**Some clever TBD title here**

And also a clever subtitle, if appropriate

**Abstract**

We did lots of great things.

**Introduction**

No Limit Texas Hold’em (NLHE) is a game that utilizes an ordinary 52-card deck of cards. Like most card games, an essential facet to the game of NLHE is that the cards are randomized prior to each round of play, or hand, via shuffling. Theoretically, a perfect shuffling procedure would be one in which, after shuffling is performed, every possible ordering of the 52 cards has a 1/52! probability. It has been previously demonstrated that the specific protocol generally used by casino poker rooms to shuffle their decks of cards does not sufficiently randomize the cards, such that the resulting card distributions deviate from that of a perfect shuffling procedure to an extent that can be exploited by players to gain an advantage (Merz & Chi, 2022).

The components of the protocol utilized by casino poker rooms are:

1. Riffle shuffle: The deck is split into two portions of approximately equal size, and then the cards are interleaved back together.
2. Cut: The deck is split into two portions, and the portion originally on the bottom gets placed on top of the portion that was originally on top.
3. Strip shuffle: Also known as a box, this is similar to the aforementioned cut but instead of splitting into two portions, the deck is split into some greater number of portions (typically four), and the portions are placed back together in reverse order.

The protocol in question, referred to as the CS by Merz & Chi (2022), consists of the following, in this order: two riffle shuffles, one strip shuffle, one riffle shuffle, and one cut.

In this work, we investigate an additional shuffling technique that is sometimes utilized prior to the CS, known as the wash. To perform a wash, the dealer spreads out all 52 cards in an arc in front of them, and then rotates each hand in opposing circular motions. As the hands move, cards move with them, thus changing the cards’ relative positions until finally the cards are gathered back together. The primary question we aim to answer in this manuscript is: when a wash is performed prior to the CS, how much does this impact the exploitability of the CS by Texas Hold’em players as demonstrated by Merz & Chi (2022)?

Enroute to an answer to this question, we first develop a probabilistic model of the wash shuffle. Unlike many other shuffling techniques, the wash shuffle has not yet been well-studied or modeled. Our development of such a model is thus an additional contribution to the literature. We then utilize this model to simulate wash shuffles prior to the CS, and investigate the resulting impact on the distribution of cards as they are dealt for Texas Hold’em. We find that while the addition of a wash prior to the CS does result in an improved shuffling procedure by several metrics, it still falls short of a perfect shuffling procedure, specifically in terms of inflated relative frequencies of neighboring cards prior to the shuffle remaining as neighboring cards after the shuffle. Due to the nature of NLHE in which certain cards are dealt face-up, this is a crucial suboptimality of the shuffling procedure that can be exploited by a poker player.

**Literature Review**

The primary component of the CS is the riffle shuffle, which has been studied extensively in the literature. A probabilistic model for the riffle shuffle was first proposed by Gilbert (1955) and then independently by Reeds (1981), and has since been dubbed the Gilbert-Shannon-Reeds (GSR) model. Notably, it has been demonstrated that the GSR model produces deck orderings that accurately reflect what occurs when riffle shuffling is performed by a human (Diaconis, 1988). The GSR model prescribes the following to be performed:

1. The deck is split into two portions, with the dividing point determined according to a Binomial(n=52, p=0.5) distribution. Let A be the size of the top portion and B be the size of the bottom portion.
2. The portions are separated, and cards are dropped probabilistically from each portion into position to form the newly ordered deck. The probability that the first card to drop is from each portion is A/(A+B) and B/(A+B) for portions A and B respectively. After the first card drops, A or B changes its value accordingly, and thus the probabilities update for the next card. Cards continue to drop and the probabilities continue to update, until all cards from both portions have dropped into place.

Utilizing the GSR model, Aldous and Diaconis (1986) investigated how quickly, in terms of the number of iterations of a riffle shuffle, an unshuffled deck moves in distribution towards that of a perfectly shuffled deck (again, one in which every deck configuration has a probability of 1/52!) according to the total variation distance metric. Their investigation corroborated earlier work that recommends seven riffle shuffles to sufficiently randomize 52 cards (Aldous, 1983).

However, the CS only contains a total of three riffle shuffles, in addition to a strip shuffle and a cut, which clearly falls short of the recommendation of seven riffle shuffles. Also, to the best of our knowledge, the CS is universally used in casino poker rooms and in major poker tournaments across the world; some examples of professional dealers using the CS are in the 2017 and 2022 World Series of Poker (WSOP), in footage that is available freely online (PokerGO, 2017; PokerGO, 2022). Further evidence as to the ubiquity of the CS was mentioned in (Merz & Chi, 2022), in which it was also demonstrated that the CS leaves a deck in configurations that are exploitable by Texas Hold’em players.

One aspect that was left uninvestigated by Merz & Chi (2022) is the wash shuffle, in which the cards are spread out in front of the dealer and scrambled at random (see e.g. Black, 2020 for a video demonstration). A wash shuffle does not always occur prior to each hand of play; typically, a wash will be performed when a new box of cards is opened, or in some other circumstances such as when a dealer change occurs, or when an error has occurred during the CS process, e.g. as indicated in a 2013 WSOP dealer’s guide that is available on a public-facing website (WSOP, 2013). However, anecdotally, it appears that a wash may also occur before any hand at the dealer’s discretion, such as what can be observed in the 2018 European Poker Tour Monte Carlo Event (PokerStars, 2018).

Unlike the riffle shuffle, the wash shuffle has remained largely unstudied in literature. The only attempt to probabilistically model the wash shuffle that we have been able to find is in an unpublished manuscript (White, 2019). In this manuscript, three probabilistic models are proposed, in order of increasing complexity:

* Shuffling Process A: Cards begin by occupying spaces 1 through *n*, where *n* is the total number of cards. At each step, a random card is chosen, and either moves one step to the left, one step to the right, or stays in place, with probabilities of ¼, ¼ and ½ respectively. If a card moves to an occupied position, it is inserted into the pile of cards currently there at a random position. This is repeated for some specified number of steps, and then all piles are gathered together in order.
* Shuffling Process B: Cards again begin by occupying spaces 1 through *n*. At each step, every card moves a random number of spaces to the right according to independent geometric random variables, and then every card moves a random number of spaces to the left according to the same distribution. When cards enter an occupied position, they are merged with the cards already there by a GSR shuffle. This is repeated for some specified number of steps, and then all piles are gathered together in order.
* Shuffling Process C: This is a generalization of Shuffling Process A into multiple dimensions. For some number of dimensions *d*, cards begin by occupying arbitrary vertices in a *d-*dimensional grid. At each step, choose a random card. With probability ½, it remains at its current vertex; otherwise, with equal probability it will move to any neighboring vertex. If a card moves to an occupied position, it is merged with the cards already there at a random position. This is repeated for some specified number of steps, and then all piles are gathered together in order.

By the author’s own admission, each of these are quite crude models of an actual wash shuffle, and do not come close to being functionally realistic in terms of what actually happens during a wash shuffle. Their purpose was to be starting points for demonstrating particular mathematical properties such as theoretical mixing times and strong stationary times. Conversely, in the present work, our aim is to demonstrate the resulting impact of the wash shuffle on the play of Texas Hold’em via Monte Carlo simulation, and thus we are less restricted to simple models that are mathematically tractable and are only restricted by the complexity of simulation code that we wish to write. Thus, we will not directly consider any of these models described above, and instead formulate our own.

**Methodology**

**Simulating the Wash Shuffle**

*Initial setup*

Consider that a dealer’s wash movement initially spreads a single vector of cards into two overlapping circles. These will be the left and right circles, corresponding to which hand is washing them. We observe that, approximately, the left hand moves the first quarter of cards upward and then the third quarter of cards downward, thus completing the left circle. The right hand moves the fourth quarter of cards upward and then the second quarter of cards downward, thus completing the right circle.

Figure 1 Initial setup of wash shuffle

1A

1B

2A

2B

3A

3B

4A

4B

1B

1A

2A

2B

3B

3A

4A

4B

The above diagram presents the initial setup. The second diagram in Fig. 1 depicts how a wash can be thought of as a matrix of four columns. The first and third columns are cycled through by the left hand, and the second and fourth columns are cycled through by the right hand. We note that both structures depicted in Figure 1 will be referred to throughout this manuscript. If a vector is referenced, then it will either be the “left vector” (corresponding to cards moved by the left hand according to the blue arrows) or the “right vector” (corresponding to cards moved by the right hand according to the red arrows). Each completion of a circular movement by a single hand will be referred to as a single wash cycle.

*Truncated Poisson (tPoisson) Distribution*

To generate random variables within the possible range of a card deck, we utilize a truncated Poisson distribution. We appeal to this distribution because its right-skewed nature represents the processes in which we are using it, whereby smaller values should be more likely to occur (as described in more detail below). We truncate it to appropriate values as needed, such as from 1 to 52 when all positive values up to the full size of the deck are possible. In this case, the probability mass function becomes:

Although it would be possible to instead use a Binomial distribution, we would still have to truncate it since we do not want to include 0 as a possible value in any of our situations. Additionally, for any given desired mean value, the truncated Poisson distribution gives us a higher variance than a truncated Binomial distribution would, thus making higher values slightly more likely than they would be with a corresponding truncated Binomial distribution, which better matches what we aim to simulate in each of the instances described further below.

*Roll*

As one hand moves in its circular motion for the wash, cards move with it, in a manner depicted in the bottom panel of Figure 1. We prescribe this motion to essentially shift the vector by some number of steps; for example, if the number of steps is four, then the card starting in position 1 moves to position 5; the card starting in position 2 moves to position 6, and so on. Cards that started towards the back of the vector wrap around to move to the front of the vector. This motion continues until the wash cycles are completed. Based on (Black, 2020), we assume that approximately four wash cycles are completed by each hand (for a total of eight wash cycles altogether). The number of steps by which a single wash cycle moves its cards is randomly generated according to a tPoisson distribution, with a minimum of one card and a maximum set by the number of cards in the vector of interest (left or right). That is, for both the left-hand and right-hand cycles, the number of elements by which cards shift is where *vec\_len* is the number of cards in the vector in question.

*Slice*

Within each wash cycle, some cards will move from one vector to the other, due to the movement of the dealer’s hands. We call this movement a “slice.” We also distinguish between what we call an “inner slice” and an “outer slice,” whereby an inner slice refers to cards moving from one inner column to the other (i.e. column 2 to column 3 or vice-versa depending on whether it is a left-hand or right-hand wash cycle), and an outer slice refers to cards moving from an outer column to an inner column (i.e. column 1 to column 2 during a left-hand wash cycle, or column 4 to column 3 during a right-hand wash).

We assume that an inner slice will occur more frequently, due to the nature of the movement of cards during any given wash cycle. Specifically, we generate a Bernoulli random variable dictating that an inner slice will occur 70% of the time, with an outer slice occurring the remaining 30% of the time. Then,

there is a nonzero probability of a single card or group of cards slicing from one column to an adjacent one. Because both hand motions overlap each other at two points, and the innermost columns move opposite to one another. Inner slicing happens in greater frequency between columns two and three which are the inner columns, and outer slicing happens either between columns one and two or columns three and four. The assignment of donor and receiver of a cards slice occurs when the function is called.

Suppose there exists a donor column and a receiver column. When the wash parent function calls the slice function, we generate a slice size of with a maximum value equal to donor length minus two. The maximum value is set as such because, although the slice amount is random, it is always centered in the donor column. We do this because it retains randomness inherent in the wash while avoiding indexing errors that would otherwise occur. The slice becomes the number of neighboring cards centered in the donor column. The slice is always a neighboring grouping of cards because, as seen in the YouTube video (Black, 2020), most cards tend to stick in small groups despite being shuffled around.

The corresponding cards are removed from the donor column and inserted in the same index positions in the receiver column. The receiver column’s previous cards of those indices are then shifted to the end of the column.

*Gathering cards*

Once approximately four cycles have been performed by each hand, the cards are then gathered back into a single deck. While cards remain in at least one column, we generate *N*, *I*, and *L* variables such that with a mean of six, *I* randomly picks a remaining column, and *L* is the length of the chosen column. From column *I*, we move the *N* cards at the bottom of the column and place them at the top of the deck. If then we take the remaining cards in the column instead. *N*, *I*, and *L*are generated after each time a grouping of cards is moved into the deck. This process is completed once all columns are empty and the deck is of length 52.

Below is pseudocode for two of the primary functions that perform our simulations, greatly simplified from the actual code. Not included are *roll* and *slice* functions; as described above, the *roll* function simply shifts the position of the cards according to how a dealer’s hand motion, and the *slice* function essentially mimics the motion of a dealer’s hand that will move a cluster of cards together out of its current position in the overall sequence of cards, and into another.

***Algorithm 1:*** *wash*

Input**:** **deck**, cycles

Output**:** shuffled **deck**

Construct **left**: reversed first quarter of **deck** and third quarter of **deck**

Construct **right:** reversed fourth quarter of **deck** and second quarter of **deck**

FOR cycles:

*roll* **left** by TPois(λ=3)

*slice* from **left** into **right**

*roll* **right** by TPois(λ=3)

*slice* from **right** into **left**

*gather* on **deck**

RETURN **deck**

**Algorithm 2:** *gather*

Input: **quarter\_1, quarter\_2, quarter\_3, quarter\_4**

Output: final **deck**

**deck** = empty deck

WHILE length(**deck**) < 52:

i = quarter index, chosen at random

k = Binom(n = length(**quarter\_i**), p=0.5)

pick k cards from the bottom of ith quarter

add cards to top of deck

END WHILE

RETURN **deck**

**A single wash shuffle and a Monte Carlo simulation**

The process for a singular wash shuffle follows. The deck of cards is initialized from one to 52. Then the number of cycles is set to four, for each hand, so a total of eight. For each cycle, the cards shift by a truncated Poisson random variable. Then a 30-70 choice is made between performing an outer or inner shuffle. Note that an outer shuffle on the left hand will switch between columns one and two while a shuffle on the right hand will switch between columns three and four. If the left cycle was previously executed, then the right cycle goes next. To avoid programming errors, such as a column becoming completely empty, we specify that a vector must have at least five cards at all times. If this rule is violated, then the slice function is called deterministically where the longer column becomes a donor to the shorter column. After all cycles are completed, the columns of cards are gathered into a deck.

This process is ultimately fed into a Monte Carlo simulation where, for 100,000 repetitions, a new deck is initialized, and the wash shuffle is executed as described. Throughout the Monte Carlo sim, we take the newly shuffled deck and record the previous locations of cards 14, 15, 16, 18, and 20 because these will be the community cards in a game of NLHE.

A group of graphs showing different locations

Description automatically generated

Figure 2. 100,000 iterations

From Figure 2 we notice there is a consistent pattern of each previous position being right skewed. From the plots, we speculate the wash shuffle on its own tends to keep cards in a general vicinity of their previous position. However, a wash shuffle would never be performed without some additional shuffling technique. Therefore, any advantage from this alone would not be sufficient for a NHLE player.

We also want to know how many neighboring cards remained neighbors post wash shuffle. As shown in Figure 3, the wash shuffle has a high predisposition to keep neighboring cards together. As seen in fig. 3 left side, for every wash shuffle there are usually 14 cards (over a quarter of the deck!) that retained one of their neighbors prior to the wash shuffle. Next, we want to see how a wash and casino shuffle performs compared to only a casino shuffle (CS).

A graph of a number of people

Description automatically generated

Figure 3. 100,000 iterations

**Results**

**Simulating the Wash Shuffle and Casino Shuffle**

Just as before, we run 100,000 repetitions of a Monte Carlo, but this time we perform a CS after each wash which we will refer to as WCS. We show the results shown by Merz & Chi, 2022 but instead comparing the standalone CS with a WCS. In Figure 4 shows a black dotted line that represent perfect randomness. The blue and red 95% confidence intervals represent the prior position of each card. Just as with Merz & Chi, the 95% CI for the CS rarely falls within what is expected to be perfect randomness. However, when a wash is performed first the previous position of just about any card is perfectly random.

A screenshot of a graph

Description automatically generated

Figure 4

**Card neighbors**

We previously saw that a standalone wash shuffle heavily biases neighboring cards to remain neighbors. This would potentially be a major advantage to players if a card shuffle is only ever a wash for a given game. However, here we examine the likelihood of cards remaining neighbors with WCS. Figure 5 represents Monte Carlo estimates of the number of cards separating previously neighboring cards. When a wash is performed prior to a CS, any potential advantage of remembering neighboring cards is reduced when the cards are close enough in a deck to be taken advantage of. Figure 6 zooms in on neighboring cards separated by six or less cards and includes the 7-riffle shuffle introduced by Merz & Chi, 2022. Although a WCS performs far better at randomizing a deck, it is still outperformed by a 7-riffle shuffle.

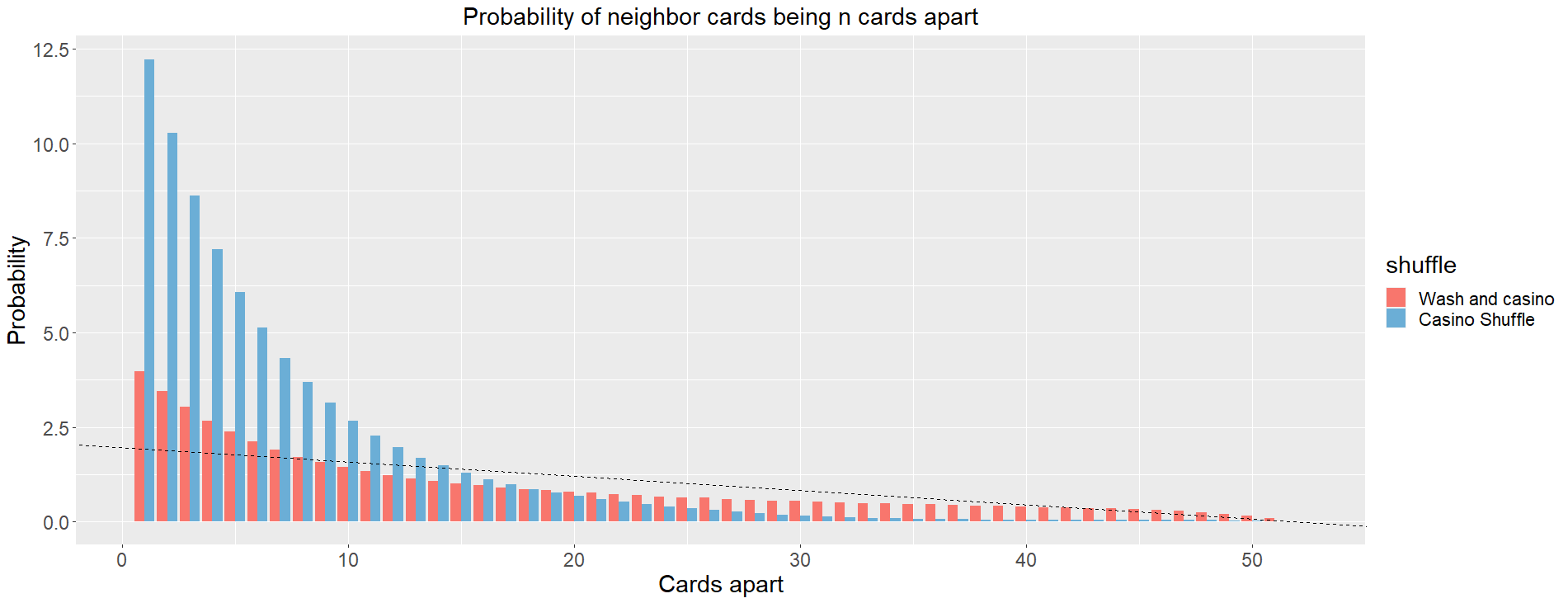


Figure 5

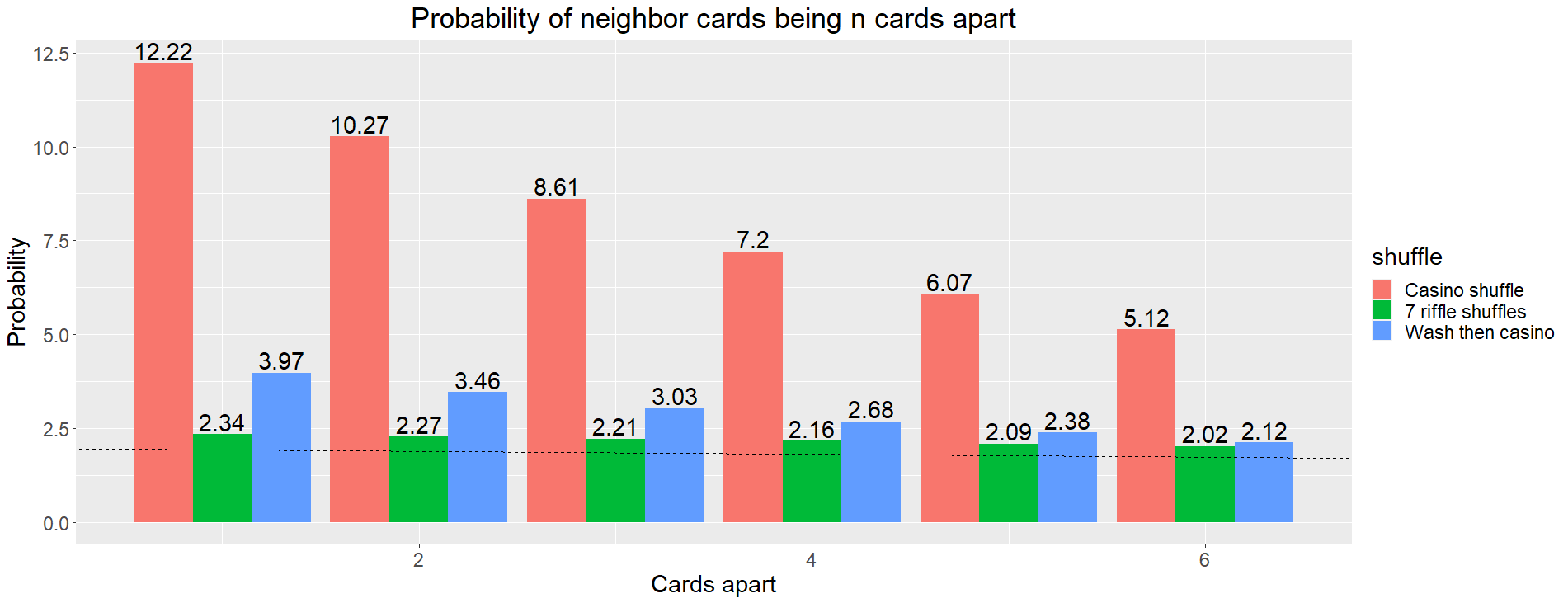
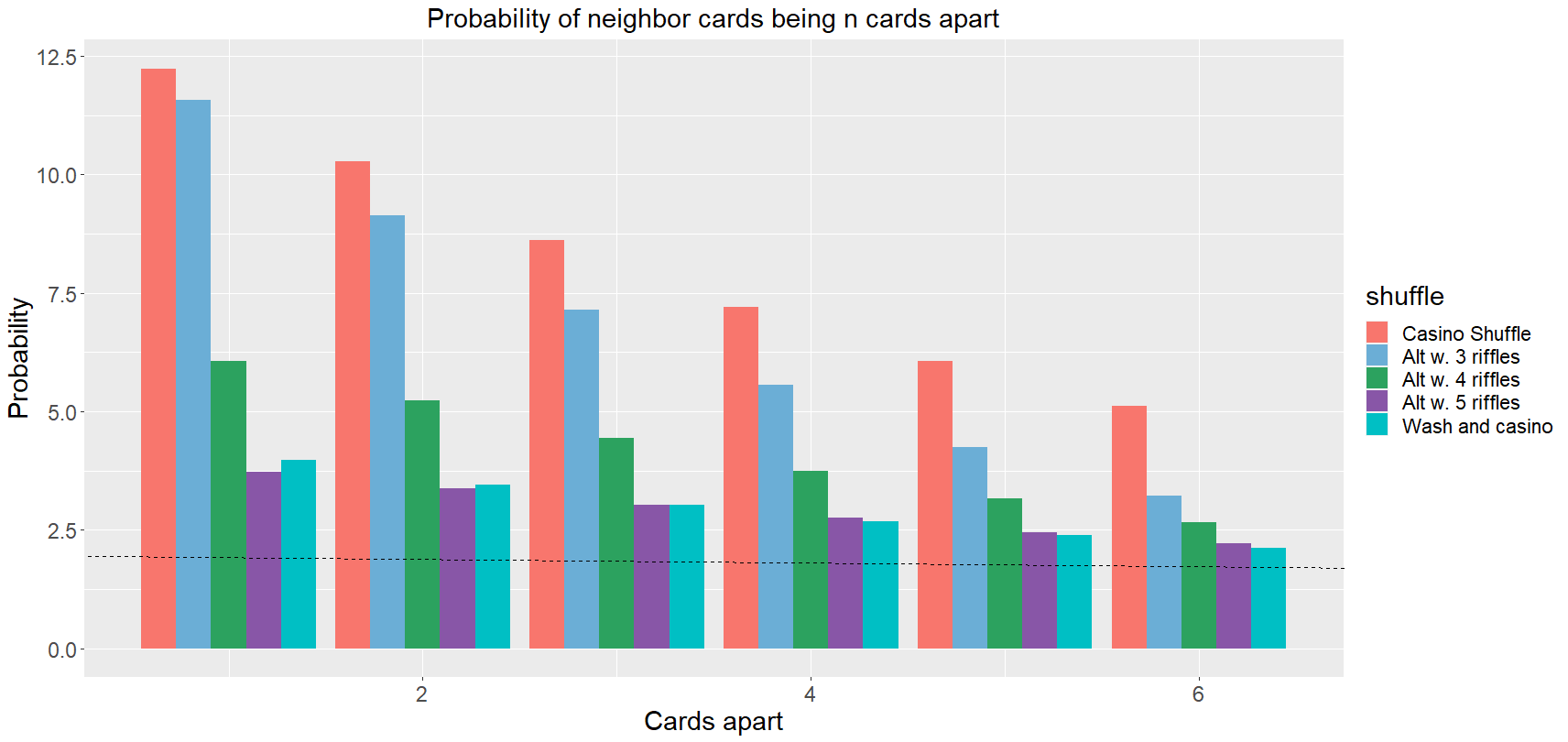


Figure 6



**References**

Aldous, D., & Diaconis, P. (1986). Shuffling cards and stopping times. *The American Mathematical Monthly,* *93*(5), 333. doi:10.2307/2323590

Black, J. (2020). How to shuffle cards professionally. *Jack Black Casino Dealer School.* Retrieved from <https://www.youtube.com/watch?v=svi7Jry-KQI&t=344s>

Gilbert, E. (1955). Theory of shuffling. Technical memorandum, Bell Laboratories.

Merz, D. W., & Chi, P. B. (2022). Knowing When to Fold'em: A Monte Carlo Exploration of Card Shuffling and How Poker Players Can Gain an Advantage. UNLV Gaming Research & Review Journal, 26(1), 7.

PokerGO. (2017). 2017 WSOP Main Event Day 1. Retrieved from <https://www.youtube.com/watch?v=zdTlTqqtcIk&t=420s>

PokerGO. (2022). 2022 WSOP Main Event Day 6. Retrieved from https://www.youtube.com/watch?v=p6\_t9ns9rUw&t=2h4m9s

PokerStars. (2018). Main Event Day 2 - EPT Monte Carlo 2018. Retrieved from <https://www.youtube.com/watch?v=WuNZEccZOsg&t=1917s>

Reeds, J. (1981). Unpublished manuscript.

White, G. (2019). A variation of strong stationary times for random walks with partial symmetries. arXiv preprint arXiv:1910.02389.

WSOP. (2013). 2013 World Series of Poker official dealer guide. Retrieved from <https://www.wsop.com/2013/2013_WSOP_Dealer_Guide.pdf>