Imagine you are walking in the empire state building and throwing the dice.

Suppose 100 times you are throwing the dice. If its 1 or 2 you will go 1 step down and if its 3, 4 or 5 you will go one step up and if its 6, you will throw the dice again and walk the resulting number on dice.

Of course, you can't go below step 0 and there is 0.1 % chance of falling down the stairs now I Bet you: You will reach step 71. What is the chance you will win this bet?

How to solve?

There are two way to solve this problem

1. Analytical
2. Simulate the process

I am going to opt for second approach.

In this project I will take all the input from users.

Please give wide range of number as an input

* To get clear idea about chances of reaching the 71th Step
* To understand when to use line plot and when histogram.

Prerequisite: Install numpy and matplotlib

Steps follows to solve:

1. First, we need random generator so we can stimulate the dice – I am using random package a sub package of numpy, inside random package there is randint () function which I will generate random number, As a result a pseudo-random numbers generated - which computer generate using mathematical formula starting from a seed.
2. Setting the seeds – This seed is set by python by default; however, we can also set it manually – I will be setting the seeds manually.

For the Same seed: Same random number will get generated (That’s why it’s called pseudo random number). Its random but consistent between runs It will ensure reproducibility.

1. Random walk – Bunch of random steps will get converted into random walk.
2. How low can we go – We can’t go below 0

To solve this, use max () function with 2 arguments – The biggest one gets return.

1. Simulate multiple walks – To increase the chances of reaching 60 steps, simulate multiple walks.
2. There is 0.1% chance of falling down – To solve this, generate a random float between 0 and 1. If this value is less than or equal to 0.001, you should reset step to 0.
3. Visualize all walk – Choose how you want to visualize the walk.
4. Using matplotlib – to build line plot – Use matplotlib if simulating random walk for a smaller number of simulation (Let’s say 10 to 20). Using matplotlib for huge random walk simulation will cause clumsiness
5. Using histogram – to plot the distribution – With histogram we can visualize end point of all the random walks, which we have simulated.

Any feedback is welcome!!