Registration number 100280305

2022

An Investigation and Simulation into Card Counting Methods in Blackjack

Supervised by Elena Kulinskaya

University of East Anglia
Faculty of Science
School of Computing Sciences

Abstract

This report is an investigation into the game of blackjack,its rules of play, playing strategies and methods of card counting. The main goal is to discover whether blackjack can be a profitable game, and if so, to determine the best playing strategies and the best card-counting methods. These methods and strategies have been implemented into simulations. The present report includes an observation of how the casino rules work for or against the players. New card counting methods have been developed in an improvement on existing methods

Acknowledgements

I would like to thank Dr Elena Kulinskaya, from the University of East Anglia, for their guidance.

Contents

List of Figures

List of Tables

1. Introduction

Blackjack is an immensely popular card game that has been played for many years. The exact origin of the game is not agreed upon, but it is believed to date from the early 18^{th} century. (Crescent Schools, 2019) For the average player at a casino, it can be seen as a simple game that relies purely on luck. However, over the years people have made a vast amount of profit by using advanced playing strategies. The method that allows players to gain an advantage over the casino, known as card counting, first captured most people's attention in 1966 after Edward O. Thorp released his book "Beat the Dealer" (Thorp, 1966). Since then, numerous counting methods have been developed and improved. Blackjack is the only game in which the player can make a profit at the expense of a casino, which makes it a particularly interesting topic to investigate. This study aims to find out the true profit-making potential of blackjack by exploring numerous strategies and game-playing scenarios through a large number of simulations. However, as we would expect, casinos are very much alert to players using card counting to try and win against them. To try and prevent this, they have found strategies of their own to combat such players' edge. They achieve this by introducing specific rules that influence how the game is played. For example, some casinos will only use a larger number of decks, or pay out less money for havinbg a blackjack. These make it particularly hard for the player to gain an edge over the casino. I will be looking into the tactics that the casinos use, in order to find the best playing conditions for the player.

2. Blackjack

2.1. Aim of the Game

The basic aim of the game is simple: to get a hand that is closer in value to 21 than that of the dealer's, but without going over 21. If the player gets exactly 21, this is known as "blackjack". You as the player have control over all your decisions, such as hitting, splitting, doubling or standing. The dealer, on the other hand, must follow the house rules, and is able only to hit or stand. The game is usually played at a table of 2-7 players, and may use 1 to 8 decks of 52 cards each. To start the game, each player and the dealer are dealt two cards. The players can see only one of the dealer's cards. Numbered cards retain their number value, face cards all have a value of 10 points, and the Ace can be used as either 1 or 11. Each player at the table is playing directly against the dealer and should not be interested in the actions of the other players, but only in the cards that they are being dealt. Bets are taken at the start of each round and can only be altered by either doubling or splitting. If the player's cards have a higher total than those of the dealer (while, as noted, being no higher than 21), the player receives their bet back, as well as the same amount from the dealer; otherwise the dealer takes the player's stake. For example, if the player places a bet of £10 and wins, they receive their £10 back and an additional £10. If the player's and the dealer's hands have the same value, then the player simply receives their bet back. The dealer will take their turn after all the players have completed their turn. If the dealer's current hand value is less than 17, they will either hit until they reach or exceed this value. There are two common rules that casinos use, S17 and H17. The S17 rule means the dealer must stay if they have a soft 17, and H17 means they must hit until they get a hard 17 or higher. These rules and their importance are explained in more detail later. If the dealer ends up going over 21, then all players with hands less than 21 will win. In the case where both the player and the dealer go over 21, the player loses their bet.

2.2. Terminology

- **Bankroll** - The money that you have to play with. (Only the portion you are willing to bet at the casino)
- **Basic Strategy** - A collection of actions that offers the best playing conditions to

minimise your loss and enhance your wins.

- **Bet Spread** - The spread of your betting amounts. In a 1-15 bet spread and bet unit of 100, your minimum bet will be 100 and the maximum 1500.
- **Blackjack** - The perfect hand to receive; you need an Ace and a ten, making a score of exactly 21. The casino will typically pay in a 3:2 ratio for this.
- **Bust** - When you receive a total score of over 21.
- **Card Value** - Numeric value of the card, between 1 and 11.
- **Casino Advantage/Edge** - The average profit the casino is expected to make from the players. This is explained in more detail in section 2.3.
- **Double** - Having received 2 cards, the player can double their bet before the dealer deals another card. The second bet is equal to the first bet. The dealer will then deal only 1 card. The dealer settles all bets at the end of the hand.
- **Hard Total** - a hand without an Ace; or: a hand with an Ace, of which the value must be 1, otherwise the hand would exceed 21.
- **Hit** - Where a player asks the dealer to deal another card from the deck.
- **Insurance** - If the dealer has an ace, the player may choose to place a bet of up to half the value of the original bet. If the dealer has blackjack then the player does not lose any money.
- **Penetration** - A number or fraction that shows how many cards or decks will be dealt out before shuffling. This may be a percentage or a fraction.
- **Push** - A tie, where both the player's and the dealer's hands have the same value, or both bust.
- **Shoe** - The device to hold the playing decks.
- **Soft Total** - When the hand contains an ace which can be counted as either 11 or 1. For example, an ace and a 5.

- **Split** - If the player holds two cards of the same value in their hand, for example two eights or two sixes, the player can split the pair, and play each card as if they were two separate hands. Following such a split, the player places their original bet on one hand, and an equal bet on the second hand. The player first plays the hand on the right in the same way as a normal hand, i.e. hit, double, stand, etc., after which the player plays the hand on the left.
- **Stand** - The player decides not to take an extra card.
- **Surrender** - The player decides to stop playing, and receives half of their bet back from the dealer.
- **Up-card** - The one dealer's card that is revealed to the players.

2.3. Casino Advantage

The "Casino advantage" is a term that describes the mathematical advantage that the casino holds over its players, for any given type of game, considered over time. This applies to all games played in a casino, and the edge they have over the player varies (Hannum, 2003). In the game of blackjack, the typical advantage over an inexperienced player is 2%. In simple terms, it can be stated that for every £100 placed as bets, a player will lose £2 (Simon Young, 2020). There are, of course, many different possible scenarios, and a player could lose an entire £100 bankroll after 200 hands, for example.

House Advantages for Popular Casino Games	
Game	*House Advantage*
Roulette (double-zero)	5.3%
Craps (pass/come)	1.4%
Craps (pass/come with double odds)	0.6%
Blackjack – average player	2.0%
Blackjack – 6 decks, basic strategy*	0.5%
Blackjack – single deck, basic strategy*	0.0%
Baccarat (no tie bets)	1.2%
Caribbean Stud*	5.2%
Let It Ride*	3.5%
Three Card Poker*	3.4%
Pai Gow Poker (ante/play)*	2.5%
Slots	5% - 10%
Video Poker*	0.5% - 3%
Keno (average)	27.0%
**optimal strategy*	

Figure 1: A summary of the house advantages for many of the popular casino games (Lucas and Spilde, 2019)

Figure 1 shows that different games provide the casino with different advantages over their players. We also notice that for Blackjack we have 3 different scenarios of play. If we begin to use basic strategy with 6 decks, we can bring the advantage down to as low as 0.5%, and as low as 0% if the game is played with a single deck. This shows us that already with two simple adjustments, in the long run, we would be on level playing terms with the casino. To push this a little bit further and tip the advantage in favour of the player, we need to start counting cards.

The casino will commonly introduce several variations to their rules to increase their advantage over the player (Vidámi et al., 2020). An increasingly common tool that some casinos use is a continuous shuffle machine that will shuffle all the cards after each round. This essentially makes card counting pointless, so we will not be considering this in our simulations. Another alteration casinos use that is very similar is the number of decks used to play blackjack. This combats card counting as the higher number of cards makes it harder to keep track and have a clear indication of when the deck is better for the player. Of course, this has an effect even when not card counting, because as the player is dealt a 10-point card the probability of the next card being 10 or an ace is greatly reduced, which in turn means there is a lower chance for the casino to pay out 3:2 for a blackjack. Other rule alterations are intended to limit when players can double or split, not allowing players to re-split aces, paying out 6:5 for a blackjack and many others. (Kendall and Smith, 2003)

2.4. Rules

The rules used for this investigation will be grouped into two types: those that will remain unchanged throughout the simulations (Table 1); and those that will be varied in order to find the effect that they have on the results (Table 2).

Rule	**Justification**
Double after split	Most casinos will offer double after split
Double on any 2 cards	The Most practical and popular option is to double on any pair
Re-split to 2 hands	Simplest to implement and no major difference in odds between 2 and 3 (0.08%) for example. (Michael Shackleford, 2015)
Can Hit Split Aces	Most common variation & allows for more hands to be played
No Surrender	Small impact & not agreed when exactly to surrender
Blackjack Pays 3:2	It is clear that 6:5 will have a highly negative effect on the player's profit

Table 1: Rules remaining **unchanged** during simulations, and justification for doing so.

Rule	Variations	Justification
Strategy	Basic, Never Bust, Mimic Dealer	To showcase the effectiveness of basic strategy
No. of decks	1, 2, 6, 8 Decks	Strongly influences the house edge
S17/H17	S17 and H17	A 0.2-0.3% Difference between the two and basic strategy differs for S17 and H17
Penetration	25%, 50%, 75%, 95%	Changes the odds of the game drastically
Bet Size	Running count, True count, True count*2, True Count +/-2	Significant effect on profit when using card counting
Card Counting	Hi-Lo, Zen Count, KISS 3, Hi-Opt 2, Revere RAPC, Wong Halves	The main purpose of the study is to analyse different methods

Table 2: Rules to be **varied** during simulations, and justification for doing so.

3. Strategies

3.1. Never Bust

The simplest strategy of all is never bust, where the player hits any hand with a total of 11 or less, and always stands at a hard total of 12 or more. This is self-evidently a poor strategy because it restrains the player from many opportunities to get a blackjack or get close to it. For example, if the player had a total of 12, they would need any card with a value of 9 or less to get closer to 21, which would increase their chance of winning. (Kendall and Smith, 2003) suggests that the casino advantage can be as high as 8% when using the never bust strategy.

3.2. Mimic Dealer

Mimic the dealer is the strategy where you follow the rules the casino enforces on the dealer. So, in a game of S17, you will always hit a total of 16 or less and always stand at 17 or more. This also means that the player never doubles or splits, because the dealer is also not permitted to do so. (Conway and Koehler, 1998) suggests that the expected casino advantage with mimic the dealer strategy is 5.5%.

3.3. Basic Strategy

We can see that from the two strategies discussed already, the player is bound to lose a significant amount of money. Basic strategy is a way of playing each possible hand combination in blackjack to minimize your loss. The strategy was created through the simulation of several hundred million hands. Through a method of computer trial and error, it found the best decisions for the player given every possible hand. In 1966 Edward O. Thorp introduced such a strategy printed in a table format. By using basic strategy you are able to bring the casino advantage down to as little as 0.5%, and possibly break even with the casino when the game is played with 1 deck.

There are some small differences between basic strategies from different sources, but they are generally very similar. Basic Strategy also varies depending on a few other factors, such as the game type (S17 or H17) and based on the count when using card counting.

HARD TOTALS

	DEALER UPCARD									
	2	3	4	5	6	7	8	9	10	A
17	S	S	S	S	S	S	S	S	S	S
16	S	S	S	S	S	H	H	H	H	H
15	S	S	S	S	S	H	H	H	H	H
14	S	S	S	S	S	H	H	H	H	H
13	S	S	S	S	S	H	H	H	H	H
12	H	H	S	S	S	H	H	H	H	H
11	D	D	D	D	D	D	D	D	D	D
10	D	D	D	D	D	D	D	D	H	H
9	H	D	D	D	D	H	H	H	H	H
8	H	H	H	H	H	H	H	H	H	H

SOFT TOTALS

	DEALER UPCARD									
	2	3	4	5	6	7	8	9	10	A
A,9	S	S	S	S	S	S	S	S	S	S
A,8	S	S	S	S	Ds	S	S	S	S	S
A,7	Ds	Ds	Ds	Ds	Ds	S	S	H	H	H
A,6	H	D	D	D	D	H	H	H	H	H
A,5	H	H	D	D	D	H	H	H	H	H
A,4	H	H	D	D	D	H	H	H	H	H
A,3	H	H	H	D	D	H	H	H	H	H
A,2	H	H	H	D	D	H	H	H	H	H

PAIR SPLITTING

	DEALER UPCARD									
	2	3	4	5	6	7	8	9	10	A
A,A	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
T,T	N	N	N	N	N	N	N	N	N	N
9,9	Y	Y	Y	Y	Y	N	Y	Y	N	N
8,8	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
7,7	Y	Y	Y	Y	Y	Y	N	N	N	N
6,6	Y/N	Y	Y	Y	Y	N	N	N	N	N
5,5	N	N	N	N	N	N	N	N	N	N
4,4	N	N	N	Y/N	Y/N	N	N	N	N	N
3,3	Y/N	Y/N	Y	Y	Y	Y	N	N	N	N
2,2	Y/N	Y/N	Y	Y	Y	Y	N	N	N	N

SURRENDER

	DEALER UPCARD									
	2	3	4	5	6	7	8	9	10	A
16								SUR	SUR	SUR
15									SUR	
14										

INSURANCE OR EVEN MONEY: DON'T TAKE

KEY		
	H	Hit
	S	Stand
	D	Double if allowed, otherwise hit
	Ds	Double if allowed, otherwise stand
	N	Don't split the pair
	Y	Split the Pair
	Y/N	Split only if 'DAS' is offered
	SUR	Surrender

Figure 2: Basic Strategy Table extracted from (Blackjack Apprenticeship, 2020a)

Figure 2 shows a basic strategy table from the blackjack apprenticeship. This is the basic strategy of choice for the simulations I will be using. Alterations to the basic strategy will be explored in the further investigation section.

4. Card Counting

Card counting is a method of determining whether the player or the dealer has the advantage for the next hand. The idea of card counting is that high cards benefit the player, while the low cards will benefit the dealer. High cards benefit the player as they increase the chance of hitting blackjack, increase the expected value of doubling and provide better splitting opportunities, as well as increasing the likelihood the dealer will bust. On the other hand, low cards benefit the dealer because they mean it is less likely that the dealer will bust.

Counting systems assign a positive, zero or negative value for every card. Different counting systems will assign different values to each card. Card counters do not memorize every single card that is played, but rather they keep track of the count. When low cards are player they increase the count, and when high cards are played they decrease the count. If the count increases, it becomes more likely that high-value cards will be dealt next; and if the count goes down, it becomes less likely that high-value cards will be dealt next. For example, one of the simplest card counting systems, Hi-Lo, adds one for cards between 2 and 6 and subtracts one for aces and cards with a value of 10. 7s, 8s and 9s are zeros and don't affect the count. (Zimran et al., 2009)

Card Counting methods also have different levels. Counters gauge the effect of removing cards from the deck and how they affect the casino advantage. Some methods use values other than –1,0 and 1 that may be higher or lower, to get a better idea of what to expect. A larger ratio between the low and high values means a higher level. This does have the disadvantage of making it harder for a human to keep track of the count and to play quickly and accurately.

4.1. Terminology

- **Balanced Count** – A counting system which has a balance between plus and minus cards. If an entire deck was counted, using the specific system, the end sum would be zero.
- **Betting Correlation** – The correlation between card point values and the effect of removal of cards. This measures how good the card counting method is at predicting good betting situations. Especially useful in 6 or 8 deck games (Norman Wattenberger, 2021).

- **Betting Spread** – The spread you are allowed to have on your bets. If there is a bet spread of 1-10 it means you will only be able to place a maximum of 10 betting units.
- **Level** - The level of a strategy refers to the number of different values assigned to cards. Higher-level strategies tend to improve the player's edge, but harder to use for most people.
- **Playing Efficiency** – How well a counting system handles alterations in playing strategy. This value is particularly useful for one or two decks.
- **Running count** – Running total that is continually adjusted to the specific card's count value.
- **True count** – The running count divided by the estimated number of decks left in the shoe (typically precise to half a deck).
- **Unbalanced count** – A count which ends up with a value other than zero after all cards in a deck have been played.

4.2. Hi-Lo

The Hi-Lo system is the most used and most basic card counting strategy. Most simulations and studies are based on this count. It was originally created by Harvey Dubner in 1963. It is a Level 1, balanced strategy that is optimized for betting. It has a betting Correlation of 0.97 and a Playing Efficiency of 0.51.

Card Value	**A**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Increment	-1	1	1	1	1	1	0	0	0	-1

Table 3: Hi-Lo Card Counting Values

4.3. Zen Count

Zen count is a slightly more complicated Level 2 card-counting system. This counting system first appeared in Arnold Snyder's book "Blackbelt in Blackjack" (Snyder, 1983).

Zen is another balanced system with a Betting Correlation of 0.96 and a Playing Efficiency of 0.63. The graded values in the system provide greater accuracy, but it is more difficult to keep track of the count.

Card Value	**A**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Increment	-1	1	1	2	2	2	1	0	0	-2

Table 4: ZEN Card Counting Values

4.4. KISS III

KISS III is the latest variation of the popular KISS system. Unlike KISS II, the latest version is appropriate to use in a multideck game. These card counting systems are impressive as they provide the player with a plan which is both simple and efficient. KISS, being a level-1 system with a Betting Correlation of 0.98 and a Playing Efficiency of 0.56, appears to be an attractive option. meaning that certain cards are counted differently, depending on their suit. In this case, the count value of the 2s is variable.

Card Value	**A**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Increment	-1	0/1	1	1	1	1	1	0	0	-1

Table 5: KISS 3 Card Counting Values

4.5. Hi-Opt II

The Hi-Opt system is one based on Edward Thorp's Hi-Lo Count and was created by Charles Einstein in 1968. The system was then adjusted slightly in the 1980 book "The World's Greatest Blackjack Book" (Humble, 1980) to come up with Hi Opt I. Lance Humble later adjusted this system further and came up with the highly effective Hi-Opt II. This counting system, which is a balanced Level 2 strategy, has been found to be more powerful than some Level 3 card counting strategies. With a Betting Correlation of 0.91 and Playing Efficiency of 0.67, it was originally developed for single deck games but works very well in games with more decks.

Card Value	**A**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Increment	0	1	1	2	2	1	1	0	0	-2

Table 6: Hi-Opt Card Counting Values

4.6. Wong Halves

Wong Halves is a very complicated Level 3, balanced counting system. It is suitable for experienced players, and if used correctly should be able to give the players significatn profit. Its complexity lies in the fact that the count uses fractions, making counting more difficult. This method was first seen in the book "Professional Blackjack," published in 1975. With a Betting Correlation of 0.99 and Playing Efficiency of 0.56, it appears to be one of the best card-counting methods. .

Card Value	**A**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Increment	-1	.5	1	1	1.5	1	.5	0	-.5	-1

Table 7: Wong Halves Card Counting Values

4.7. Revere (Advanced Point Count) RAPC

Revere RAPC and 14 Count are incredibly complex balanced card counting strategies. These level 4 counts were developed in 1971 and 1974 and were used by professionals in the early days of counting. Reverse RAPC will be explored in the simulations. With its high complexity, Betting Correlation of 1.00 and Playing Efficiency of 0.53 it should produce very profitable results. (Vidámi et al., 2020)

Card Value	**A**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Increment	-4	2	3	3	4	3	2	0	-1	-3

Table 8: Revere RAPC Card Counting Values

4.8. Betting Systems

To realize a profit when card counting, we need to vary our bets. By keeping track of the running count throughout the game the player is then able to use this count to determine the optimal amount of money to bet. Typically, the running count is used to calculate a true count. This is simply the running count divided by the estimated number of decks remaining. If there is a true count of 5 that suggests the game is very favourable to the player and they should place a bet that is 5 times the size of their minimum bet. This is the most common way of betting. However, there are various betting ramps people suggest using. The true count is not only used for placing bets. It is also used to deviate from basic strategy and change some of the ways the players play certain hands to further increase their advantage. (Vaidyanathan, 2014)

5. Implementation

The Blackjack game and simulations were implemented in the Python programming language. The main idea was that the game can be implemented in a modular way so that the game rules, playing strategies, counting methods and others can easily be varied during the simulations. Classes were used to represent Card, Deck, Hand and Player, with the Game itself being implemented in a separate file implementing all these classes and extra functionalities such as recording the bankroll at end of 100 hands. Several different functions were also used throughout, the most important of which will be discussed below in more detail. The UML diagram outlines all classes, functions and fields.

5.1. UML

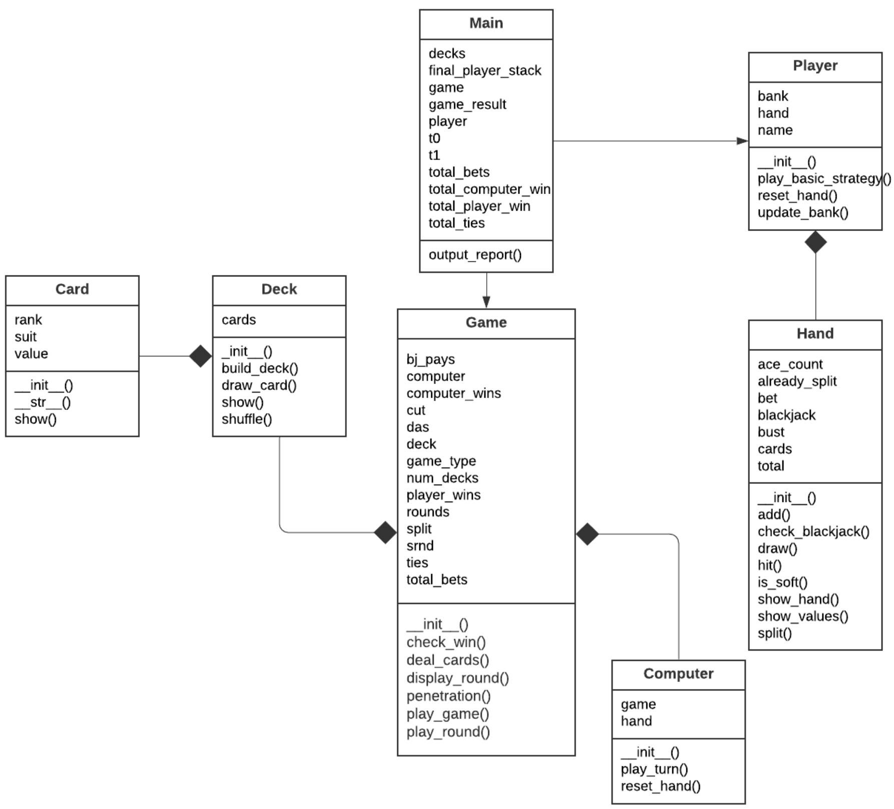

Figure 3: UML Diagram for Blackjack Simulations

Card

The card class is as simple as possible. It contains the rank of the card, the suit and the value for the game of blackjack.

Deck

The Deck class is slightly more complex, containing functions to build a single deck, build multiple decks, shuffle the deck, draw a card and determine the shoe size. Building multiple decks simply takes the size, e.g. 6 decks and runs the single building function a corresponding number of times..

```
class Deck:
    def __init__(self, size):
        self.cards = []
        self.build(size)
```

The function to build a single deck takes no parameters and is comprised of two loops to create the 52 cards in a deck and assign their values.

```
def build_deck(self):
    for s in ["Spades", "Clubs", "Diamonds", "Hearts"]:
        for r in ["2", "3", "4", "5", "6", "7", "8", "9", "10",
            "Jack", "Queen", "King", "Ace"]:
            if r.isdigit():
                val = int(r)
            elif r == "Ace":
                val = 11
            else:
                val = 10
            self.cards.append(Card(r, s, val))
```

All the cards in a deck are stored in an array of Card objects. Shuffling is done using the built-in shuffle function in the python random package. The ability for the ace card to be either 1 or 11 is handled in the Hand class.

Hand

Since a hand in blackjack can have many different cards and options to hit, double, split and stand, it has been dedicated its own class. The hand class has the following fields: cards - an array of cards, total - current total for hand, bust - True or False, ace_count - number of aces, already_split - True or False, bet: the bet for the hand, blackjack - True or False. Functions other than the hit, double, split and stand, which are discussed in detail below are: checking for blackjack, adding a card and checking if the hand is soft.

```
class Hand:
    def __init__(self):
        self.cards = []
        self.total = 0
        self.bust = False
        self.ace_count = 0
        self.already_split = False
        self.bet = 0
        self.blackjack = False
```

Hit

The hit function takes a deck parameter from which it should draw the card and then pops the top card. If the card is an ace and the count would go above 21 it counts the ace as 1 instead of 11. If the total goes over 21 then bust is set to True, and if it doesn't it simply adds the value of the card to the total.

```
def draw(self, deck):
    drawn_card = deck.draw_card()
    self.cards.append(drawn_card)
    if drawn_card.value == 11:
        self.ace_count += 1
    if self.total + drawn_card.value > 21 and self.ace_count >
        0:
        self.ace_count -= 1
        self.total -= 10
        self.total += drawn_card.value
    elif (self.total + drawn_card.value) > 21 and
        self.ace_count == 0:
```

```
        self.total += drawn_card.value
        self.bust = True
    else:
        self.total += drawn_card.value
    self.check_blackjack()
    return drawn_card
```

Split

The split function has to split the two cards in the hand into two hands and draw a card into each one. It then checks for blackjack in each hand and finally returns the new hand.

```
def split(self, deck):
    a = self.cards.pop()
    self.total -= a.value
    if a.value == 11:
        self.total += 10
    self.draw(deck)
    self.check_blackjack()
    new_hand = Hand()
    new_hand.add(a)
    new_hand.draw(deck)
    new_hand.check_blackjack()
    return new_hand
```

Double

When it comes to doubling all we need to do is double the bet amount and then hit. The only thing to ensure is that the player is only able to double if it is their first two cards and we achieve this by checking if the hand size is equal to 2.

```
elif turn == "D" and len(hand.cards) == 2:
    hand.bet *= 2
    self.update_running_count(hand.hit(deck))
```

Player

The player is also another simple class that stores the bankroll and the hand of the player, with two functions to update the bankroll and to reset the hand. Code Example 1

Dealer

Like the player, this class contains the hand for the dealer and it is just missing the bankroll as we do not need to keep track of the dealer's money. Code Example 2

Basic Strategy

The basic strategy was implemented using 2-dimensional arrays and a number of conditional statements. I defined 3 arrays representing basic strategy for pairs, soft_totals and hard totals. Below is the soft array to demonstrate the structure.

```
soft_total=[['S','S','S','S','S','S','S','S','S','S'],#Soft 20
            ['S','S','S','S','S','S','S','S','S','S'], #Soft 19
            ['D','D','D','D','D','S','S','H','H','H'], #Soft 18
            ['H','D','D','D','D','H','H','H','H','H'], #Soft 17
            ['H','H','D','D','D','H','H','H','H','H'], #Soft 16
            ['H','H','D','D','D','H','H','H','H','H'], #Soft 15
            ['H','H','H','D','D','H','H','H','H','H'], #Soft 14
            ['H','H','H','D','D','H','H','H','H','H']] #Soft 13
```

Then the arrays are used in the basic strategy function which returns a letter: "S"-Stand, "H"-Hit, "D"-Double, "Y"-Split. The rules are as follows: if card values are equal look in pair array, if the hand has an ace look in the soft array, if it doesn't have an ace look in the hard array and if the hand is already split we can't split any longer.

```
def basic_strategy(hand, up_card):
   card1 = hand.cards[0].value
   card2 = hand.cards[1].value
   if hand.total == 21:
      choice = "S"
   elif len(hand.cards) == 2 and card1 == card2:
      choice = pair_splitting[11 - card1][up_card - 2]
   elif hand.ace_count == 1:
      choice = soft_total[10 - (hand.total - 10)][up_card - 2]
```

```
    else:
        if hand.total > 17:
            choice = "S"
        else:
            choice = hard_total[17 - hand.total][up_card - 2]
    if hand.already_split and choice == "Y":
        if hand.total > 17:
            choice = "S"
        else:
            choice = hard_total[17 - hand.total][up_card - 2]
    return choice
```

No Bust Strategy

The No Bust Strategy is a simple strategy as discussed earlier in this report. The function only returns either "S"-Stand or "H"-Hit. If the hand total is more than 11 then return "S" else return "H".

```
def no_bust_strategy(hand, up_card, das):
    choice = "S"
    if hand.total > 11:
        choice = "S"
    elif hand.total <= 11:
        choice = "H"
    return choice
```

Mimic Dealer

Mimic the Dealer is exactly the same as No Bust Strategy except for the value that determines whether we stand or we hit is now 17.

Card Counting Methods

In order to represent each individual card counting strategy I decided to use a dictionary where each card will have a value assigned to it which will affect the count. Here is an example of the Hi-Lo count.

```
hi_lo = {
    "name": "Hi-Lo", "2": 1, "3": 1, "4": 1, "5": 1, "6": 1,
        "7": 0, "8": 0, "9": 0, "10": -1,
    "Jack": -1, "Queen": -1, "King": -1, "Ace": -1,
}
```

Updating Running Count

Later in my main game class, I use the predefined dictionary to look at the card and simply add the value associated with it to the running count.

```
def update_running_count(self, card):
    val = self.card_counting[card.rank]
    self.running_count += val
```

Game Class

The game class is the main part of the simulation code and keeps track of a lot of information. It takes in the strategy (basic/no bust/mimic), the number of decks (1/2/6/8), game type (S17/H17), penetration (25%/50%/75%/90%), bet size (true/running etc.) and the counting method, which is a dictionary representing card value for each card that appears like the example above.

```
class Game:
    def __init__(self, strategy, num_decks, game_type,
        penetration, bet_size, card_counting):
```

The play_round() function does as its name suggests, i.e. it plays a single round of blackjack. First of all, it deals the cards both to the player and to the dealer, updating the count each time. It then runs the code to decide the player's next move depending on the strategy we are using and the current cards that have been drawn. The dealer then plays his turn, after which all of the cards are added to a history of bets, the winner is checked and betting is adjusted accordingly. Finally, both the player and dealer's hands are reset. Code Example 3

The main portion of this class is the play_game() function, which uses 2 loops, the first of which checks if the round limit has been reached. If it is not yet reached, we play a round of the game. The second loop checks if all of the cards have been played; if they have, it resets the count and reshuffles the deck. Finally, the function returns all of the relevant information to be stored. Code Example 4

6. Simulation Results

6.1. Simulation Design

All playing conditions under Table 1 were kept the same throughout all simulations. Basic strategy, Never Bust and Mimic Dealer were simulated without card counting. This is done simply to demonstrate that Basic Strategy is the best way to play and that other methods are not good enough. The rest of the variations are all performed in a combination. For example, a single combination will be a game simulated with 1 deck, playing S17, 50% deck penetration, true count, Hi-Lo counting. Details of each variation that will be used can be found in Table 2. The total number of combinations from all variations comes out to be 960 (not including pure basic strategy), which is calculated as follows:

$$\mathbf{960} = 4(No.decks) \times 2(S17/H17) \times 4(deckpenetration) \times 5(betting) \times 6(cardcounting)$$

Each simulation will be performed under the same simulation conditions:

- 10,000 games played for each simulation
- 100 hands in each game
- Initial Bankroll of 10,000 units
- Each bet unit is 100
- Bet Spread Limited to 1-15
- Same random seed for all simulations
- Same process for shuffling the decks

Single simulations for different strategies, counting methods, number of decks, shuffle points etc. Will be performed for comparison under the exact same rules to allow us to see how the player's bankroll changes over the period of 100 hands. In order to compare all of the different scenarios and playing strategies the average profit will be used. This will be the profit the player is expected to make, on average (out of the 10,000 samples), after playing 100 hands of blackjack. For example, if the average bankroll of the player at the end is 10200 this would mean the average profit is 2%. Which is calculated using the following formula:

$$\textbf{Average Player Profit} = \frac{AverageEndingBankroll - InitialBankroll}{InitialBankroll} \times 100$$

6.2. Basic Strategy, Never Bust and Mimic Dealer

The first thing to test was the strategy used in blackjack. From current research and findings, it is already known that basic strategy should come out on top, but it will still be useful to compare our results and see how significant the difference really is. The initial test was to see how the player's bankroll changes over time during the game. For this experiment 6 decks, S17, with a penetration of 75% was used, varying only the strategies. The average bankroll after each hand was taken from the 10,000 simulations.

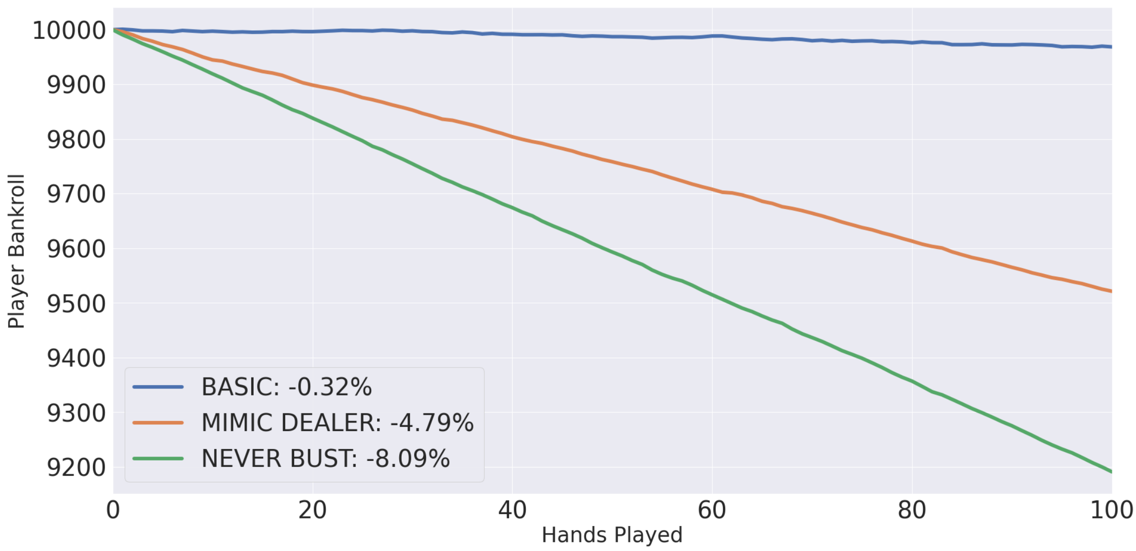

Figure 4: Comparison of Blackjack strategies - average bankroll across 100 hands

After the initial test, further simulations were performed to ensure the results were similar when using different rules as well. Each playing strategy was performed under a total of 32 conditions as card counting methods and betting strategies are not considered yet. Each was tested under 1, 2, 6 and 8 decks, as well as using a deck penetration of 25%, 50%, 75% and 90% respectively. The mean casino advantage for each strategy is shown in Table 9.

Strategy	**Average Profit**
Basic	-0.34%
Mimic Dealer	-5.05%
Never Bust	-8.01%

Table 9: Summary of strategy comparisons

From the results, we can see that Basic strategy is the least disadvantageous to the player and has brought the player's average profit down to -0.34%, almost breaking even. This is of course to be expected as it was created using mathematical probabilities. Never Bust was the worst strategy to use and resulted in an expected average profit of -8.1%. Once again as expected, Mimic the Dealer strategy was slightly better than never bust, but was nowhere near the basic strategy. After proving that basic strategy is the optimal way to play blackjack, and since it has such a significant edge over other popular methods, it will be used as the strategy of choice for the rest of the experiments. Using any of the other methods further would not be useful or practical in any way as they provide the casino with a very large advantage.

6.3. S17 vs H17

The next step was to compare the two playing strategies the casino might enforce on the dealer. We would expect that S17 is better for the player as that means the dealer is less likely to get a high total because they stop taking cards if they reach 17. The simulations suggest this as well, however, the difference was not very large. When playing S17 with basic strategy and 6 decks with 75% penetration we saw the player's advantage was -0.32% and when playing H17 we see the profit goes down to -0.50% on average.

When comparing the two strategies under different conditions like before, we saw the same trend, in that S17 with an average of -0.34%, had a slight edge (0.10%) over H17's -0.44% profit. Both game types are popular in casinos. Hence the simulations to follow will use both S17 and H17, which will also provide us with a larger sample of data.

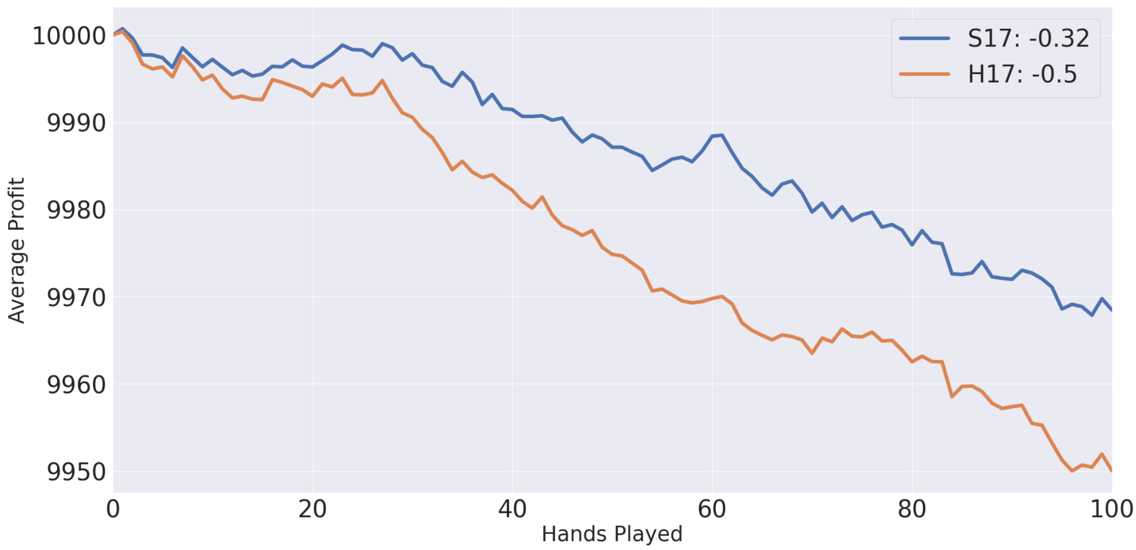

Figure 5: The average bankroll of S17 and H17 across 100 hands

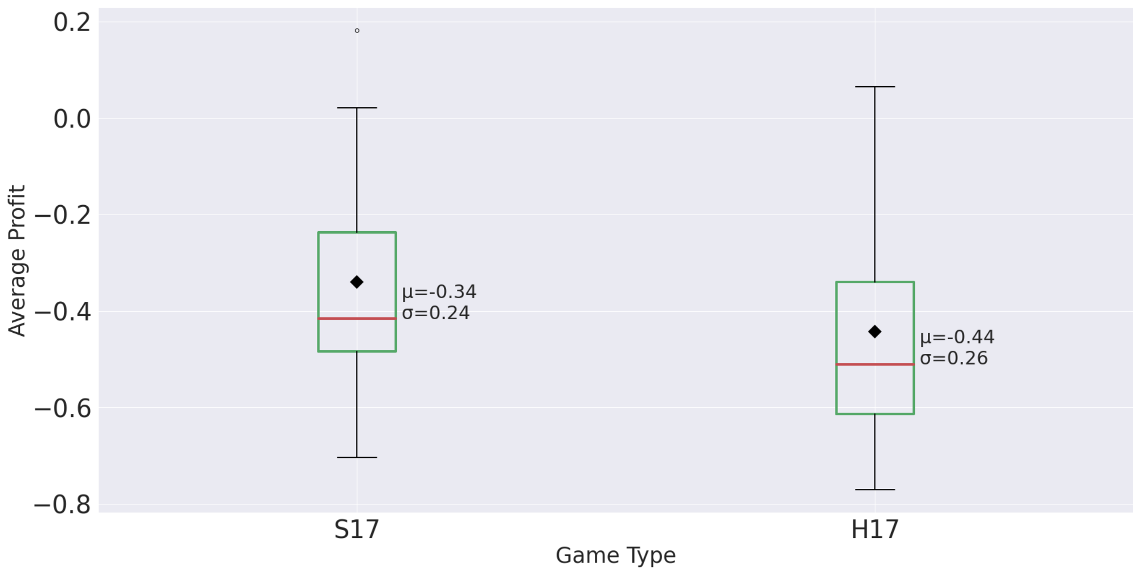

Figure 6: Box Plot showing the average profit distribution of S17 and H17 across all simulated playing conditions ($\mu = mean, o = median$)

6.4. Card Counting

Since we have proven that Basic strategy provides the player with the best odds of winning, and we know that S17 is the slightly more beneficial game type for the player, we can now explore which counting method is the best. The most commonly used deck penetration tends to be around 75% or 4.5 out of 6 decks and most card counting methods tend to use the true count for determining bet size. The effect of these factors will be explored further later on, but for now, these playing conditions will be used to allow us to compare the counting methods themselves.

Basic Strategy Only

First, we looked at pure basic strategy with no card counting so we have a baseline. As we have already discovered, basic strategy alone under the specified conditions produces an average profit of -0.32%, but Figure 7 shows us the way the player bankroll varies over the 10,000 simulations. This type of simulation was performed for all other simulations and the graphs looked very similar. We can see in the two graphs that when counting cards the bankroll can vary a lot more than with pure basic strategy.

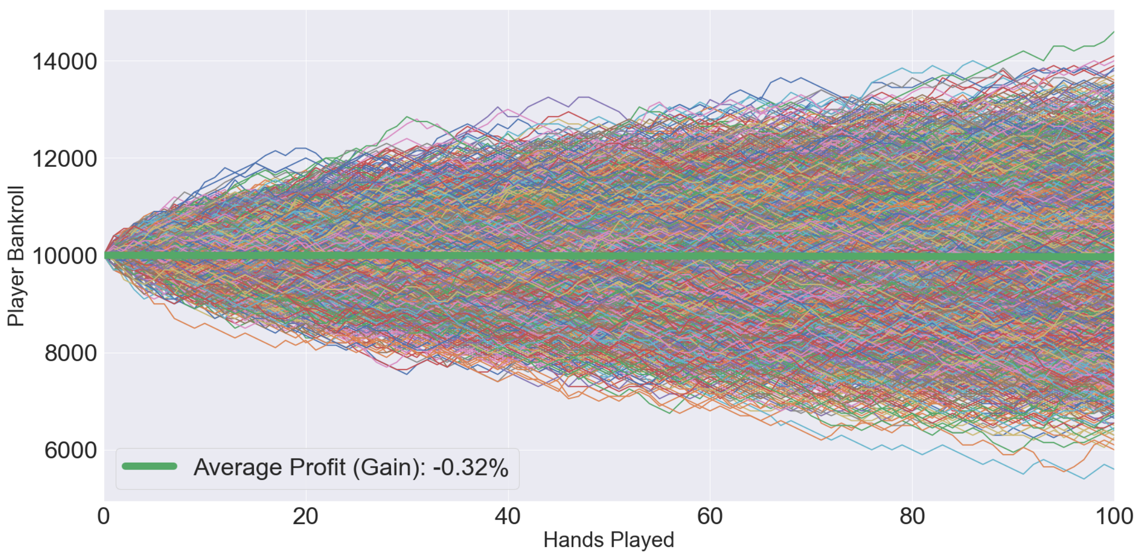

Figure 7: No Card Counting Bankroll across 10,000 simulations of 100 hands results

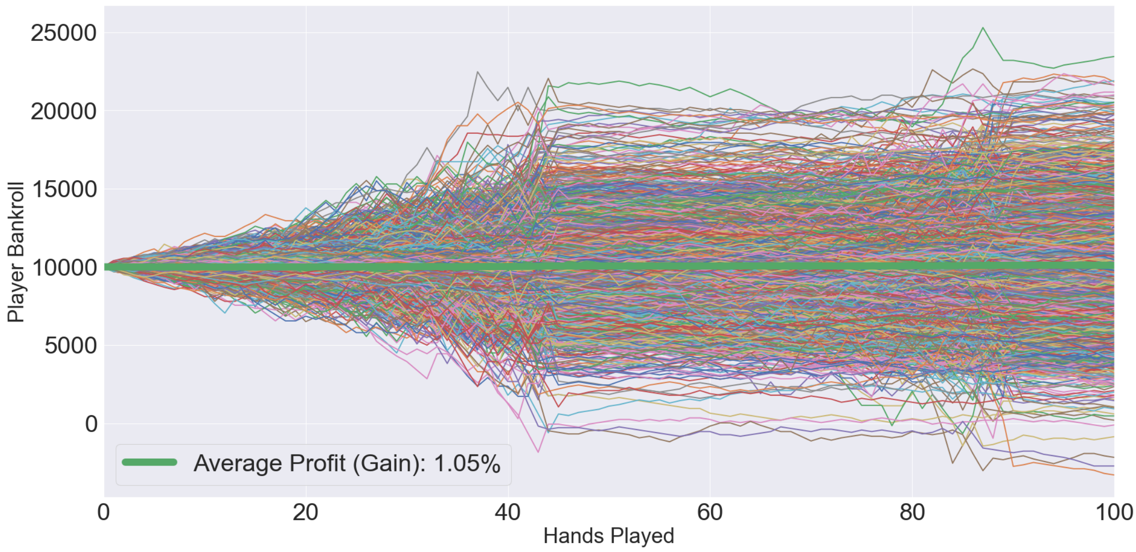

Figure 8: Hi-Lo Bankroll across 10,000 simulations of 100 hands results

Comparison

The counting methods all produced a positive result for the player and an increase in profitability. All methods had a profit of over 1% except for KISS III being slightly under. The best performing method was the most complex, Revere RAPC with an average profit of 2.49%. Zen Count, however, was not far below, with a profit only 0.36% off Revere RAPC, and considering its much lower complexity, it seems like a very good option. Hi-Opt II was another 0.27% lower than Zen and the rest were quite close together Figure(9).

When we look at the bigger picture where we run all card counting methods under multiple different playing scenarios we notice similar results (Figure 10). We now notice a very wide spread, suggesting that it is possible not only to make a lot more profit than the average, but also to lose a lot more. However, it should be noted that these results include games players will probably never actually play, such as 25% deck penetration or using the running count as the betting scale.

This box plot still shows us that Revere RAPC still has the highest average, followed by Zen, but we also notice that Hi-Lo had the worst performance. Despite RAPC having a very high average it also had a very high standard deviation of 4.77. Across all simulations Revere has an even bigger average profit compared to Zen Count, however, Zen has a lower standard deviation of 3.09. The median values are very closely aligned with the mean and also suggest Revere is the best card counting method. One very clear thing is that card counting increases your risk significantly and the bankroll will fluctuate a lot more than when playing only basic strategy.

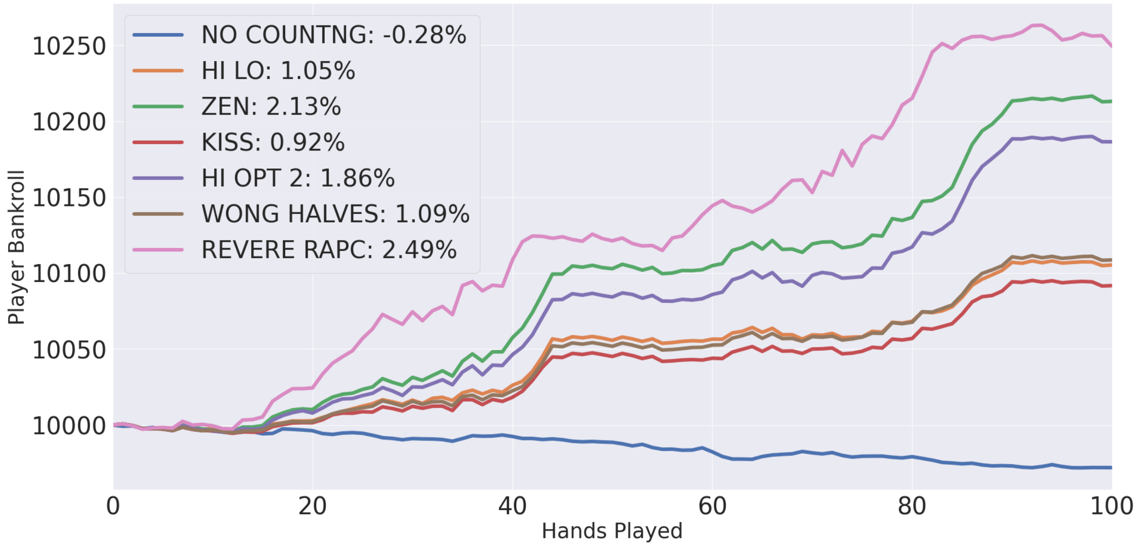

Figure 9: Average bankroll for all card counting methods across 100 hands

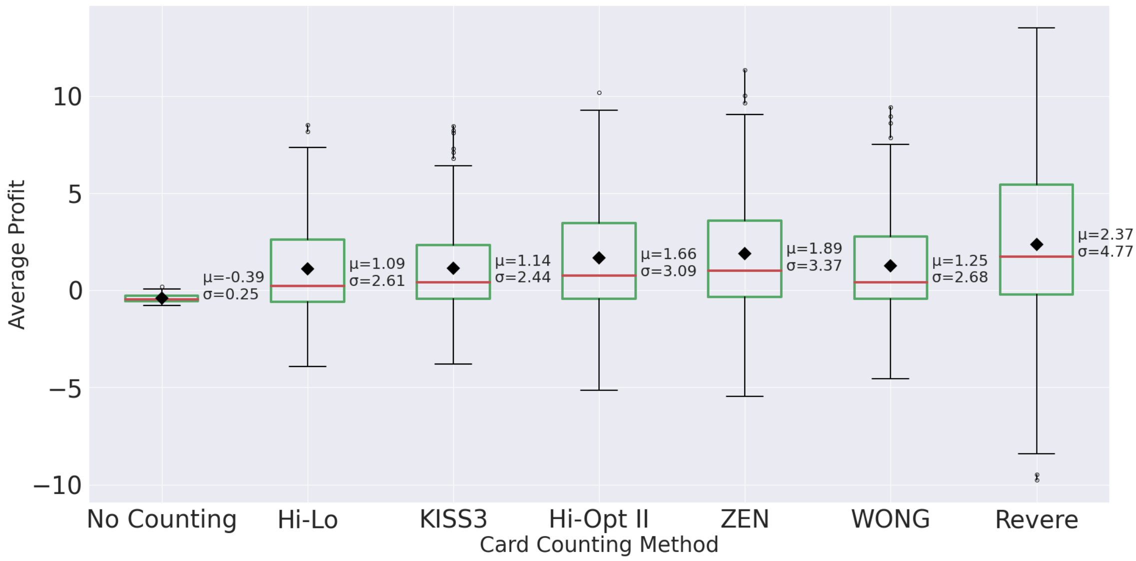

Figure 10: Box plot showing the average profit distribution of all card counting methods across different playing conditions. ($\mu = mean, o = median$)

7. Further Analysis

In order to gain a better understanding of the effect on a player's advantage of specific play variations, further experiments were carried out. This will help determine which elements of the game have the biggest effect. For this portion, the RAPC counting method will be used, since it produced the best results out of the 6 methods we have compared. All variables will again be kept the same except for the factor that is being analysed (i.e. Number of Decks, Deck Penetration, Bet System etc.).

7.1. Number of Decks

In order to discover the significance of the number of decks used to play, two tests will be performed: one with pure basic strategy and no card counting, and one where we use the Revere counting system. This will give us an indication of how the number of decks affects the player's edge both when card counting and simply playing basic strategy. Playing conditions are basic strategy, S17, 75% penetration and true count for bets.

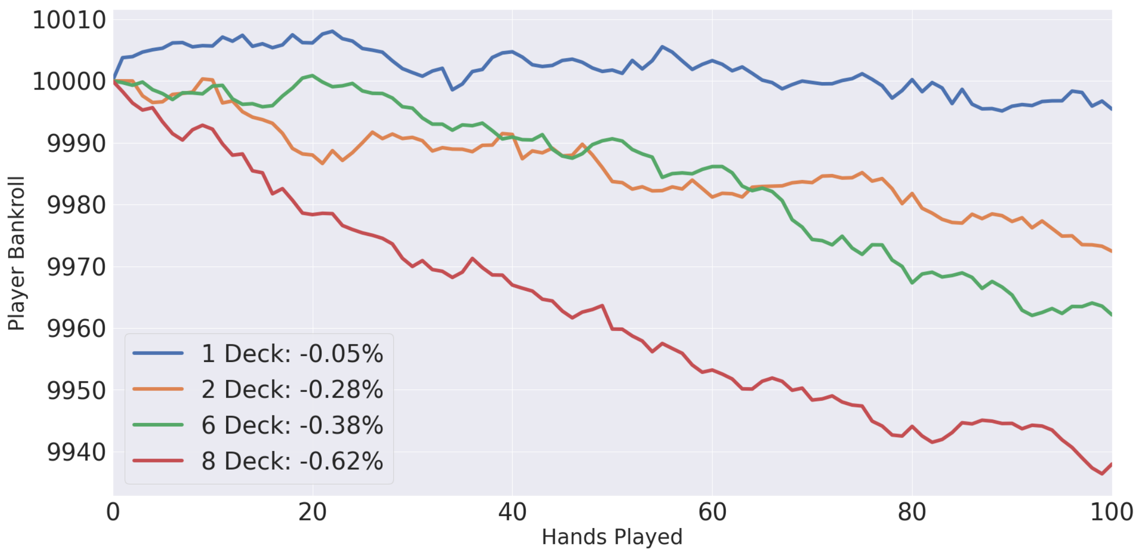

Figure 11: Varying Number of Decks: Average bankroll across 100 hands using only Basic Strategy

With basic strategy only, in Figure 11, we notice a very clear trend that as the number of decks decreases, the player's advantage increases over the casino, with 1 deck being best for the player and 8 being the worst. When it comes to card counting, the number of decks seems to have an even more significant role, with an increase of 7.9% between 1 and 8 decks. The Revere RAPC here gives the player an 8.98% average profit with 1 deck.

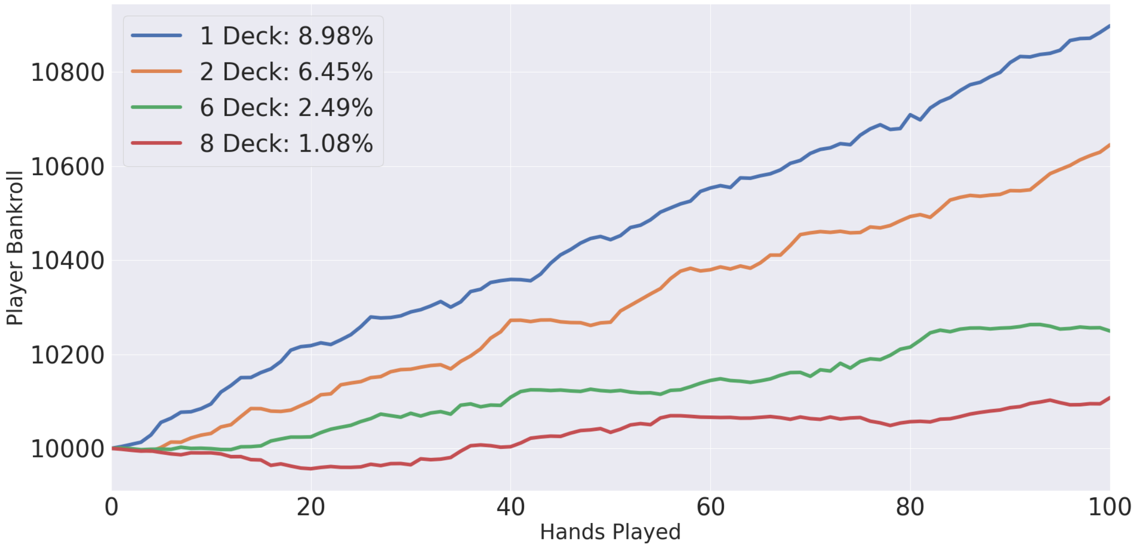

Figure 12: Varying Number of Decks: Average bankroll across 100 hands using Revere RAPC

In order to further confirm that the average profit is inversely proportional to the number of decks, the following table shows the average profit for 1, 2, 6 and 8 decks, across all the playing conditions and all the card-counting methods that we have simulated. Once again it should be noted that these results include games with 25% penetration and running count as the bet size, hence the negative profits.

Number of Decks	**Average Profit**
1 Deck	4.52%
2 Deck	2.39%
6 Deck	-0.05%
8 Deck	-0.59%

Table 10: Average Profit across all simulations with different number of decks

7.2. Deck Penetration

Deck Penetration refers to the point at which the deck is shuffled. Shuffling the deck is important in card counting, as it can help to determine how many high-value cards are left in the deck. The closer to the end of the deck the shuffle occurs, the more accurate the count will be. The investigation looked into how this aspect affects the player's expected return. The shoe was shuffled after playing 25%, 50%, 75% and 90% of the cards. A game with 6 decks and Revere RAPC count was used for the initial experiment, with the average results of all combinations being shown in Table 11

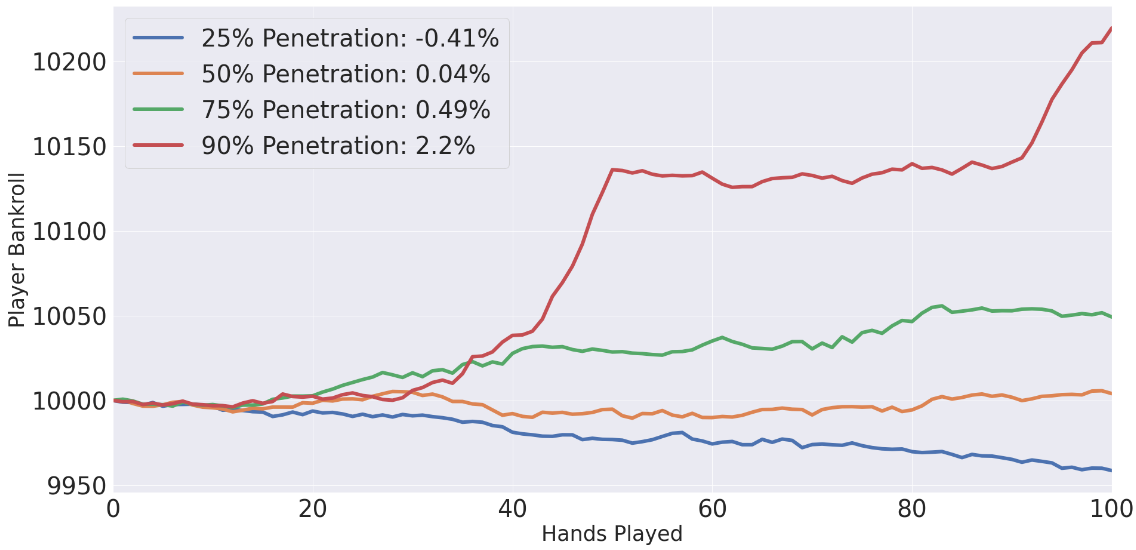

Figure 13: Varying Deck Penetration: Average bankroll across 100 hands using Revere RAPC

Deck Penetration	Average Profit
0.25%	-0.35%
0.50%	0.98%
0.75%	2.41%
0.90%	3.23%

Table 11: Average Profit across all simulations with different number of decks

The results we see on the graph indicate that it is advantageous for the player that the shuffle happens after a larger number of cards have already been played. If the deck is shuffled after playing only 25% of the cards, we see that the player loses money, with an average profit of -0.41%. When 50% of cards are played the player can come out very slightly on top, and when we reach 75% we start to see an actual profit. With 90% penetration, we notice a very large jump with a 2.2% average profit. We can also confirm that this is true for all other counting methods and combinations of rules from Table 11.

7.3. Betting Systems

When playing blackjack with card counting, it is most important to vary the size of the bet, such that the bet is increased when the player has an edge. Failure to do so means that it is pointless to keep track of the count. The betting strategy employed by the great majority of card counters is to use the "True Count", which is simply the running count, divided by the estimated number of decks remaining to be played. However, for this experiment, slight variations of the true count were used, in order to test whether there exists a more efficient betting strategy. These variations are: (a) the Running Count; (b) the True Count multiplied by 2; (c) the True Count plus 2; (d) the True Count minus 2. The initial experiment (Figure 14) was done using Revere RAPC and the results from all simulations are shown in (Figure 15).

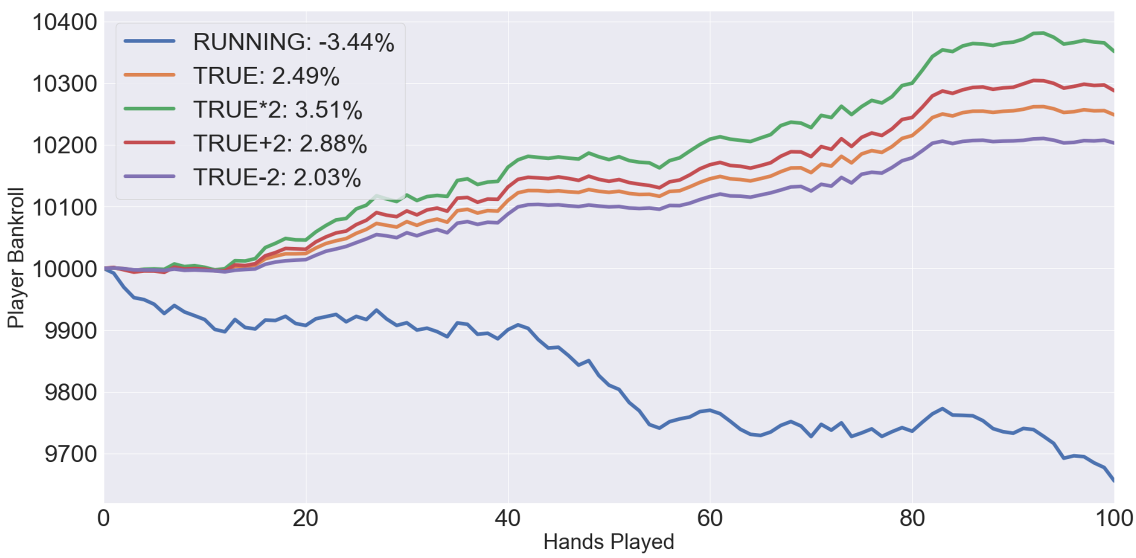

Figure 14: Varying Bet Size: Average bankroll across 100 hands using Revere RAPC

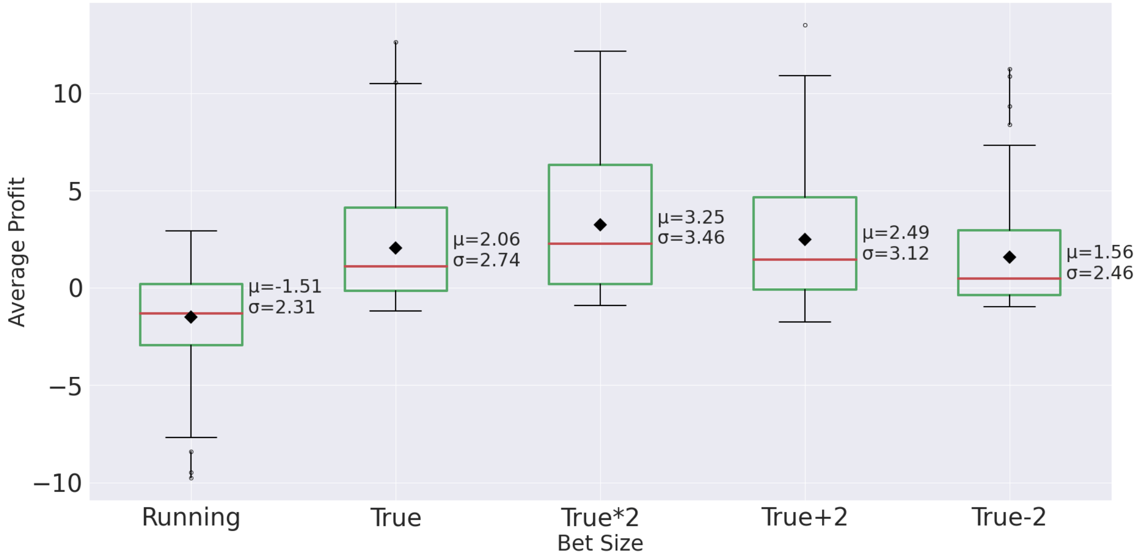

Figure 15: Box plot showing the average profit distribution for different bet sizes across all card counting methods across all simulations ($\mu = mean, o = median$)

The results from the initial test show us that $TrueCount \times 2$ is the best betting system on average for the Revere counting method, with the standard true count coming second. The rest of the systems performed worse than the true count. It could have been that the $TrueCount \times 2$ simply performed better when using Revere, however, we notice similar results when using all counting methods and all combinations. $TrueCount \times 2$ is again the best with a 3.25% average profit and interestingly, on average, across all simulations, $TrueCount - 2$ performed better than $TrueCount$ only. By multiplying the true count by 2 the player is taking a bigger risk, and the results suggest that it pays off. The two graphs also confirm that running count is not a profitable option.

7.4. Bet Spread

"Bet Spread" refers to the distribution in the size of bets. When playing at a table where the minimum bet is £100, a "1-10 bet spread" means that the minimum bet is £100 and the maximum is £1000. When the true count is below 1, the player should always bet £100. The most important part of the strategy is the size of the bet when the true count is 1 or above. The size of the bet is proportional to the true count. This is the central reason for card counting, and it is the reason why the bet spread plays such an important role in blackjack. For these experiments, we will examine spreads of 1-5, 1-10, 1-15, 1-30, and uncapped, where the bet is equal to the true count with no specific limit. All other simulations in this report were performed with a 1-15 bet spread. A 1-5 bet spread is a very conservative, while 1-10 and 1-15 are more commonly used bet spreads, with 1-30 being very risky.

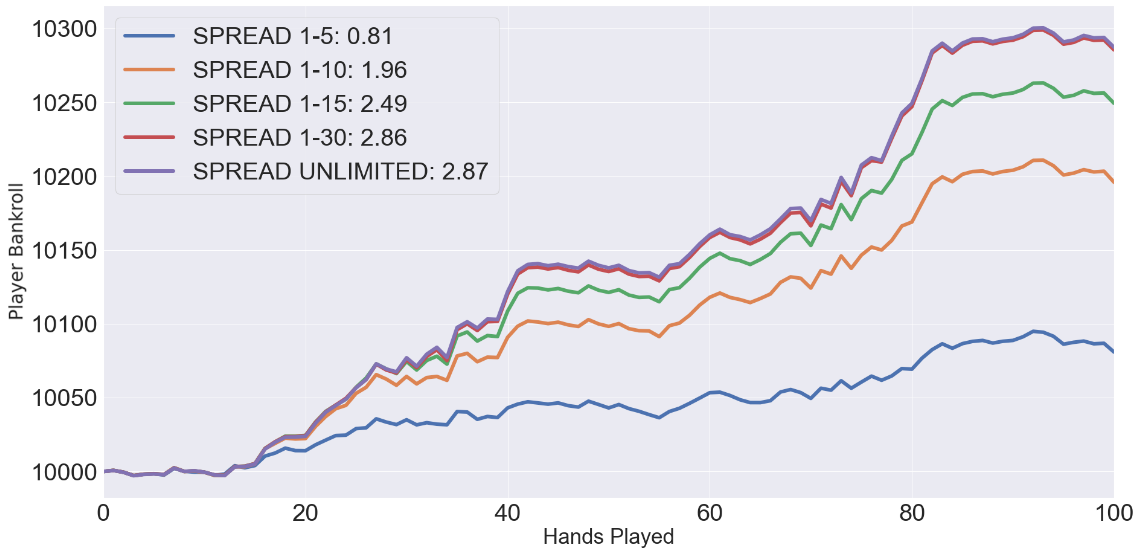

Figure 16: Varying Bet Spread: Average bankroll across 100 hands using Revere RAPC

From these results, we notice a clear trend that the wider the bet spread, the larger the player's profit becomes. A 1-5 bet spread produced only a 0.81% advantage, whereas 1-10 produced more than double that at 1.96%. Going up to 1-15 again shows a large increase of 0.53%. Going above that again sees an increase in the advantage of 0.38%, however, this increased risk might not be something the player would like to take. It is also very clear that going above 1-30 bet spread has no positive effect, and in any case it is most unlikely that any player would consider such a wide bet spread.

7.5. Basic Strategy Deviations

Colin Jones from (Blackjack Apprenticeship, 2020b) says that "Roughly 60-80% of a card counter's edge comes from PERFECT basic strategy, PERFECT counting, and PERFECTLY betting according to the true count. Then, deviations account for the remaining 20-40%". This provoked a lot of interest for me, since it appears to be a very bold claim, suggesting that it is very important to use deviations from Basic Strategy when counting cards. The reasoning behind the deviations is that they lower the risk and variance by playing the move with the highest chance of winning. For example, in the case of 12 vs 2, if the true count is below 3, the player will make more money in the long run if they always hit, but if the true count is above 3 the player will make more money if they always stand. A table showing the suggested deviations can be seen in Figure 17

PLAY	INDEX NUMBER	ACTION
Insurance	+3	*Take insurance at +3 or higher*
16 vs 9	+5	*Stand at +5 or higher*
16 vs 10	0	*Stand at 0 or higher*
15 vs 10	+4	*Stand at +4 or higher*
13 vs 2	–1	*Stand at –1 or higher; otherwise hit*
13 vs 3	–2	*Stand at –2 or higher; otherwise hit*
12 vs 2	+4	*Stand at +4 or higher*
12 vs 3	+2	*Stand at +2 or higher*
12 vs 4	0	*Stand at 0 or higher*
12 vs 5	–1	*Stand at –1 or higher; otherwise hit*
12 vs 6	–1/–3*	*Stand at –1/-3 or higher; otherwise hit*
11 vs A	+1/–1	*Double Down at +1/–1 or higher*
10 vs 10	+4	*Double Down at +4 or higher*
10 vs A	+4/+3	*Double Down at +4/+3 or higher*
9 vs 2	+1	*Double Down at +1 or higher*
9 vs 7	+4	*Double Down at +4 or higher*
10-10 vs 5	+5	*Split at +5 or higher*
10-10 vs 6	+5	*Split at +5 or higher*

*Where two indices are shown, the second is for h17.

Figure 17: Deviations from Basic Strategy in blackjack (Tamburin, 2022)

These deviations were designed to be used with the Hi-Lo counting method so the first experiment will be done under the following conditions: Basic Strategy with Deviations, S17, 6 decks, 75% penetration, true count. This will be compared with the pure basic strategy.

Unfortunately, the results from the simulations did not align with the expectations and in fact deviating from basic strategy resulted in a 0.28% decrease in the average profit. This could be due to the fact that most deviation strategies include insurance bets and surrenders, which are not part of this implementation of blackjack. It is also possible that that deviations perform better when the number of decks in play is smaller, so the same test was also performed for 1, 2, 4 and 8 decks.

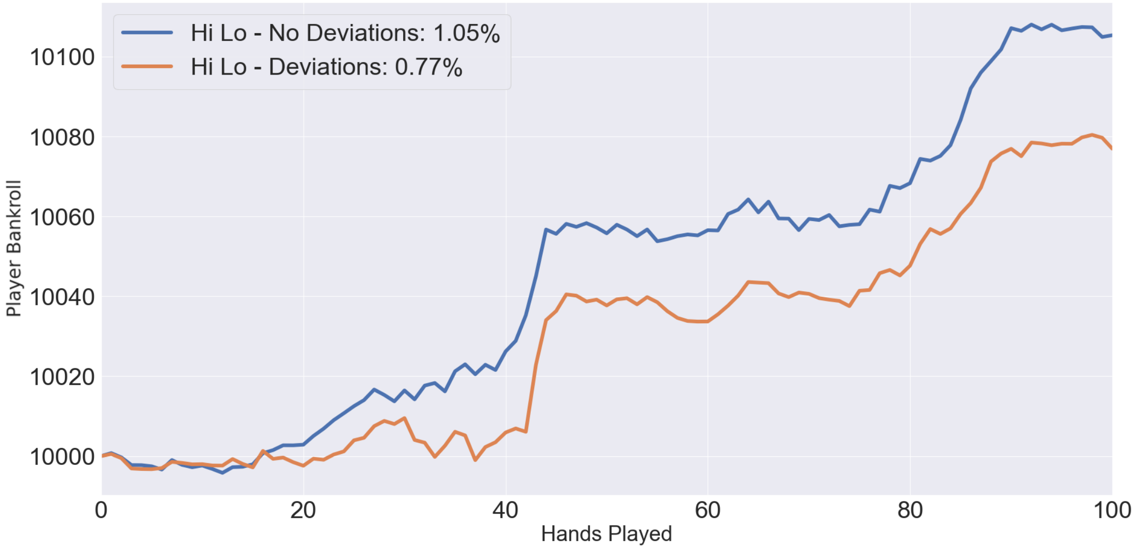

Figure 18: Basic Strategy with and without Playing Deviations using Hi Lo

	Average Profit	
Number of Decks	**Using Deviations**	**Without Deviations**
1 Deck	4.80%	4.19%
2 Deck	4.14%	3.18%
4 Deck	1.82%	1.48%
6 Deck	0.77%	1.05%
8 Deck	-0.10%	0.06

Table 12: Effect of using deviations for different number of decks

It can indeed be seen that a smaller number of decks produces significantly better results when using basic strategy deviations. When playing with a single deck deviating from basic strategy gives the player an extra 0.51% of an edge, 2 decks give a 0.96% edge, 4 decks: 0.34%, 6 decks: -0.28%, 8 decks: -0.16%. It is therefore likely that basic strategy deviations are useful when playing a game with a small number of decks.

7.6. Combination of Factors Effect on Profit

It may be possible that there is some interaction between all of the variables that have so far been analysed, and that one might have an effect on another. In an attempt to find out if this is the case, many combinations of playing variables were experimented with, but no unusual results were found. Three examples have been included below as evidence. The results come from the results of all simulations. Pivot tables were used to demonstrate how each playing factor, for example, counting method, performed against another e.g. betting size.

	Betting Size				
Counting Method	**Running Count**	**True Count**	**True Count*2**	**True Count+2**	**True Count-2**
Hi-Lo	-1.39%	1.46%	2.76%	1.83%	0.78%
Hi-Opt 2	-0.98%	2.0%	3.13%	2.51%	1.65%
KISS 3	-0.82%	1.4%	2.61%	1.82%	0.67%
Revere RAPC	-3.45%	3.64%	**4.58**%	3.86%	3.21%
Wong Halves	-1.13%	1.42%	2.96%	2.09%	0.93%
ZEN	-1.28%	2.42%	3.42%	2.79%	2.09%

Table 13: Pivot Table Showing Counting Methods Against Bet Size

	Betting Size				
H17/S17	**Running Count**	**True Count**	**True Count*2**	**True Count+2**	**True Count-2**
H17	-1.65%	1.92%	2.95%	2.17%	1.4%
S17	-1.37%	2.19%	**3.54**%	2.8%	1.71%

Table 14: Pivot Table Showing Counting Methods Against Bet Size

In the first experiment, it can be seen that, as before, Revere RAPC was the best counting method, independent of the betting system. It can also be seen that every counting method performed best when using the $TrueCount \times 2$ and there are no outliers in the data.

From the second table, we see that the combination of game type (H17 or S17) and

	Number of Decks			
H17/S17	**1**	**2**	**3**	**4**
H17	4.32%	2.16%	-0.24%	-0.8%
S17	**4.73**%	2.63%	0.12%	-0.38

Table 15: Pivot Table Showing Average Profit for Game Type (H17/S17) Against Number of Decks

bet size once again does not effect on the average profit of the player. As seen previously, S17 outperformed H17 amongst all bet sizes, and $TrueCount \times 2$ was the best betting system in comparison with all other systems. Hence we see that using S17 in combination with $TrueCount \times 2$ yields the best results for the player at 3.54%.

The third table looks at the game type against the number of decks. Previously we found that the fewer decks that are used in the shoe the better it is for the player, and also that S17 is more favourable than H17. Here we notice the same trend, S17 produces higher average profits for the player than does H17, and the smaller the number of decks, the greater the advantage to the player. This is why S17 in combination with 1 deck produces 4.73% higher than other combinations.

8. Developing New Card Counting Methods

8.1. Method Based on Advantage After Card Removal

Looking at the paper "The Expected Value of an Advantage Player" (Jensen, 2014) we see the effects on the player's advantage after removing each card. Looking at the table below, the higher the number means the more of an advantage the player has after the given card is removed from the shoe. For example, removing 5 is best for the player's expected value and removing the Ace is worst.

Effect of Card Removal*

Card	Effect
2	0.3875%
3	0.4610%
4	0.6185%
5	0.8018%
6	0.4553%
7	0.2937%
8	-0.0137%
9	-0.1997%
10	-0.4932%
A	-0.5816%

Figure 19: Effect of Removing a Card From the Shoe on the Player's Advantage (Jensen, 2014)

We can use these percentages to develop what might be the best counting method by simply moving the decimal point one place to the right. Of course, this method would not be practical at all in a real-life playing scenario, but it is a valid experiment to confirm if the findings from (Jensen, 2014) are true. It is very likely that all other counting methods try to replicate these numbers in a more practical system with whole numbers and numbers that do not go as high as 8 or -5. This is what the method below will be trying to achieve.

Card Value	**A**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Increment	-5.8	3.8	4.6	6.1	8.0	4.6	2.9	-0.1	-2.0	-4.9

Table 16: New Card Counting Method From Expected Advantage After Card Removal

8.2. A More Practical Method

This method was designed to resemble the method above, while instead using practical numbers that a player would be able to keep track of. Where the number is more than 0 in the original method, it is replaced with 1, and where it is more than 5 it is replaced with 2. The same principle applies to the high cards, but using negative numbers. Cards valued 8 have an effect of removal of only -0.01 so this will be the only card which is not counted.

Card Value	**A**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Increment	-2	1	1	2	2	1	1	0	-1	-1

Table 17: A practical adaptation of the new card counting method

8.3. Analysis

For the first experiment, the same rules were used as when we compared all of the existing counting methods, as follows: S17, 6 deck game, 75% deck penetration, true count, basic strategy with no deviations. Revere RAPC is included in the graph as it was the best-performing method. The second method I developed is a Level 2 counting system, so Hi-Opt 2 was included for comparison. Figure 21 looks at the average player profit across all combinations of simulations in the form of a box plot.

As expected the new counting system based on the effects of card removal outperformed even the Revere count by a significant amount of 0.72%. The new more practical level 2 method managed to come close to Revere and even slightly outperformed Hi-Opt 2. Overall this initial experiment showed great success for the development of new methods. From the second experiment, we once again see a slight edge, but not as significant as in the first experiment. Using the first counting method on average it performed 0.04% better than the previous best (Revere) as well as having a bigger distribution, suggesting a higher risk. The second of the new methods achieved a 1.81% average profit. This was 0.15% better than Hi-Opt II, but was slightly worse than average for Zen count which was 1.89%, which is another Level 2 counting system.

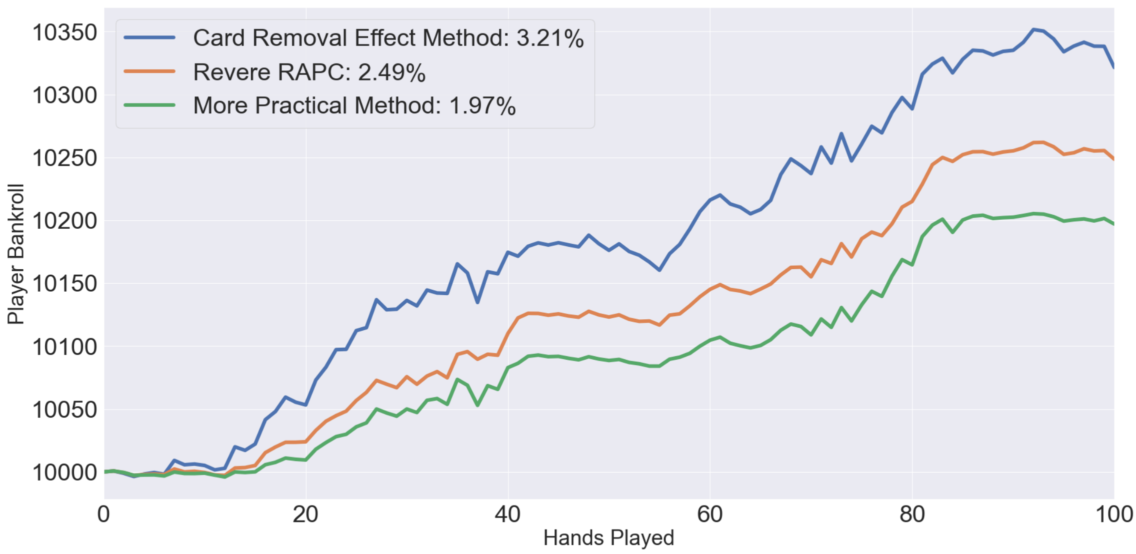

Figure 20: New Counting Methods Average Bankroll across 100 hands compared to Revere and Hi-Opt II

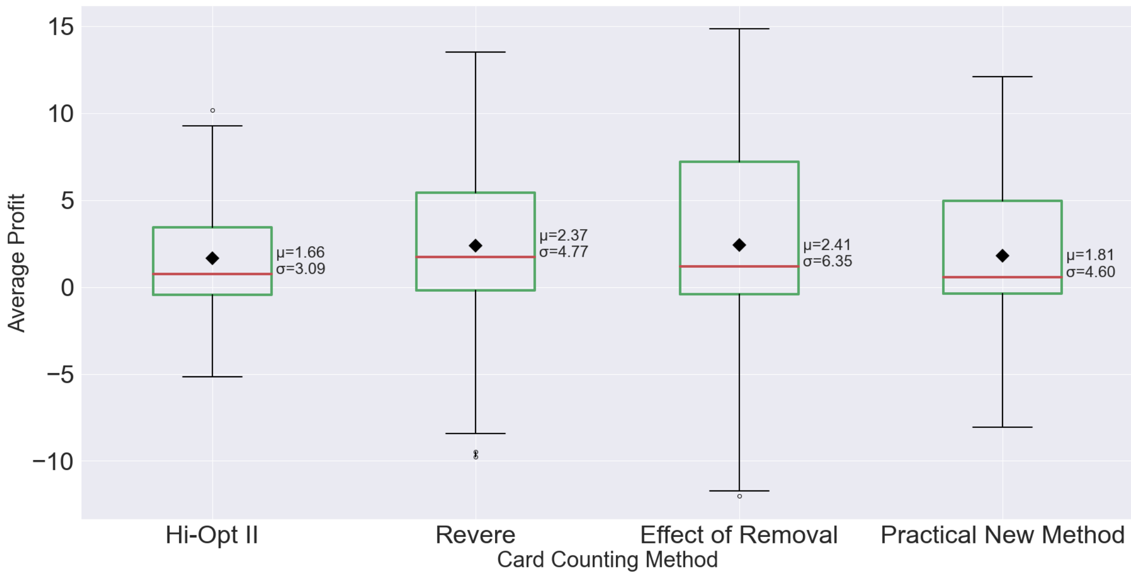

Figure 21: New Counting Methods Average Profit Across All Simulations Compared to Hi-Opt II and Revere

9. Conclusion

This project was designed to find out if blackjack is actually a profitable casino game and what would be the most profitable to play it. For this investigation, a lot of background knowledge of the game and the card counting strategies was required before implementing the simulations in order to understand the profitability of blackjack. Two new unique card counting methods were also developed through research and analysis of results.

Implementing the simulations created many challenges, as there were many factors that had to be taken into consideration. The most difficult part was implementing the splitting option in the game. Many small, but very important details such as deciding the bet, playing the computer's hands, updating running count, paying out 3:2 and many more had to be implemented and checked that they work correctly. Implementing Basic Strategy was not too difficult but a lot of manual checking in the debug mode was performed to ensure every single playing choice was being made correctly. After ensuring that no detail had been forgotten about, and all of the small issues were fixed, the results from pure basic strategy matched what was discovered in the prior research. In his research (Thorp, 1966) found that using a perfect basic strategy could reduce the casino's advantage to 0.5%, and in our results we achieved a 0.32% edge in favour of the casino, which validates that the results as being an accurate representation of expected profit. The addition of card counting methods was relatively easy with the only implications being the need to ensure that each card was counted correctly. After implementing Hi-Lo, the remaining methods presented no difficulties. All of the card counting methods proved effective as they improved the player's advantage significantly from the pure basic strategy. It also demonstrated a clear trend that the more complex the counting method, the better it performed to the player's advantage. The most profitable method, amongst those already developed by others, was the Revere RAPC with a 2.49% average profit in the most likely playing conditions in a casino. Different betting strategies were evaluated for varying numbers of decks in the shoe, for S17, H17 and different penetration percentages across every counting method. Following the evaluation of a large number of combinations and simulations, Revere RAPC was once again found to be the most profitable card-counting method (amongst methods already created by others) with an average player profit of 2.37% and a median of 4.77%. However, it also is the most risky amongst all such methods.

The best betting strategy was found to be the true count multiplied by 2, even though this increases player risk. S17 was confirmed to be a 0.10% more profitable game than H17. The deck penetration also had a clear trend, with the more cards being played resulting in more profit for the player. On average a 90% deck penetration was 0.82% profitable across all simulations. Bet spread also had a very big impact on profitability, with a 1-15 bet spread being 1.61% more effective for the Revere RAPC strategy. Basic Strategy deviations when counting cards gave a slight improvement when playing in a 1, 2 and 4 deck games but resulted in a detrimental effect when playing 6 and 8 deck games. Combining various different factors such as card counting method with betting strategies saw no significant changes in average profit for the player.

The final part of this investigation examined the possibility of developing new and even more advantageous card-counting strategies. The first strategy was slightly better than all existing counting methods with a 0.05% average profit edge, across all simulations, over the previous best, Revere RAPC. It had a much higher median profit, 1.58% higher than Revere. However, large numbers with a whole and a fractional part, which are not practical for a player to use. Nevertheless, it was a good starting point from which to develop a more practical method that outperformed other level-2 methods. It achieved a 1.81% edge across all simulations, which was 0.15% higher than Hi-Opt II, but Zen count still had a slightly better average at 1.89%.

An interesting further study would be how well Artificial Intelligence could play blackjack and if it can discover a more effective way of playing. A test where each card that is played is kept track of, including its value and suit, could be interesting as it would allow for the player to know exactly what is left in the shoe.

Overall the present study can confirm that blackjack is a profitable game, but only when playing in optimal conditions and using a card counting method. The present research shows that the best advantage to the player was produced by an impractical new method, using count numbers with decimal points, while Revere RAPC was the best existing counting method. However, it should be noted Revere is a level 4 method, which means it is very hard to use, and a more practical level 2 method such as Zen count, could be a more viable option for most players.

A. Code Examples

```python
class Player:
    def __init__(self):
        self.hand = Hand()
        self.bank = 10000
    def update_bank(self, amount):
        self.bank += amount
    def reset_hand(self):
        self.hand = Hand()
```

Listing 1: Player Class Code

```python
class Computer:
    def __init__(self):
        self.hand = Hand()
    def reset_hand(self):
        self.hand = Hand()
```

Listing 2: Computer Class Code

```python
def play_round(self, player):
    self.deal_cards(player)
    up = self.dealer.hand.draw(self.deck)
    self.update_running_count(up)
    player.hand.bet = self.decide_bet()
    h = self.play_basic_strategy(player.hand, up.value,
        self.deck)
    self.dealer_turn(self.computer, self.deck)

    for hand in h:
        self.bet_history.append(hand.bet)
        self.total_bets += hand.bet
        self.check_win(hand, player)
    player.reset_hand()
    self.computer.reset_hand()
```

Listing 3: Play Round Function Code

```
def play_game(self, player, rounds_limit):
    stack = [10000]
    player.bank = 10000
    while self.rounds <= rounds_limit:
        self.deck = Deck(self.num_decks)
        self.deck.shuffle()
        self.running_count = 0
        self.true_count = 0
        while len(self.deck.cards) > self.penetration():
            self.play_round(player)
            self.rounds += 1        stack.append(player.bank)
            self.true_count = round((self.running_count /
                self.deck.size()) * 2) / 2

    return player.bank, self.total_bets, self.player_wins,
        self.ties, self.computer_wins, self.bet_history, stack
```

Listing 4: Play Game Function Code

References

Blackjack Apprenticeship (2020a). Blackjack strategy chart. *https://www.blackjackapprenticeship.com/blackjack-strategy-charts/*.

Blackjack Apprenticeship (2020b). Blackjack strategy chart. *https://www.blackjackapprenticeship.com/blackjack-deviations/#:~:text=So%20what%20are%20blackjack%20deviations,pattern%20of%20wins%20or%20losses.*

Conway, D. G. and Koehler, G. J. (1998). Casino gambling on the internet: Scope, issues, and opportunities. In *Decision and Information Sciences*. Citeseer.

Crescent Schools (2019). The history of blackjack. *https://crescent.edu/post/the-history-of-blackjack*.

Hannum, R. C. (2003). A guide to casino mathematics. *UNLV Gaming Studies Research Center*.

Humble, L. (1980). *The World's Greatest Blackjack Book*. Main Street Books.

Jensen, K. (2014). The expected value of an advantage blackjack player.

Kendall, G. and Smith, C. (2003). The evolution of blackjack strategies. *Evolutionary computation*, 4:2474–2481.

Lucas, A. F. and Spilde, K. (2019). A deeper look into the relationship between house advantage and reel slot performance. *Cornell Hospitality Quarterly*, 60(3):270–279.

Michael Shackleford (2015). Blackjack house edge calculator. *https://wizardofodds.com/games/blackjack/calculator/*.

Norman Wattenberger (2021). Card counting. *https://www.qfit.com/card-counting.htm*.

Simon Young (2020). The house edge in blackjack: Everything you need to know. *https://edge.twinspires.com/casino-news/the-house-edge-in-blackjack-everything-you-need-to-know/#:~:text=The%20house%20edge%20is%20the,%2C%20you'll%20lose%20%242*.

Snyder, A. (1983). *Blackbelt in Blackjack*. Cardoza Publishing.

Tamburin, H. (2022). *Ultimate Guide to Blackjack*. 888casino.com.

Thorp, E. O. (1966). *Beat the Dealer: a winning strategy for the game of twenty one*, volume 310. Vintage.

Vaidyanathan, A. (2014). Monte carlo comparison of strategies for blackjack.

Vidámi, M., Szilágyi, L., and Iclanzan, D. (2020). Real valued card counting strategies for the game of blackjack. In *International Conference on Neural Information Processing*, pages 63–73. Springer.

Zimran, A., Klis, A., Fuster, A., and Rivelli, C. (2009). The game of blackjack and analysis of counting cards.