Elias Yishak

ISYE-6644 Project Report – Blackjack Simulation

**Abstract**

Blackjack is a popular casino card game that is played all over the world. The game involves a player and a dealer, with the objective being to reach a total card value of 21 or as close to it as possible without exceeding it. Over the last few years, computer generated simulations of the game have become increasingly popular, allowing interested parties to test methods and strategies to study the game's mechanics in a controlled environment. Monte Carlo simulations are a popular method for simulating blackjack games, as they allow for a statistical analysis of outcomes over many hands as opposed to running several trials by hand. By simulating blackjack games using a Monte Carlo approach, we can gain insights into the expected outcomes of various strategies and betting systems, as well as study the probabilities of certain events occurring during gameplay. For this report, we will simulate 1 million games for seven different players, each with their own level of risk aversion. The results show that the riskiest and least riskiest players don’t win the most on average. Instead, there is some kind of optimization that can be implemented in a player’s strategy that will maximize how much that player can win. This report also analyzes every possible combination of the hands that are initially dealt (2 cards) against every possible face up dealer card. The insights from this report can be used to generate strategies for a game of blackjack by referencing outputs of this analysis. Additional future work is also included that have not been accounted for in the simulation.

**Introduction**

Blackjack is a popular casino card game that is played all over the world. The game involves a player and a dealer, with the objective being to reach a total card value of 21 or as close to it as possible without exceeding it. Over the last few years, computer generated simulations of the game have become increasingly popular, allowing interested parties to test methods and strategies to study the game's mechanics in a controlled environment. Monte Carlo simulations are a popular method for simulating blackjack games, as they allow for a statistical analysis of outcomes over many hands as opposed to running several trials by hand. By simulating blackjack games using a Monte Carlo approach, we can gain insights into the expected outcomes of various strategies and betting systems, as well as study the probabilities of certain events occurring during gameplay. In this report, we will explore the methodology and results of simulating blackjack in a Monte Carlo style, and discuss the implications of these findings for players and researchers alike.

For this project, I decided to use the popular computer language, Python to build the simulation from scratch. This enabled me to control the environment in ways that I may not have been able to do with other simulation programs. Below, I will explain how the code was setup so that I can tweak the simulation as needed. Additionally, this project looks to see how different risk levels can help in blackjack when compared to the dealer’s face up card. For example, a given player may play conservatively by never hitting when they have a hand value of 16 or greater, while someone that is much more risk averse will hit on anything up to 18 for their hand value. Furthermore, I will also be analyzing the pairs of cards a player is dealt compared to the dealer’s face up card and trying to identify hands that yield higher wins. Historically, the accepted chance of winning a game of blackjack is around 42% so we will see if we can reproduce this in our simulation.

**Methodology/Implementation**

To start building out this simulator for blackjack, I have decided to build it all entirely in Python since it will be much easier for me to control the entire simulation as opposed to having some abstractions put in place through simulation software packages, like Arena. Being able to organize all the entities in a blackjack game into Python classes also makes it easier to program each role. The following entities (classes) will be created:

* **Card**
  + This class served as a data class that would hold information about the rank and suit
  + It would also handle aces by allowing them to have two values, 1 and 11, which is based on the hand the player has dealt
* **Dealer**
  + A special subclass of **Player** since they will also have a hand
  + Really only responsible for giving out the cards or ends the game if they have blackjack
* **Deck**
  + Handles different shuffling styles
  + Responsible for implementing a reshuffle, activated by **Dealer**
  + Returns the next card, activated by **Dealer**
* **Hand**
  + Will be responsible for holding all the **cards** that have been dealt to a player
  + Also has methods available that will add a new card or get the value of the hand
    - For this simulation, we always returned the soft value (i.e. an ace and 5 will always be 16, but an ace, 5, and 8 will return 14)
* **Player**
  + Will have a parameter that sets how risk-averse each player will
    - Between risky or safe
  + Will have an attribute that holds the current **hand**

The classes described above will provide all the functionality to run several games in python. The simulation will begin by determining how many games will be simulated and by setting a random seed (optional) for repeatability of the experiment. There are also seven players that have been initialized, each with their own level of risk denoted by the value they will stop taking hits from the dealer. The player that has the least amount of risk is going to be player “A” who will stop hitting after reaching 15 or greater. The most risk averse player will be “G” who will continue to play until they hit 21 or greater. The same can be observed for player “B” who will stop hitting after getting a value of 16 or higher. Each of the seven players will increment the number they will stop hitting by 1, from 15 to 21, and “A” to “G”.

**Findings/Results**

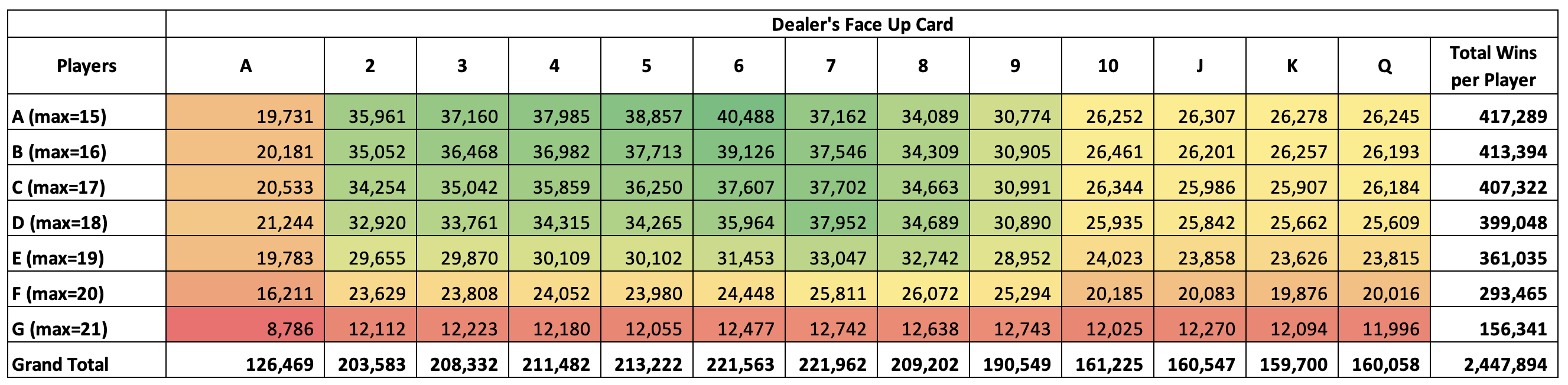
****

Table 1: Player profiles vs dealer's face-up card

After conducting 1 million simulations, the results are shown above in the table. As discussed in the above, we created seven different player profiles, each with their own level of risk denoted by the max value in the first column. Player “A” will continue to hit if their hand value is less than 15, once it is 15 or greater, they will stop. The top row also has the dealer’s face up card that is dealt. The data within the table is an aggregate count of wins for each player and dealer face up combination. For example, with player “A”’s risk aversion and the dealer having a 10 face up, we counted 26,252 wins. It is important to note that this is just an aggregate count of wins, it does not look at how many loses each player had for a given dealer’s hand. The last column also includes the total number of wins per player for hands that were dealt by the dealer. A color scale has also been added on top of the data to easily find patterns in the number of wins.

When analyzing the pattern created by the color scale, the most striking finding is that there seems to be a hot spot towards the center of the chart that shows many more wins. We also observe that the player with the most wins was player “A” when the dealer has a 6 face up. However, because this is just an aggregate count of wins, we cannot conclude that having a low risk aversion will lead to more wins on average, just total. This can be affected by how the cards have been dealt in the random order, although the randomness should be fairly uniform for 1 million games. But even with the uniform assumption for large numbers of games, we see a pattern where the hotspot begins to fizzle out for larger dealer values which indicate that if the dealer has a mid-value card, the player is has a higher chance of winning. Next, we will look at one player’s simulated games and analyze how the different pairs of cards yield wins when compared to the dealer’s face up card. However, due to the size of the table, it will be included on the next page to allow readers to analyze the data in its entirety. Additionally, only player “C” will be analyzed in this report, but reviewers of this report can check the accompanying zip file to find the pivot table for each player.

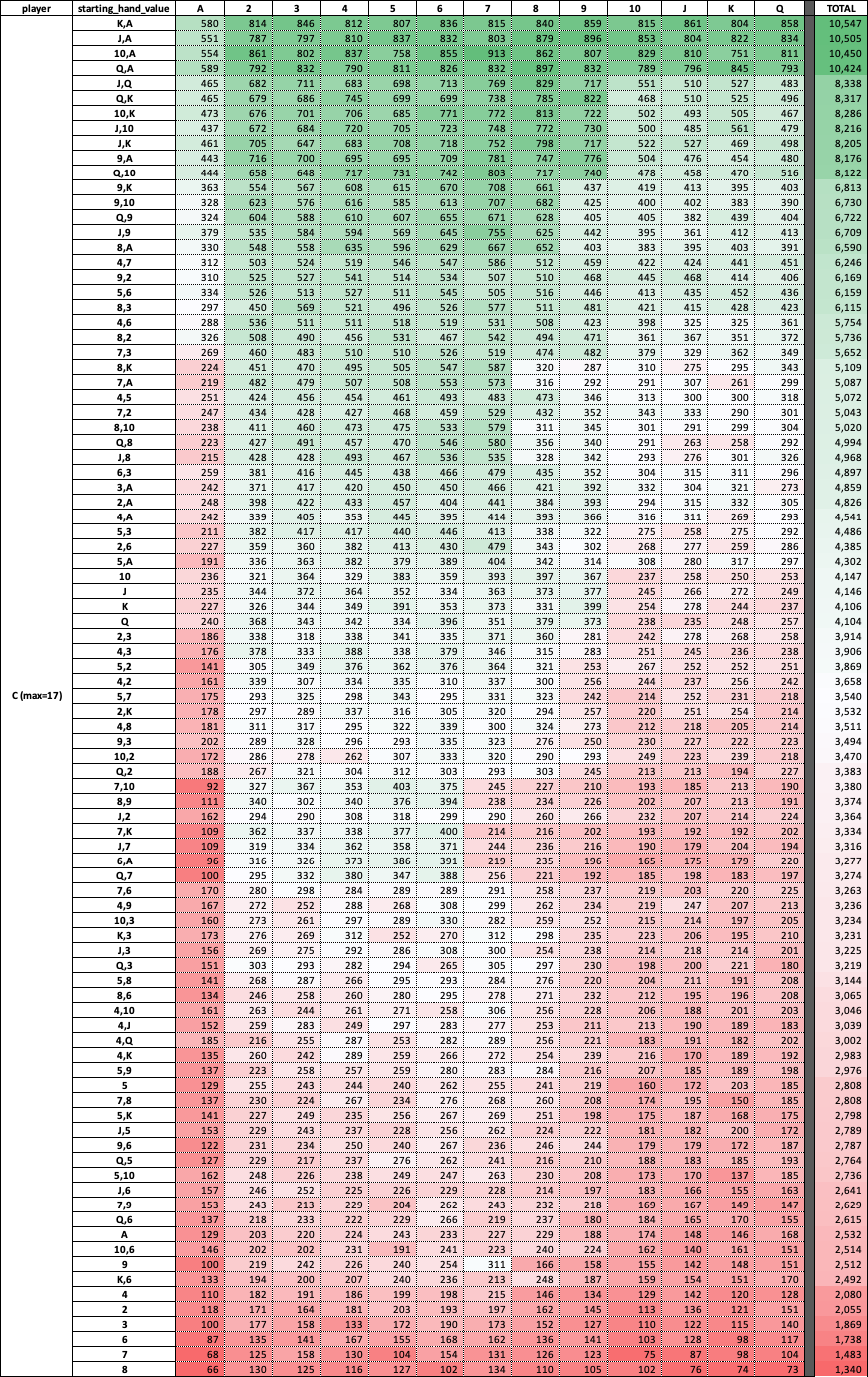


Table 2: Each hand for player "C" vs dealer's hand

The table above shows the specific hands for player “C” in the second column. Each cell in the column is a unique set so if the cards ace, king will be the same as king, ace since they are identical in the game of blackjack. Additionally, if there is only one character in the column, such as “J”, then that means the player was dealt two jacks for that game. This covers every combination of cards that a player can be dealt on the first hand. The top row, similar to table 1, also has the dealer’s face up card for each column and the data in each cell is an aggregate count of wins for that given hand. A color scale, similar to table 1 has also been added to help identify patterns, however in this instance, there are two scales being used. One is applied on the middle of the table for each cell, and the second is on the total column on the right.

This view of simulation shows much more information than table 1 about specific strategies a player can take. For example, we can observe that the win counts are significantly lower when the dealer’s face up card is an ace. Additionally, as we move to the right, or increasing the rank value of the dealer’s face up card, we can observe that the wins increase, and then decreases again, highlighting the hotspot phenomena we observed in table 1. Additionally, we can observe that the count of wins is significantly less when receiving the same card twice when the rank is low, as it can be seen when looking at the last 6 rows in the table.

**Conclusions**

In conclusion, the Monte Carlo simulation of blackjack games provides a powerful tool for studying the game's mechanics and evaluating various strategies and betting systems. Through this simulation, we have been able to examine the probabilities of certain events occurring during gameplay, such as the likelihood of busting or reaching a natural blackjack. We have also gained insights into the expected outcomes of various betting strategies, such as flat betting or progressive betting. Overall, the results of this simulation can be applied to real-world blackjack games, as players can use this knowledge to make more informed decisions during gameplay. Moreover, this study contributes to the broader field of game theory and probability, demonstrating the usefulness of Monte Carlo simulations for analyzing complex systems. As technology continues to advance, simulations like these will likely become even more sophisticated, offering researchers and enthusiasts new avenues for studying and enjoying games like blackjack.

Future work:

* Analyze how many games you lose instead of win based on risk and dealer’s face up card?
* How does the hands of players around you affect your chances of winning?
* Does splitting your hand when you get doubles increase your chances of winning?
* Can we include a “bank of money” for each player to see how successes in between games builds on the money bet?
  + Does varying the amount bet on each game help you or hurt you over hundreds of games?