Capstone Project: Blackjack Simulation

Jiani Chen
Partner: Rachel Redman
Physics 88, Y. Kolomensky

May 17th, 2019

Abstract

This project makes use of card counting to simulate a game of Blackjack. The user can see how a game of Blackjack is played out and use this for further analysis. By implementing card counting into the game, we can analyze the benefits of making risky plays. The simulations showed that being the last player of the game has a slight advantage compared to being the first player. Making riskier plays over the other players also shows a slight advantage. However, further analysis is necessary to make concrete conclusions.

Contents

1 Introduction

Blackjack, once known as *Vingt-un* or *Vingt-et-un*, means 21 and was a game that became popular across Europe in the 17^{th} century. This game is popular for making use of card counting techniques that provides the player with improved chances of winning it big in casinos. However, being able to successfully card count in a live game is no simple task. This project looks into the statistics of Blackjack by implementing the advantages of card counting.

This project is a python script that simulates a game of Blackjack between 2-5 players. In a standard game of Blackjack, each player starts with two cards. The player decides whether to draw another card (or *hit*) depending on whether the sum of the values of the hand adds up to 21 (or *Blackjack*). The goal is to have the hand sum up to as close to 21 as possible but without going 21 (which is a *bust*). The simulation shuffles a standard deck of cards and each player is controlled by the computer AI. The computer AI uses a card counting system that allows it to calculate the probability of reaching Blackjack each round and uses that probability to decide whether it's safe to take a hit or not.

This simulation allows the user to analyze the outcomes of a game of Blackjack with a desired number of players. The user can see the printed results of one game of Blackjack or disable the printed text, allowing the user to analyze the results of multiple games without effecting efficiency. For each game, the user can change the number of players, number of rounds per game, and the risk factor of taking a hit. The risk factor is a value from 0-1, where it represents the minimum probability for the next hit taken to sum up to 21. Thus a lower risk factor indicates the riskier the play.

2 Implementation

We've designed the code by breaking down all the steps to playing Blackjack into individual functions. This includes the game setup, turn process, and results of the game. After the actual simulation is complete, we've also implemented statistical analysis for the simulations.

The first step is setting up a game. For a game to be played, we must have a deck of cards and to provide a hand for each player. The deck contains 2 item lists of the 52 playing cards where the first item is the name of the card and the second item is the value it represents in Blackjack. When a card is handed to a player, the card is also removed from the deck to reflect that change. This setup process has also been adapted so that if the game were to run out of cards, the deck can be reshuffled and played again.

The next implementation of the code is the turn process. During the turn process, a player can choose to take a hit or stand depending on the sum of their hand. Because the special card, Ace, can be a value of 1 or 11, the *hand_values()* function accounts for all possible summations, given the maximum number of Aces in a game is 4, as shown below

```
def hand_values(hand):
    """Accounts for all possible combinations of Aces in a hand."""
    s = [0]
    for i in hand:
        s[0] += i[1]
    aces_count = 0
    for i in hand:
        if i[1] == 1:
            aces_count += 1
    if aces_count >= 1:
        s.append(s[0]+10)
        if aces_count >= 2:
            s.append(s[0]+20)
            if aces_count >= 3:
                s.append(s[0]+30)
                if aces_count == 4:
                    s.append(s[0]+40)
    return s
```

For the computer AI to decide to take a hit or not, we've implemented a card counting system based on the known and unknown cards to the player. In a standard game of Blackjack, the player can only see one revealed card from the other players hands. This factor changes what is unknown to the player and each player is able to keep track of all unknown cards. To do this calculation, the following portion of the code shows how a hit is decided.

```
def probability(sum_, unknown, prob):
    count = 0
    for value in unknown:
        if value + sum_ <= 21:
            count += 1
    if count/len(unknown) > prob:
        return True
    else:
        return False
```

This function calculates the whether the new sum will be less than 21 and calculates the probability based on all of the unknown cards. Where the probability is given by

$$probability = \frac{count}{lengthofdeck} \tag{1}$$

If this is greater than a given risk factor, *prob*, then a hit will be taken. We will discuss more about the risk factor later in the Analysis portion.

The final portion of the code involves tying all of these functions together to generate the results of a simulation. To keep track of winners of each game, there is a points system where a player that wins a round receives 1 point or winning with Blackjack receives 2 points. If there are ties, the points are split between each round winner. The final *game()* function keeps track of each players points and game winners after each game. The following shows how an output of the simulation will look like for a game of 2 players and only one round for simplicity.

```
This is a Blackjack simulator. If you wish to read the rules, enter "Y".
If not enter any other character:  n
Input the number of players (2-5): 2
Now enter the risk factor for each player. (Value between 0 to 1)
Risk factor for player 0: .5
Risk factor for player 1: .5
How many rounds in the game? 1

The simulation will now begin...

Results for round 1
Player 0 : stand
Player 1 : stand
Player 0 wins with a hand value of 19
Current points: [1.0, 0.0]
The winner(s): 0 with 1.0 point
```

Finally, there is an option to disable the printed text as shown above to provide quicker statistical analysis of running multiple trials of the simulation.

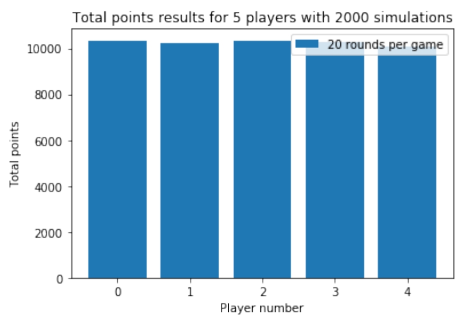
3 Analysis

In our simulations, there are two factors that can affect the results of the game, the order of the players and the risk factor. In a Casino game of Blackjack, the dealer always goes last because this is supposed to be a slight advantage for the dealer. We can put this to the test in our simulation by counting the number of wins and total points of each player over many iterations of the game. To perform these tests, 2000 simulations of 5 player games of 20 rounds each game is played out. Figure 1a shows a histogram of total points of each player over 1000 simulations and Figure 1b shows the total games won.

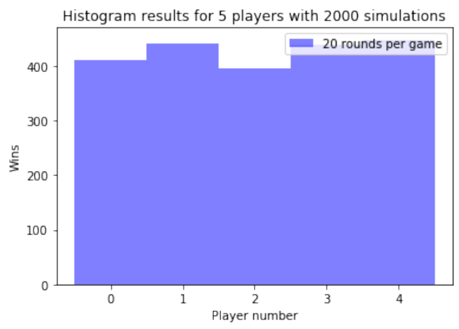
We notice that despite there appearing to be a disadvantage for the player who goes first, the points distribution is pretty even between all players. Another test we can perform is finding the difference in points compared to the average points of the other players. Figure 2 shows the distribution of these results with the same set-up for the player that goes first (2a) and the player that goes last (2b).

Keeping in mind that the order of the player has little impact on the results of the game, we can test the risk factor that decides how each player takes a hit. Recall the the risk factor represents the minimum probability of the sum not going over 21 when taking a new card. Figure 3 and 4 shows the difference in points compared to the average of all the other players for the first player and the fifth player respectively. Notice that the distribution shows the greater deficit in points that the first player suffers compared to the player who goes last. This shows that in a Casino game of Blackjack, the dealer who usually goes last does indeed have an advantage.

We can also change up the risk factor to see if making riskier plays is beneficial to the player. Figure 3 shows the results of increasing the risk by changing the risk factor to 0.45. Notice that by making a riskier play, both players are able shift their points difference distribution to the right, increasing their odds of winning. Comparing this to a less risky play (Figure 4), there also appears to be advantages there compared to taking the middle ground of risks.



(a) Total points of each player over 2000 simulations.



(b) Total wins of each player over 2000 simulations.

Figure 1: First run of simulations.

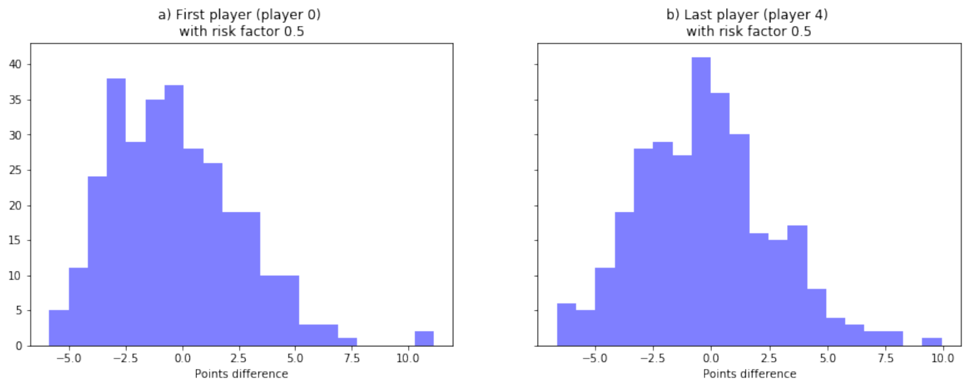


Figure 2: Differences of points distribution

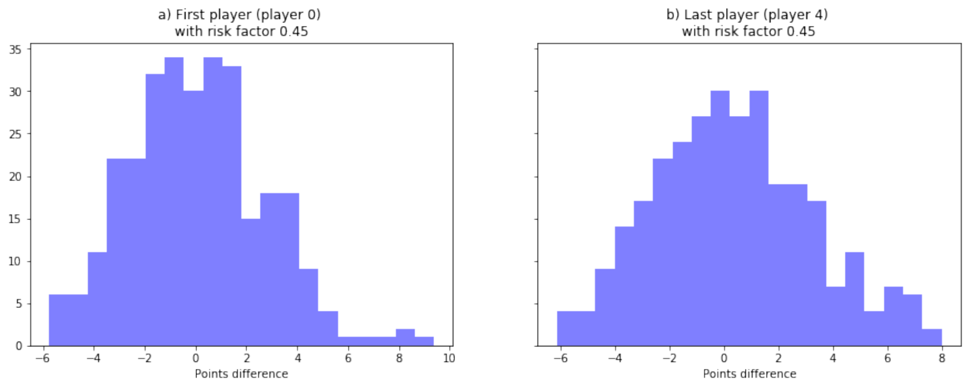


Figure 3: Differences of points distribution with risk factor 0.45.

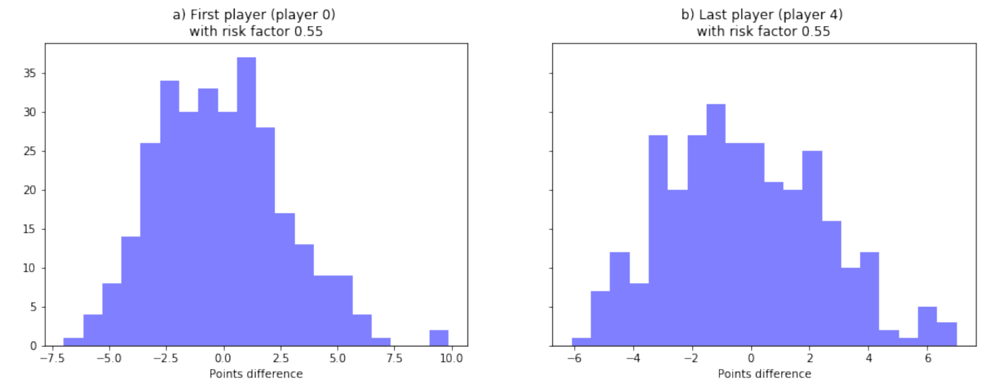


Figure 4: Differences of points distribution with changed risk factor 0.55.

4 Conclusion

This project aims to simulate games of Blackjack with the use of card counting and use the simulations to analyze the statistics of the game. We've found that the dealer or the player who goes last usually has an advantage over all the other players. Taking risks with card counting is also another big deciding factor of increasing your odds of winning. Despite having covered many fundamental aspects of Blackjack, there can still be many improvements to be made. In particular, we can develop a more realistic Blackjack simulation or implement a more sophisticated way to count cards. This project will be useful for any future analysis of find the perfect risk taking methods and card counting techniques that is beneficial to winning Blackjack.