CS7646

Gianfranco Dutka
gdutka3@gatech.edu

***Abstract*—** The purpose of this paper is to investigate the Martingale betting system on the casino game of American roulette. The roulette wheel will be simulated with the assistance of the python programming language. A Monte Carlo method will be used to sample values from a roulette wheel and test the Martingale betting strategy. The betting algorithm will also be implemented with the assistance of the python programming language and supplementary libraries used for random number generation and for placing bets.

1 INTRODUCTION

American roulette is a version of roulette where the player is allowed to place a bet on a number, whether the color is red or black, whether the number is odd or even or whether it is high or low(Gillray). For this investigation we shall be creating a gambling simulator of American roulette and placing bets using the Martingale betting system. The system follows a simple algorithm where the player wagers a bet on either a black or red space in the roulette wheel. The algorithm is the following:

1. Wager a bet on either black or red slot in the roulette wheel.
2. If a bet is won, keep betting without changing the bet amount and set the bet amount to $1.00. Repeat step 2).
3. If a bet is lost, double the bet amount and bet and go back to step 2) until 1000 games have been played or until the cumulative winnings have reached $80.00.
4. If at any point you are playing with a bankroll and the bankroll runs out, you must stop playing.

The Martingale algorithm shall be implemented in python using the numpy library to keep track of cumulative winnings/losses per each hypothetical spin of the wheel. The reasoning behind using numpy lies in numpy's performance when allocating fixed memory arrays. Numpy also allows one to calculate statistics regarding winnings efficiently and quickly. The simulations will be run in epochs known as "episodes" where each episode will spin the simulated roulette wheel 1000 times and adjust the bet amount accordingly. The winning probability for the roulette wheel simulations was set to %47.4 since that is the probability of selecting either red or black space in a single spin of American roulette (Gillray). The odds are slightly less than %50 because American roulette uses two green spaces as opposed to other versions of roulette which only have one green space. Results will be plotted using matplotlib. Second simulation will use a bankroll of $256.00.

2 RESULTS

Experiment 1 consisted of running 10 episodes sequentially and collecting all the data. Afterwards the experiment was run again for 1000 episodes sequentially and likewise collecting all the data. Each episode spun the hypothetical wheel 1000 times. If at any moment the target winnings of $80.00 was reached or exceeded the cumulative winnings amount was used to forward fill the remaining iterations.

In the case of experiment 1 all 10 episodes managed to reach the target goal of $80.00 within the 1000 iterations. Manually inspecting the numpy arrays returned by the simulations showed that all 10 episodes reached the target goal by 210 hypothetical wheel spins. Given that the target was exceeded significantly below our maximum iterations threshold it is fair to say that given no limit to how many losses one can incur there is a %100 percent success rate in reaching the $80.00 target. Although that might seem promising it is worth pointing out that not taking into account potential losses is not a realistic scenario. It should also be mentioned that since we are doubling the bet amount every time that wager is lost, losses are dramatically increased. The longer the game is played therefore increases the odds of significantly incurring heavy losses.

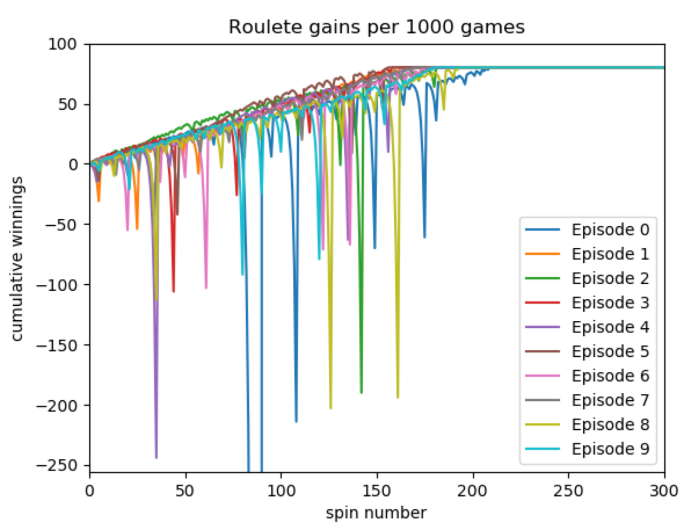

Figure 1—Roulette gains of 10 episodes. We see from figure 1 that all episodes reached the target $80.00 but every now and then significant losses were encountered.

Additionally, after running 1000 episodes and following the Martingale strategy, taking the mean of the cumulative winnings shows that the expected value of the winnings is 81. We know that in probability theory the central limit theorem proves that for independent and identically distributed random variables the sample mean will begin to approximate a normal distribution which is precisely what we see. The expected value was acquired by using numpy arrays and taking the mean of all the 1000 episodes(The mean of all the columns). Figure 2 provides a graphical representation of this with the mean plotted along with the standard deviation. Another thing we can see clearly from the graphs is that every now and then the mean cumulative earnings drops drastically. The reason for this was mentioned earlier but has to do with the betting amount doubling on every loss. If at any point several losses in a row end up taking place, that causes the cumulative earnings to drop drastically with every doubling until a winning bet is finally achieved. Fortunately for this experiment, since there is no bankroll and experiment 1 cannot run out of money the target is reached with every single run.

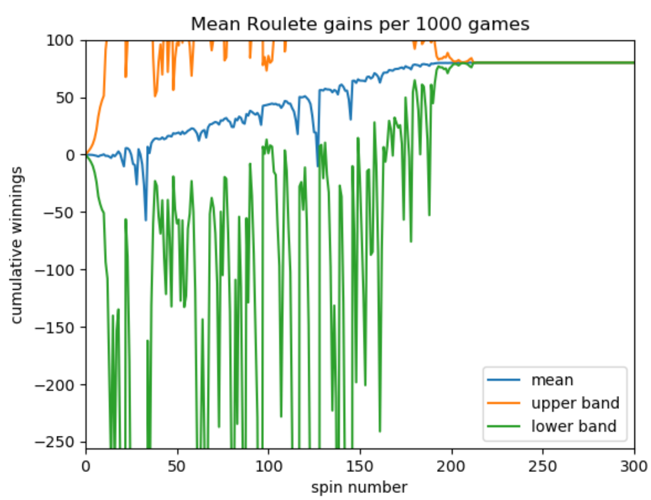

Figure 2*— Mean Roulette gains after 1000 episodes*. Mean cumulative gains seem to converge around the probability density function. Upper and lower bands above and below two standard deviations around the mean also converge to the mean. The doubling of bets after every loss means that consecutive losses significantly hurt cumulative earnings until a win is finally achieved.

What's more, the upper and lower standard deviation bands seem to reach a maximum and minimum limit respectively. After that point the upper and lower bands seem to converge to the mean centered around the middle of the probability density-function which in this case is 80. The maximum value of the upper band is 4132 and the minimum value of the lower band is

-4201. The reason for such a large standard deviation also has to do with the fact that the bet amount is doubled on every loss. This increases volatility in our data significantly until convergence happens once the target of $80.00 is reached. Once the target is reached since betting is brought down to zero and data is forward filled, all the variance in the earnings drops and the upper and lower bands converge to $80.00.

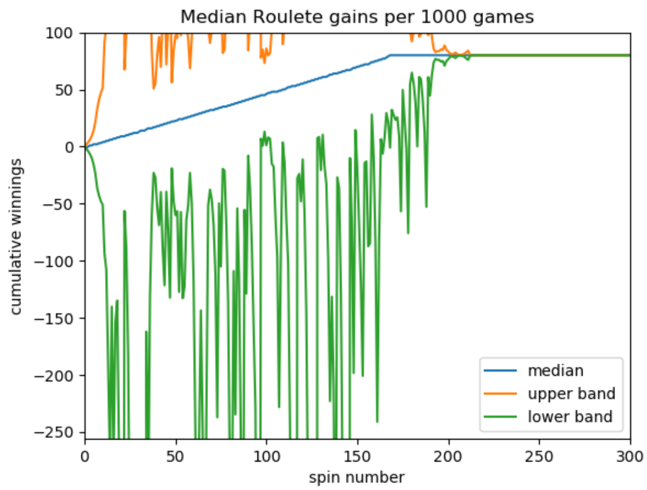

Figure 3— *Median Roulette gains after 1000 episodes*. Median cumulative gains seem to converge around the probability density function. The blue line which plots the median doesn't have the drastic drop in earnings seen with the mean. The reason is that the median sorts the earnings and picks the middle. Due to weighing the losses differently this drastic loss is not plotted. Despite this the earnings target is reached as well in the case of the median.

For experiment 2 after 1000 episodes have been simulated more than half of the cumulative earnings are above $80.00. To be exact, 634 of the episodes managed to reach a cumulative earnings that is above $80.00. On the other hand 366 of the episodes run with the bankroll ended up with 0 earnings. Given this ratio we can give an estimate that the probability of winning $80 given our strategy is 0.634 or %63.4.

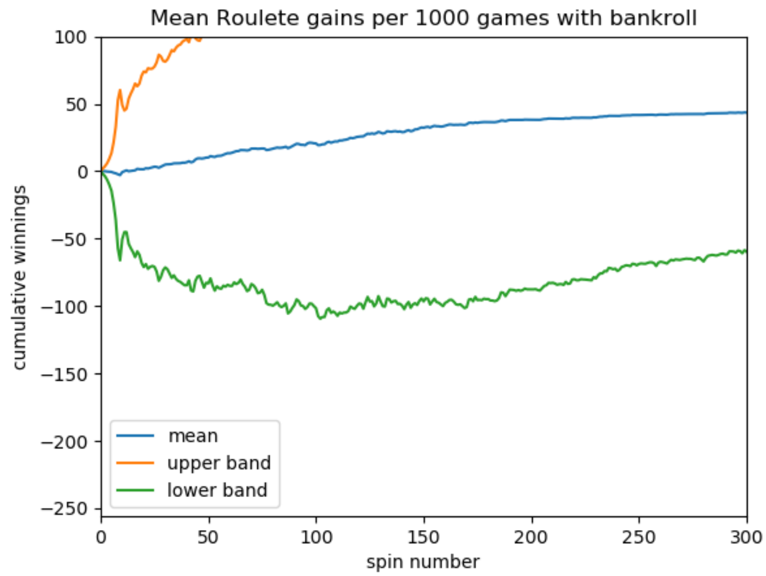

Figure 4— *Mean Roulette gains after 1000 episodes with bankroll*. Mean cumulative gains seem to converge around the probability density function. Because we have a bankroll which can hit zero, this effectively minimizes the drastic drop in earnings seen in experiment 1. In addition, since many of the episodes end up being zero due to running out of money the expected value drops to about $50.00.

The expected earnings for experiment two ended up being 50.64 despite having a limited bankroll. The reason for this is that after the 1000 iterations most of the episodes resulted in hitting the target 634 times out of 1000. This results in an expectation of 50.64 as mentioned previously. Since the rest of the episodes resulted in zeros this brings down our cumulative earnings when compared to the experiment which does not use a bankroll. In summary, having the limitation of a bankroll causes many of the episodes to zero-out therefore dropping our expected value. It might also be worth pointing out that the median value of the cumulative earnings did manage to hit the target earnings goal. The reason for this is that more than %50 of our simulated episodes did manage to reach the target.

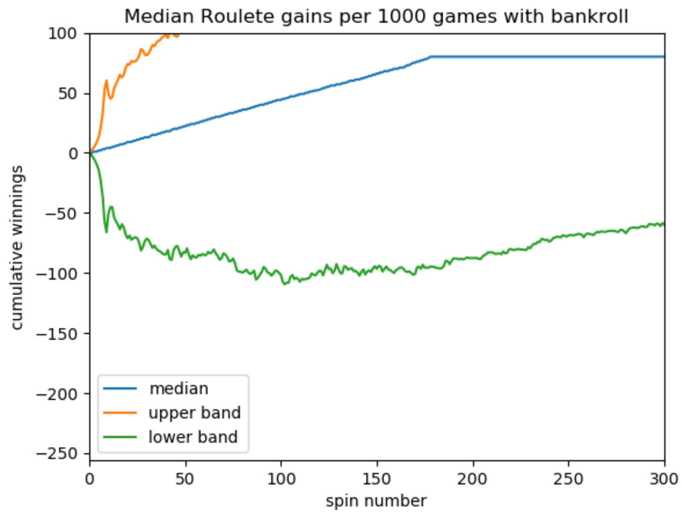

Figure 5*— Median Roulette gains after 1000 episodes with bankroll*. Median cumulative gains seem to converge around our target goal. Because the median doesn't take the average of our gains and/or losses the plot shows that after 1000 episodes more than %50 of our episodes reached the goal of $80.00.

What's more, during experiment 2 the upper maximum value is 170 and the minimum value for the lower band is -109. The upper band stabilizes around 127 and the lower band stabilizes around -26. This happens once the expected value stabilizes at 50.64. This makes sense since the upper and lower bands are 2 times the standard and lower deviations and the standard begins to stabilize at around 38.55. Unlike experiment 1 we also see that the upper and lower bands never converge to the mean. The reason for this is that in experiment 1 all of the episodes resulted in the same cumulative earnings so there was no variance in the final result. This is not the case once we add in the bankroll in experiment two. Having the bankroll in experiment two means that a large amount of the final earnings in some of the episodes will be zero thus causing the standard deviation never to hit the value it did in experiment 1.

2 CONCLUSION:

From running experiments one and two we can see that the benefit of using the expected value from 1000 episodes lets us come to a conclusion about the likelihood of reaching our target goal of $80.00. Running a single episode would not give you much information about the actual likelihood of hitting the target goal of $80.00. The reason for this is that it is more than likely that if you were to run just a single experiment that single experiment wouldn't be representative of the actual likelihood of achieving your target. The only way to figure this out experimentally is to calculate the expectation of many simulations/experiments to get an accurate likelihood.

5 REFERENCES

1. Gillray, James. “Roulette.” *Wikipedia*, https://en.wikipedia.org/wiki/Roulette#. Accessed 1 September 2023.