Martingale

Josh Adams
jadams334@gatech.edu

Abstract—Martingale, simple gambling simulator! This simple gambling simulator is based around the American[1] roulette wheel which the odd numbers from 1 to 10 and 19 to 28 are red and even are black, the odd numbers in 11 to 18 and 29 to 36 are black and even are red. The American roulette wheel varies from other roulette wheels by containing one extra value, the double-zero (Wikipedia, 2020). Martingale simple gambling simulator will make successive spins with some arbitrary chance of winning which will be calculated and explained later in the paper.

1 EXPERIMENT ONE – EXPLORE THE STRATEGY

Experiment one takes the Martingale simple gambling simulator and makes consecutive bets on black numbers. One stipulation is that once the total winnings for a simulation reaches $80, the simulation will end. The value $80 is input for the remaining spins. The American roulette wheel has a total of 38 different possible values which could be selected. Out of those 38 there are 18 red numbers and 18 black numbers. With two green numbers, the zero and double-zero, both do not count as winning numbers. The chance of winning a spin during any simulation is 47.4% based on the calculation below.

$$\left(\frac{1}{Total\ Number\ of\ Values}\right) * Total\ Winning\ Values$$

$$\left(\frac{1}{38}\right) * 18 = 0.47368 * 100 = 47.4\%$$

1.1 Figure One

Figure one was used to explore the Martingale simple gambling simulator. Each simulation will have at most 1000 spins but will end early if total winnings reach the goal of $80. Ten different simulations were executed, and all the simulations performed similarly. On average, each of the simulations reached the goal of $80 within spins ranging from 150 to 200.

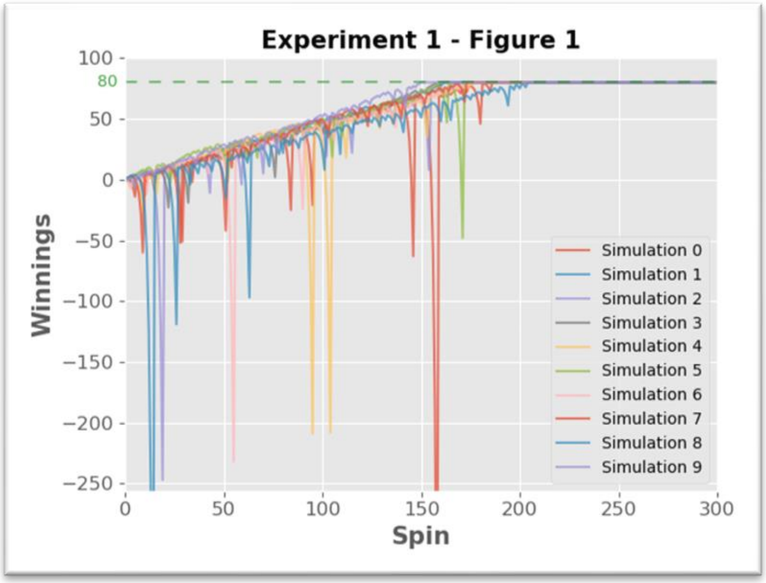

Figure 1—The winnings are graphed for 10 consecutive simulations each with a limit of 1000 spins.

1.2 Figure Two

Figure two, the Martingale simple gambling simulator is executed 1000 times. The mean and standard deviation as calculated from the results.

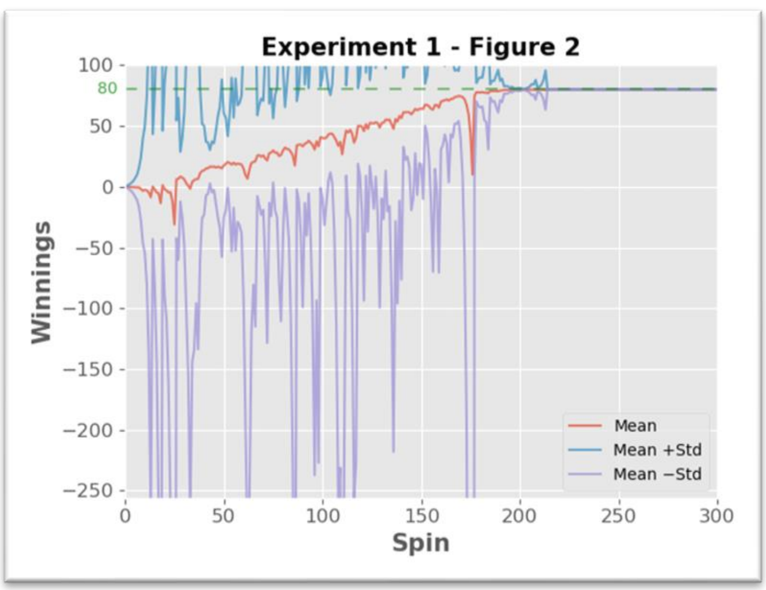

Figure 2—Mean with ± standard deviations lines graphed for 1000 simulations using the Martingale simple gambling simulator.

The mean line stabilized at the value 80 because this was the goal for the simulations. Once that goal was reached, the simulation ends and all simulations managed to reach that goal again in less than 250 rolls, due to having an unlimited bankroll.

1.3 Question 1

The estimated probability of winning $80 within 1000 sequential spins was approximately 100%. In my experiment all 1000 simulations managed to reach the goal of $80, this was aided by having an unlimited bankroll. The unlimited bankroll allowed the simulations to continually gamble until the goal was met.

1.4 Question 2

The Expected Value [2] for experiment one after running 1000 simulations would be approximately 80. My results for experiment showed that all 1000 of the simulations, $\left(\frac{1000}{1000}\right)$ reached the goal of 80.

$$Expected\ Value = \sum_{i=0}^{n} result * probability\ of\ result$$

$$\left(\frac{1000}{1000}\right) * 80 = 80$$

1.5 Question 3

The standard deviation of experiment one will converge and stabilize once a maximum value has been reached. The Law of Large Numbers [3], which states "the results obtained from a larger number of trials should be close to the expected value and will tend to become closer to the expected value as more trials are performed" (Wikipedia, 2020), coupled with the unlimited bankroll, allow for the standard deviation to behave in this manner. The value to which the standard deviations converge to is zero. After spin 210 almost every simulation has reached the goal of 80, thus a standard deviation of zero.

1.6 Figure Three

Figure three used the same data that was used to generate figure 2 but calculates the median, instead of the mean.

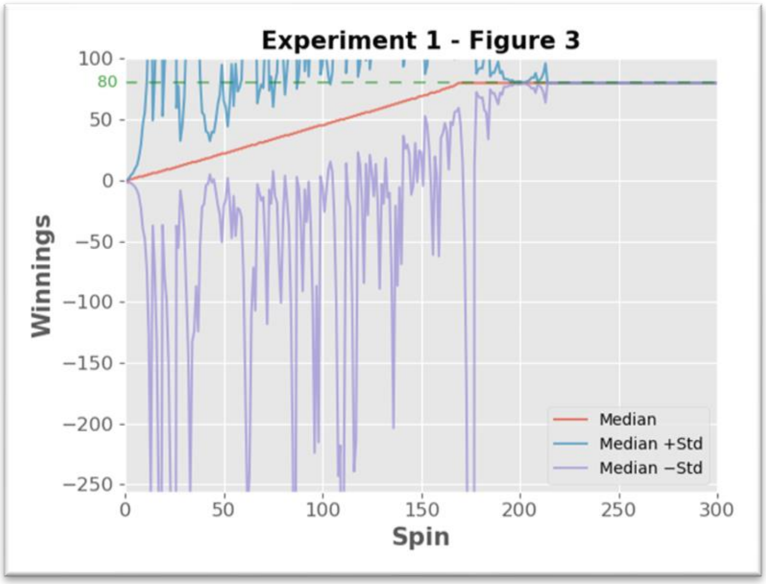

Figure 3 — Using the same data from figure 2, the median with ± standard deviations lines are graphed.

The median line shows a steady positive trend, ultimately stabilizing at 80. The reason is that when simulations reach the goal of $80, it will stop spinning and propagate their winnings forward for remaining spins. Ultimately stagnating at a value of 80, once more than half of all the simulations have reached their goal.

2 EXPERIMENT TWO – A MORE REALISTIC SIMULATION

Experiment two takes a more realistic approach. The same betting scheme as experiment one is active. When a spin is won, the bet amount is reset back to 1. When a spin is lost, the bet amount is doubled, and the simulation continues. What makes this simulator more realistic is having a limited bankroll, which is 256 in each simulation, once the bankroll runs out of money, the simulation is over.

2.1 Figure Four

Figure four shows the mean from running 1000 simulations of the more realistic simulator.

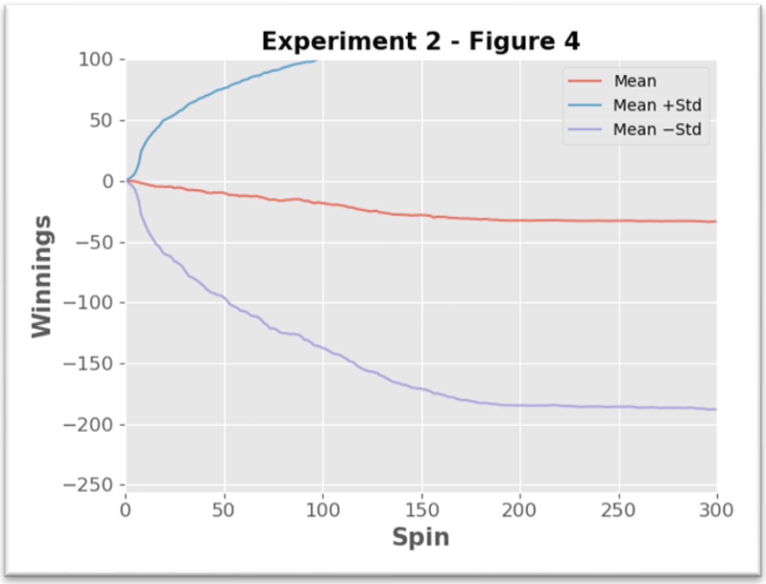

Figure 4 — Using the more realistic simulator chart the mean with ± standard deviations.

The mean of the winnings decreases as spins increase. The reason for this is due to multiple reasons. The first is having a limited bankroll and once depleted that value is filled forward through remaining spins. As more and more simulations run out of money, this will continue to reduce the mean of the winnings.

2.2 Question 4

The estimated probability of winning $80 within 1000 sequential spins is approximately 65.6%. Because out of the 1000 simulations 656 had reached 80.

$$\left(\frac{Number\ of\ times\ \$80\ is\ won}{Total\ Simulations}\right) = \left(\frac{656}{1000}\right) * 100 = 65.6\%$$

2.3 Question 5

The estimated Expected Value [4] for experiment two after running 1000 simulations was -35.25. I calculated this by getting the frequency of each value in my results. I ended up with -256 a total of 342 times, 80 a total of 656 times and one instance of -94 and one instance of -80. I used numpy to get an array of unique values and an array with their corresponding counts. I divided the counts array by 1000, the number of simulations. This results in an array of percentages. I take that array of percentages and multiply it by the array of unique values, and this gives me the expected value for each result. I then added up all the results to get -35.25. The expected value is also calculated manually below.

$$Expected\ Value = \sum_{i=0}^{n} result * probability\ of\ result$$

$$\left(\frac{342}{1000}\right) * -256 + \left(\frac{1}{1000}\right) * -94 + \left(\frac{1}{1000}\right) * -80 + \left(\frac{656}{1000}\right) * 80 = -35.25$$

2.4 Question 6

The standard deviation of experiment two, does in fact reach a maximum value and begin to stabilize. The lines for the two standard deviations, do not converge as the bets are increased. The reason that the lines stabilized and do not converge is due to the fact of ending a simulation once the goal of $80 has been met or once the bankroll has been depleted.

2.5 Figure Five

Figure five used the same data that was used to generate figure 4 but calculates the median, instead of the mean.

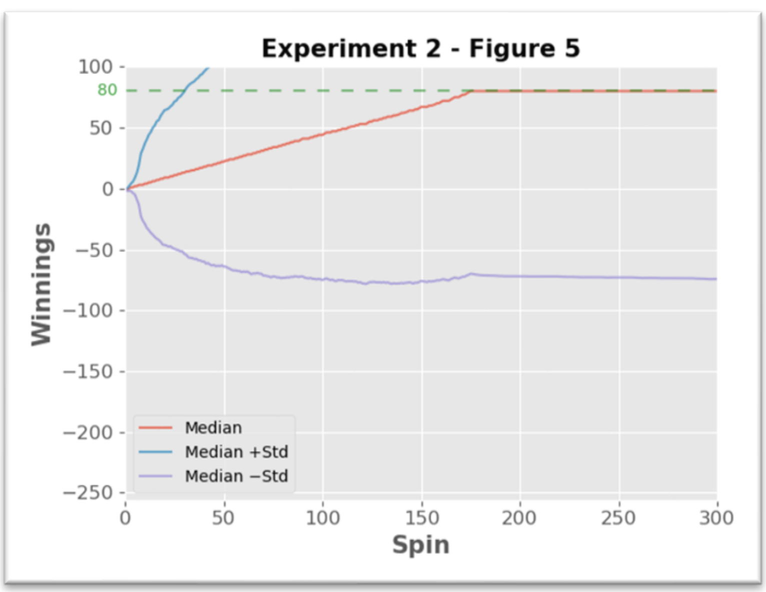

Figure 5 — Using the same data from figure 4, the median with ± standard deviations lines.

Figure 5 looks a little odd at first but when you think about how the simulations work, then the behavior is understood. At the beginning of the graph the median has a positive correlation with very large standard deviations. The reason is that early in the experiment there are more simulations which have a value greater than zero. The large standard deviations are due to randomness and the way the betting works. Having multiple losses in a row, quickly increases the bet amount, which in turn impacts the winnings more. Conversely having multiple wins consecutively, will cause winning to increase at a much slower rate. This is what drives the large standard deviations within the data.

3 REFERENCES

1. Roulette. (2020, August 06). Retrieved August 20, 2020, from https://en.wikipedia.org/wiki/Roulette
2. Expected value. (2020, August 13). Retrieved August 20, 2020, from https://en.wikipedia.org/wiki/Expected_value
3. Law of large numbers. (2020, August 15). Retrieved from https://en.wikipedia.org/wiki/Law_of_large_numbers