ISYE 6644 Simulation: Project Progress Report

Topic 14: Dice and Coin

Group 54 – Wenyu Sui

Abstract

This work tries to determine the expected value and the probability distribution of the number of cycles that a Dice and Coin game will last for. Suppose that for each new game, the number of cycles follows an identical independent distribution, and the game could be simulated numerous times by computer using Python programming language. According to the Central Limit Theorem, the true expected number of cycles that the game will last for can be estimated using the mean number of cycles of all the simulated games, provided the sample size of the simulated games is large enough.

Overview

Here is an overview of the rules of the Dice and Coin game that we try to simulate.

There are two players in the game, A and B. Each player will have 4 coins at the beginning of the game, and there will be another two coins in the pot. After the game starts, the players will toss a 6-sided dice in turns, starting from player A. Each player needs to do the following activities when they roll different numbers.

- If the player rolls 1, the player will not do anything.
- If the player rolls 2, the player will take all the coins in the pot.
- If the player rolls 3, the player will take half of the coins (rounded down) in the pot.
- If the player rolls 4 - 6, the player will put a coin back to the pot.

The game will end only if one player rolls 4, 5 or 6 and the player does not have enough coins to put back into the pot. The player who fails to put back a coin into the pot will lose the game. A "cycle" is defined as both player A and B completing their games. The exception is that if one player loses the game, the game will end, but it will still be counted as a cycle (the last cycle of the game).

The purpose of this project is to determine the expected number (and the probability distribution) of cycles that the Dice and Coin game will last for by using simulation. The following sections of this progress report are outlined as follows:

- Proposed research methodology (including theory and application)
- Current project progress
- Next steps

Proposed Research Methodology

Theory

According to the rules of the game, each new game is played under the same rules and settings, and each new game is independent from all the other games. Therefore, we can assume that the number of cycles of a new game, X , follows an independent and identical distribution with mean value μ , and variance σ^2 .

Suppose that n games are played. The number of cycles of each game can be denoted as X_1 , X_2 , X_3 … X_n. According to the Central Limit Theorem, if the sample size n is large enough, the sample mean $\overline{X_n} = \sum_{i=1}^{n} X_i/n$ will follow a normal distribution with mean value μ , and variance $\frac{\sigma^2}{n}$. Therefore, $\overline{X_n}$ can be used as an unbiased estimator of μ. Its accuracy of estimation will increase as n increases, because the variance of $\overline{X_n}$ will get smaller when n gets larger.

In order to estimate μ using simulation, we need to determine a sample size n , which is the number of simulations that we need to run. The sample size needs to be large enough to avoid significant variances of the estimated value $\overline{X_n}$. We will record the number of cycles of each simulated game, X_1 , X_2 , X_3 … X_n. After running n simulations, we will calculate the sample mean $\overline{X_n}$, and use it as an estimate of μ . The probability distribution of random variable X can also be estimated by observing the sample distribution of X_1 , X_2 , X_3 … X_n . The next section ("Application") of this report will discuss how to implement this research method in detail.

Application

The simulations and result analysis will be executed by a Python script in Jupiter Notebook. The script will contain the following parts:

1. Define a Function that Simulates the Dice and Coin Game

In this section, a python function will be defined to simulate a new game. The function will perform the following activities every time when it is called:

1) **Initialized the game.** Three new variables will be set up, representing the number of coins owned by player A, the number of coins owned by player B, and the number of coins in the pot. The values of these variables will be initialized to make sure that each player will start the game with 4 coins, and that there are 2 coins in the pot.
2) **Simulate the game.** In each cycle of the game, the python function will simulate the process of rolling a 6-sided dice by using the random number generator, $random.uniform()$, from $random$ library. When each player rolls the dice, the random number generator will generate a floating-point number from a uniform distribution, $Unif(0,6)$. This number will be rounded up to its nearest integer. Therefore, we can derive a random integer ranging between 1 to 6, with each integer having the same probability to be generated. After the random integer is generated, the python function will then simulate the player's activity by updating the number of coins owned by player A, the number of coins owned by player B, and the number of coins in the pot. The script will repeat this process for player A and player B alternatively in each cycle until one player loses the game. After the game is ended, the python function will stop running, and it will return the number of cycles of the game as an integer.

2. Determine the Sample Size of Simulated Games

As discussed in the "Theory" section, the variance of $\overline{X_n}$ will decrease as the sample size n increases. Therefore, we need to determine a proper value of n so that after we run the simulation n times, the value of $\overline{X_n}$ can be used as a reliable estimate for the true value of μ (the expected number of cycles).

In the python script, we will try to run the simulation with sample size 10, 100, 1000, 10^4, 10^5 and 10^6 . At each level of sample size, we will run the simulation 40 times and calculate 40 different values of $\overline{X_n}$. For example, if the sample size is 100, we will simulate 100 different games, calculate $\overline{X_n}$, and

repeat the same process 40 times. We will get 40 different $\overline{X_n}$ s, and each $\overline{X_n}$ will be derived from 100 different games. After repeating this process with different sample sizes, we will derive 40 $\overline{X_n}$ s with sample size 10, 40 $\overline{X_n}$ s with sample size 100, 40 $\overline{X_n}$ s with sample size 1000 … and 40 $\overline{X_n}$ s with sample size 10^6.

For each sample size level, we can calculate the sample variance of $\overline{X_n}$, using the 40 values of $\overline{X_n}$ derived from the simulations. Then we can compare the sample variance of $\overline{X_n}$ at different sample size levels. Finally, we can choose a proper sample size level so that the variance of $\overline{X_n}$ will be small enough and will not affect the accuracy of our estimate.

3. Execute Simulations

After the value of n (sample size) is determined from the previous step, the python script will then simulate n different games by calling the function defined in step 1. The number of cycles of the simulated games X_1 , X_2 , X_3 … X_n will be stored in a data frame.

4. Analyze the Results of Simulations

After n games are simulated, the sample mean $\overline{X_n} = \sum_{i=1}^{n} X_i/n$ can be calculated using the simulation results. We would use the value of $\overline{X_n}$ as an estimate of the true value of μ (the expected number of cycles). The confidence interval of μ can also be calculated since we know that $\overline{X_n}$ follows a normal distribution.

More descriptive statistics can be generated from X_1 , X_2 , X_3 … X_n, in order to better depict the distribution of random variable X (the number of cycles of a game). These statistics may include sample variance, standard deviation, quantiles, etc. We could also create a histogram of X_1 , X_2 , X_3 … X_n, so that we could observe the distribution of X in visuals.

If time allows, we could also use Kolmogorov–Smirnov test in python to test if the sample distribution of X_1 , X_2 , X_3 … X_n fits any widely seen probability distributions such as normal distribution, exponential distribution, Poisson distribution, etc.

Current Progress

1. The research method has been determined and described in the section "Proposed Research Methodology".
2. The python function which simulates a new Dice and Coin game (Application.1) has been finished.
3. The python script which determines the appropriate sample size n (Application.2) is 50% finished.

Next Steps

1. Finish the rest of the python script to:
    1) Determine the appropriate sample size n.
    2) Simulate n games and record the number of cycles X_1 , X_2 , X_3 … X_n .
    3) Analyze the simulation results by calculating the sample average number of cycles $\overline{X_n}$, calculating the confidence interval of μ , and generating other descriptive statistics.
    4) Perform any further analysis as needed.

2. Finish writing the final project report.