14. Dice and Coin Game Simulation

Group 337 – Jason Pappalardo

Abstract

This study aims to simulate a two-player game in which two players take turns rolling a six-sided die and following corresponding instructions based on the outcome of the roll. The rules are deterministic and there is no decision-making element to the game, thus, it is easily simulated using a pseudo-random number generator and variables that store the defining attributes of the game. The methodology of this project is completed in Python through a Jupyter notebook. The foundation of the simulation is the 'random.uniform' function from the 'random' package in Python. Random samples are taken from a Unif[0, 1] distribution and are inverted to simulate a die-roll using the Inverse Transform Theorem. The game is simulated 100,000 times and the cycle length of each game is recorded and stored for analysis. The experiment uncovered that the distribution of cycle length resembles a distribution as follows: $Y \sim 4 + \text{Geom}(0.057)$.

Project Description

This section will explain the details of the simulated game. There are two players in the game, player A, and player B. Player A executes their turn first, followed by player B, then player A, ... and so on. At the start of the turn, the player rolls a six-sided die, which is simulated by a pseudo-random Unif[0, 1] which we will call n. The formula for the die roll is the ceiling function of 6*n. The outcome of the die roll will determine the action the player will take. Each player begins with four coins. There is a 'pot' containing two coins at the start of the game. If the player rolls a one ($p = 1/6$), the player does nothing. If the player rolls a two ($p = 1/6$), the player takes all of the coins in the pot. If the player rolls a three ($p = 1/6$), the player takes half or the coins in the pot, rounded down (floor function). If the player rolls a four, five, or a six ($p = ½$), then the player places a coin in the pot from their possession.

The game concludes when a player rolls a four, five, or six and cannot place a coin in the pot because they have no coins in their possession. This player loses the game and the opposing player is declared the winner. In this experiment, we define a 'cycle' as a pair of turns completed (one from each player). We also define the last turn as a complete cycle in the event that the game ends on the turn of player A. We aim to derive the expected number of cycles and the distribution of cycle length as a random variable.

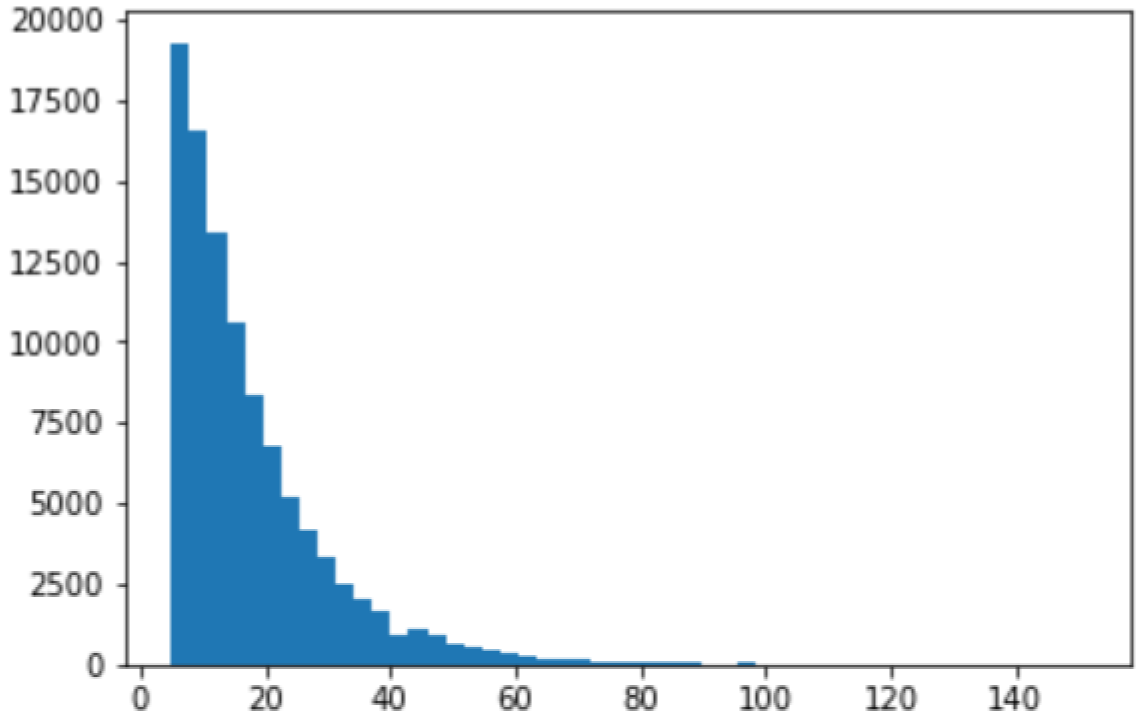
Methodology

A singular Python function is enough to simulate a single run of the game described above. We begin by initializing variables for the coin inventories for player A, B, and the pot. We also initialize a turn counter variable and increment with the conclusion of each turn. At each turn, we generate a new variable representing the outcome of the die roll and adjust the values of the above variables in correspondence with the die roll. Print statements are utilized at the end of each turn to ensure that the coin values are updating correctly. Use the even/odd attribute of the turn counter to cycle between turns for player A and B. We initialize the loop to run in perpetuity, and manually break the loop when a player rolls and four, five, or six and cannot complete the task due to an empty coin inventory. At the conclusion of the loop, the function returns the ceiling function of t/2, where t is the total number of turns observed.

Once the code has been tested and cleared of errors, we can run the code over a large sample and record the cycle length of each run in a list. We completed 100,000 runs and observed a processing time of about 3-4 seconds.

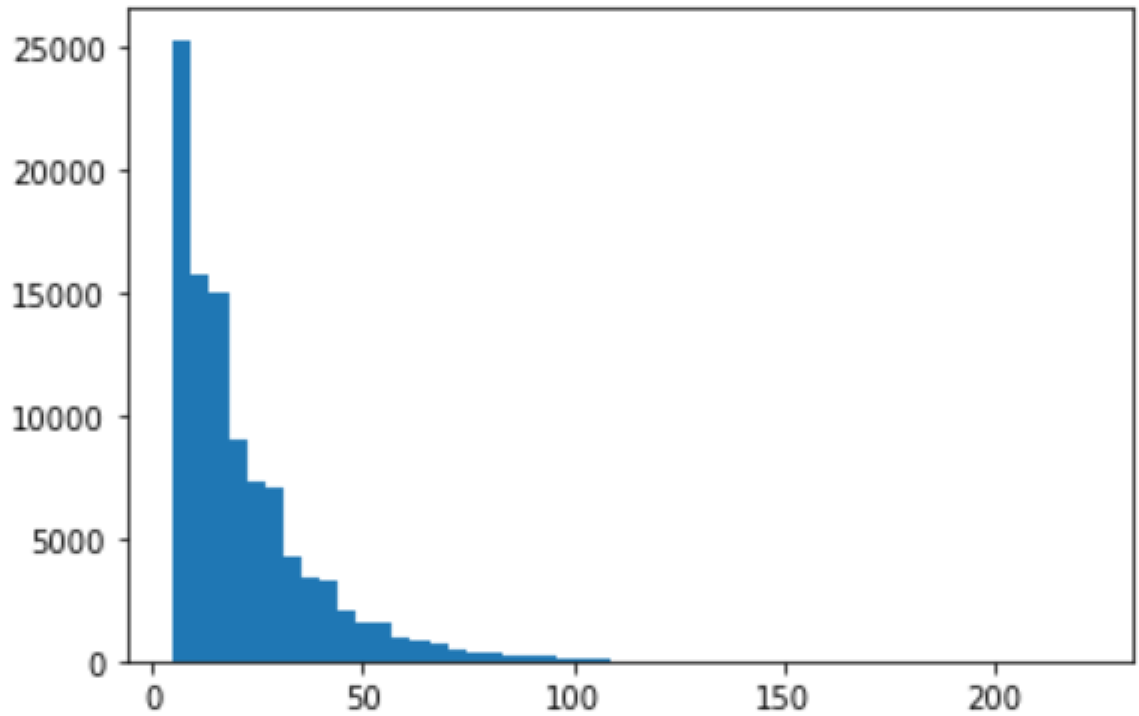
The minimum cycle length is five. Since each player begins with four coins, it takes four consecutive turns of observing a roll greater than or equal to 4 to deplete the coin inventory of a single player. Thus, if the player observes a roll greater than or equal to four on the fifth consecutive turn from the beginning of the game, the game will end. This is the quickest method to ending the game, thus, we observe a minimum cycle length of five. The maximum observed cycle length is 151, however, we expect this random variable to have a large variance. The analytical derivation for the distribution of the maximum of X1, X2, …, Xn ~ Geom(p) random variables does not have a nice solution and is beyond the scope of this project. The observed mean of our sample is 17.59. Since the algorithm is relatively fast to generate 100,000 random variables for cycle length, I ran it many times and the mean is relatively stable around 17.5.

We can use the mean cycle length to estimate the expected value of cycle length. First, we must choose a distribution in which to model cycle length. Cycle length is a discrete random variable, so we limit our options to discrete statistical distributions. Observe the histogram below.



The distribution of cycle length is naturally representative of a geometric distribution. The geometric random variable observes potential values of positive integers greater than 0, i.e. {1, 2, 3, …}. Since our random variable of cycle length observes a minimum value of five, we simply shift the distribution by four integers on the x-axis. Our random variable for cycle length will be modeled as the following moving forward: Y ~ [4 + Geom(p)].

Now we must estimate p. The maximum likelihood estimator (MLE) for the geometric distribution is 1/x-bar, where x-bar is the sum of the observed random variables divided by n (number of trials). Using this estimator, our estimate of p, or p-hat, is approximately 0.0569. Our distribution can now be written as Y ~ [4 + Geom(0.0569)]. See the image below for the probability mass function of this distribution.



The last step of this experiment is to test the validity of our estimator. We accomplish this using a chi-squared goodness of fit test. We perform the test with $k = 114$ after matching values from the cycle length distribution with the pmf for Y. The chi-squared value is 5,896, which is significantly higher than the test-statistic with a p-value of 0.05 with 112 degrees of freedom, which is 137.701. Therefore, we must reject the null hypothesis that the distribution of cycle length fits a geometric distribution with $p = 0.0569$.