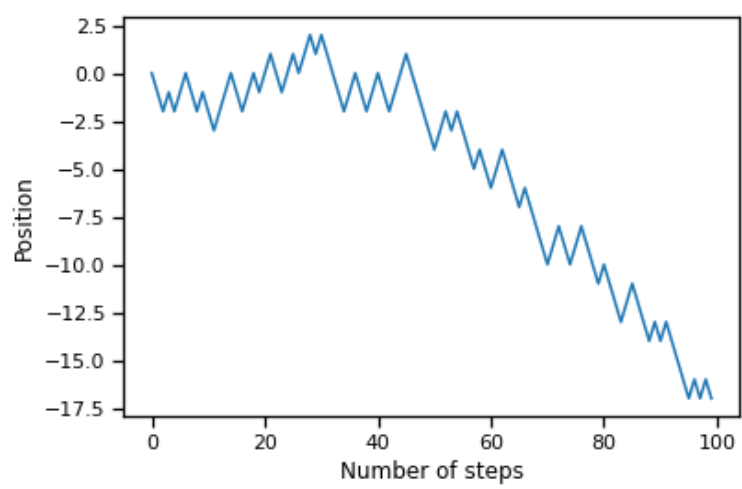


Figure 3: Track for  $p=0.4$

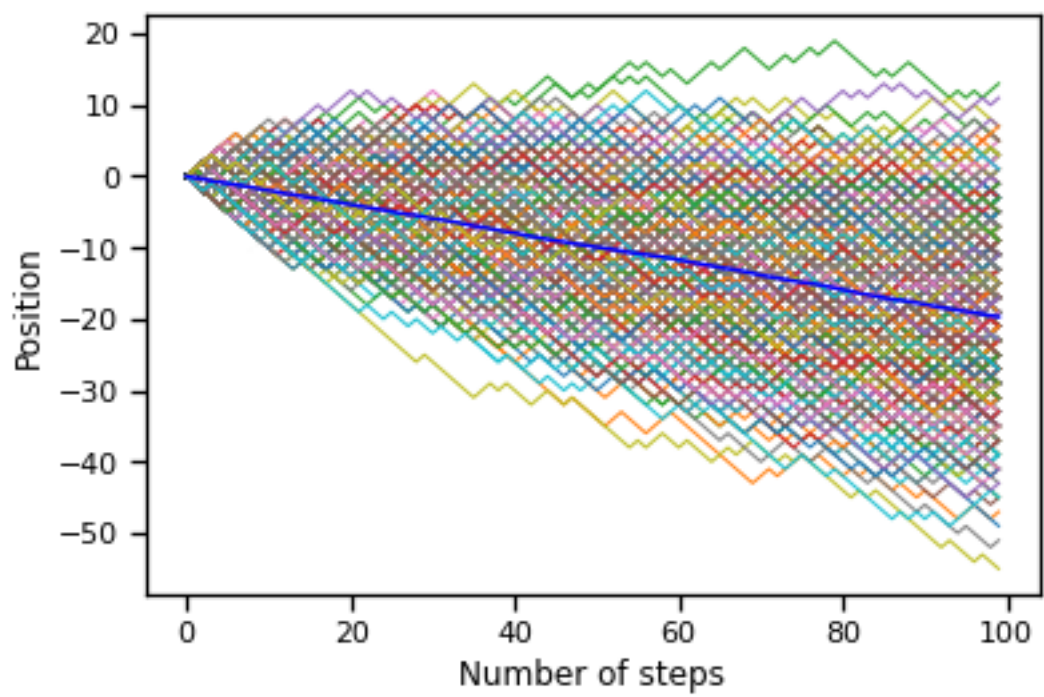


Figure 4: Set of tracks for  $p=0.4$