Simulating the number of shuffles to randomize a deck of cards with R

Michael Rogers

Key Words: card shuffling, rising sequences, randomization

Abstract:

The riffle shuffle is the most common shuffle used in card games to randomize a deck of cards. The riffle shuffle can be simulated programmatically using probability models to determine the riffle shuffle's cut location and the mixing of the deck. Furthermore, the randomization of a deck of cards can be determined by comparing the number of rising sequences in the deck to the distribution of rising sequences for decks that are known to be in a random state. When a large number of ordered decks are simulated to have varying degrees of shuffling, the overlap of the observed and expected number of rising sequences for random decks indicates that 8 riffle shuffles are required to sufficiently randomize a deck of cards and that further shuffling does not add significantly more benefit to the degree of randomization of the deck.

1-Introduction:

When playing a game of poker, hearts, or bridge, the deck of 52 playing cards becomes structured as each player groups and organizes their hands and then strategically discards playing cards. Because of this, players typically shuffle a deck of cards two to five times between each game in order to randomize the deck and remove any organization that the previous game had created in the deck. However, are most card players shuffling the deck enough to sufficiently randomize it? Allowing even a small amount of order in the deck can create huge benefits for opportunistic players. By simulating in R a series of riffle shuffles on an organized deck of cards, I will explore the effect on randomization of varying numbers of riffle shuffles and will strive to answer the question: how many shuffles are required in order to sufficiently randomize a deck of cards?

2-Measuring deck randomization:

The concept of rising sequences will be used to measure the degree of randomization of a deck of cards. A rising sequence is a maximal increasing sequential ordering of cards that appear in the deck. For example, the 9-card deck 123789456 contains two rising sequences: 123456 and 789. Each rising sequence can have other cards interspersed within and around the sequence, just as the cards 789 appear in the middle of the 123456 rising sequence.

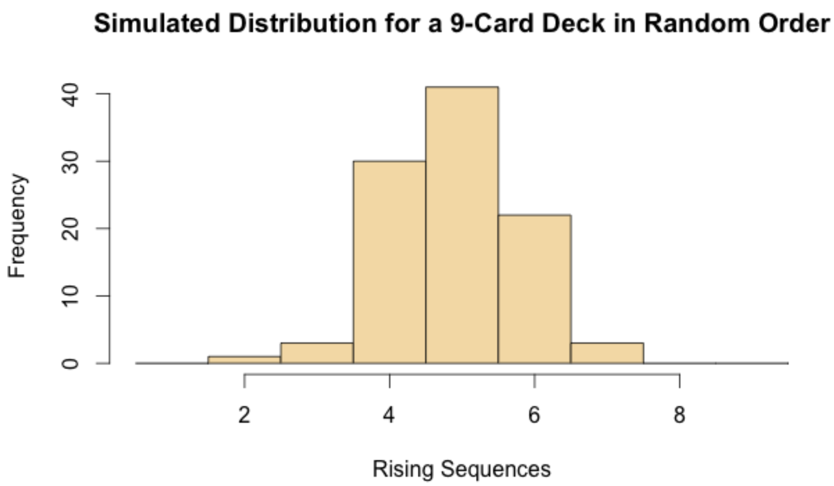
To understand how rising sequences apply to randomization, one can observe that the highly structured 9-card deck 123456789 has only one rising sequence, indicating that a low number of rising sequences does not likely correspond with a random deck. After one simulation of a riffle shuffle, the deck has the orientation 127348596 and now contains two riffle shuffles. This new orientation is more random than the original deck but still appears to be highly structure. It should be noted, though, that a high number of rising sequences also does not necessarily correspond with a random deck. For example, the deck 987654321 contains nine rising sequences—the maximum number possible for a 9-card deck. However, this deck is also highly structured as it displays the cards in perfect descending order.

In order to determine the expected number of rising sequences associated with a random deck, one can simulate a large number of random decks, count the rising sequences of each simulated deck, and find the distribution of these rising sequences. I accomplished this simulation in R by taking a random sample of size 9 without replacement from the sequence of integers from 1 to 9. The number of rising sequences is then determined by recording the location index of each card in the random deck and counting the number of times the location of a card is at a lower position than the preceding valued-card. For example, if the “5” card is at location 4 and the “6” card is at location 2, this would indicate that the “5” card is at the terminal of a rising sequence since the next valued card does not appear at a higher indexed location. The following R-program illustrates the simulation of 100 9-card random decks and displays the mean, standard deviation, and distribution of the number of rising sequences for these 100 random decks:

Program 2.1: Rising sequences for 100 simulated 9-card random decks

```
> m = 100; x = numeric(m)
> for(i in 1:m)
+ {
+   deck = sample(1:9, 9)      # Randomize deck of 9 cards
+   loc = match(1:9, deck)     # Index in 'deck' of each card
+   back = diff(loc) < 0       # Where rising seqs end (except last)
+   x[i] = sum(back) + 1       # Count rising sequences (including last)
+ }
> mean(x); sd(x);
[1] 4.89
[1] 0.9199802
> hist(x,breaks=(0:9+.5),col="wheat", xlab="Rising Sequences",
+      main="Simulated Distribution for a 9-Card Deck in Random Order")
```

Table 2.1: Distribution of rising sequences for 9-card random decks



The histogram from Program 2.1 indicates that the number of rising sequences in the random 9-card deck has approximately normal distribution with mean of about 4.89 and standard deviation of about 0.92.

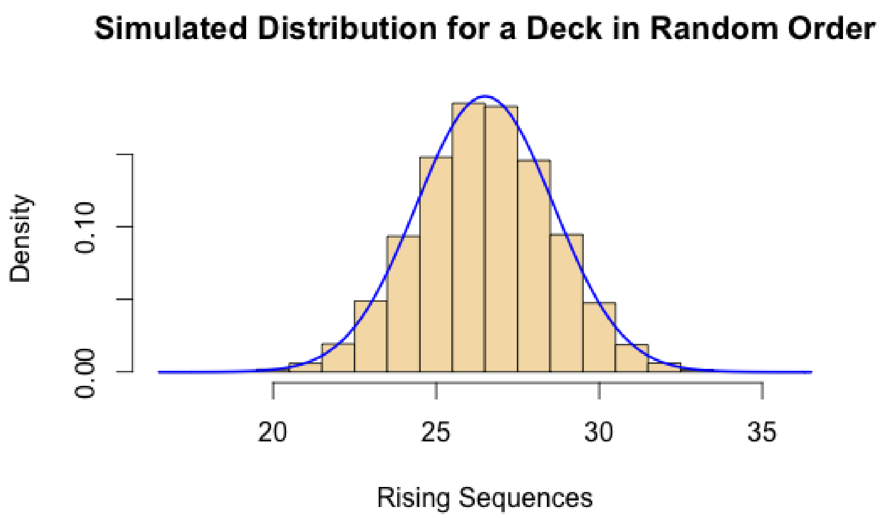
In order to determine the distribution of rising sequences in a randomized playing card deck, the simulation from the Program 2.1 is expanded from a 9 cards to 52 cards (where the integers 1 to 52 indicate the identities of each card in the deck's original structured orientation) and the number of simulations is increased to 100,000 to provide a more detailed and precise distribution.

Program 2.2: Rising sequences for 100,000 simulated 52-card random decks

```
> # Simulate 100000 random decks of cards and count the number of
rising sequences for each
> m = 100000; x = numeric(m)
> for(i in 1:m)
+   {
+     deck = sample(1:52, 52)    # Randomize deck of 52 cards
+     loc = match(1:52, deck)    # Index in 'deck' of each card
+     back = diff(loc) < 0       # Where rising seqs end (except last)
+     x[i] = sum(back) + 1       # Count rising sequences (incl last)
+   }
> mean(x); sd(x);
[1] 26.4899
[1] 2.099253
> # 99% probability interval
> mean(x)+c(qnorm(.005),qnorm(.995))*sd(x)
[1] 21.09143 31.92339
> # Proportion of rising sequences between 21 and 32
> mean(x>=21 & x<=32);
[1] 0.99601

> # Histogram of distribution of rising sequences
> cutp = seq(min(x)-1, max(x)) + .5
> hist(x, breaks=cutp, prob=T, col="wheat", xlab="Rising Sequences",
+      main="Simulated Distribution for a Deck in Random Order")
> curve(dnorm(x, 26.5, 2.1), add=T, col="blue", lwd=2)
```

Table 2.2: Distribution of rising sequences for 52-card random decks



The mean number of rising sequences of the simulations is about 26.5 with a standard deviation of about 2.1. Table 2.2 above shows the histogram of the number of rising sequences from the simulations plotted along with the normal curve with mean of

26.5 and standard deviation of 2.1. The table provides strong evidence that the rising sequences of a random deck closely follows this distribution. The 99% probability interval for the normal distribution (mean 26.5, S.D. 2.1) is 21.1 to 31.9, and the proportion of deck simulations with the number of rising sequences between 21 and 32 inclusively is 0.996. Because of the high majority of random decks have rising sequence counts from 21 to 32, this range provides a good criterion for whether a deck has reached a randomized state.

3-Simulating random shuffles:

The riffle shuffle is the most common shuffle used by card player to randomize a deck of cards. The riffle shuffle consists of cutting a deck into two piles of approximately equal size, and then placing cards in the new deck from each pile using roughly alternating order. In order simulate the riffle shuffle, one must first choose a cut location in the deck. The cut location (or number of cards in the first pile) can be simulated by taking a random sample from the binomial distribution with size of 50 and probability of 0.5 and adding 1. The simulated cut location has an expected value of 26, a 95% chance of being in the inclusive range of 19 to 33, and a 99% chance of being in the inclusive range from 17 to 35.

After a cut location is determined, the riffle shuffle is then simulated by placing cards from the tops of each pile into the new deck with probabilities corresponding to the relative size of each pile. For example, assume that the cut location is 23. This produces the first pile with 23 cards and the second with 29 cards. One of the piles is then chosen to produce the first card in the new deck with probabilities of 23/52 and 29/52 respectively. Suppose the first pile is picked—the first card from this pile is placed in the new deck. The source of the second card is then chosen with the new relative probabilities of the two piles, 22/51 and 29/51. This process is repeated until all 52 cards from the piles are placed into the new deck. To simulate more than one shuffle on a deck, the cut and shuffle procedure can be repeated using each new deck as an input to the next shuffle. Program 3.1 below demonstrates a single simulated riffle shuffle:

Program 3.1: Simulation of one riffle shuffle on an organized deck

```
> deck = 1:52 # fresh deck
> # Cut/Shuffle Deck 1 Time
> cut.nr = rbinom(1, 50, 1/2) + 1   #Location of cut in deck
> ix = sort(sample(1:52, cut.nr))   #Randomly select where the cards
                       before-the-cut will be placed in the new deck
> nd = numeric(52)            #Initialize temporary new deck variable
> nd[ix] = deck[1:cut.nr]     #Place the cards-before-the-cut in the
                              new deck
> nd[nd==0] = deck[(cut.nr+1):52]   #Place the cards-after-the-cut in
                                    the new deck
> deck = nd    #Redefine "deck" variable as the new deck
> deck         #Display new deck after 1 shuffle
 [1] 31  1  2  3  4  5 32 33  6 34  7 35  8 36  9 10 37 11 12 13 38 39
40 41 14 42 43 15
[29] 44 16 17 18 19 45 20 46 21 47 22 48 23 49 24 50 25 26 27 51 28 52
29 30
```

4-How many shuffles randomize a deck:

In Part 2, the criterion for sufficient deck randomization was constructed to be when the number of rising sequences in the deck is within the inclusive range of 21 to 32. In Part 3, a mechanism to simulate a series of imperfect riffle shuffles on an organized deck was assembled. In order to determine the ideal number of riffle shuffles to randomize a deck, the shuffling of a large number of ordered decks is simulated for varying numbers of riffle shuffles. The number of rising sequences is counted for each of these shuffled decks and is then used to determine if the deck is sufficiently randomized. For each degree of shuffling, the proportion of shuffled decks is compared to a level of significance. This study uses significance level of α=0.01, and so the smallest number of shuffles that produces randomization in at least 99% of the simulated decks is determined to be the required number of shuffles to properly randomize a deck of cards. Program 4.1 below uses 10,000 deck simulations for each degree of shuffling:

Program 4.1 Simulation of deck shuffling and proportion of deck randomization

```
> shuffles=c(1,2,3,4,5,6,7,8,9,10)
> num.mean.sd.prop=matrix(nrow=length(shuffles), ncol=4)
> for (s in 1:length(shuffles))
+ {
+ m = 10000; x = numeric(m); n = shuffles[s]
+ for (i in 1:m)
+ {
+   deck = 1:52 # fresh deck
+   # Cut/Shuffle Deck n Times
+   for (j in 1:n)
```

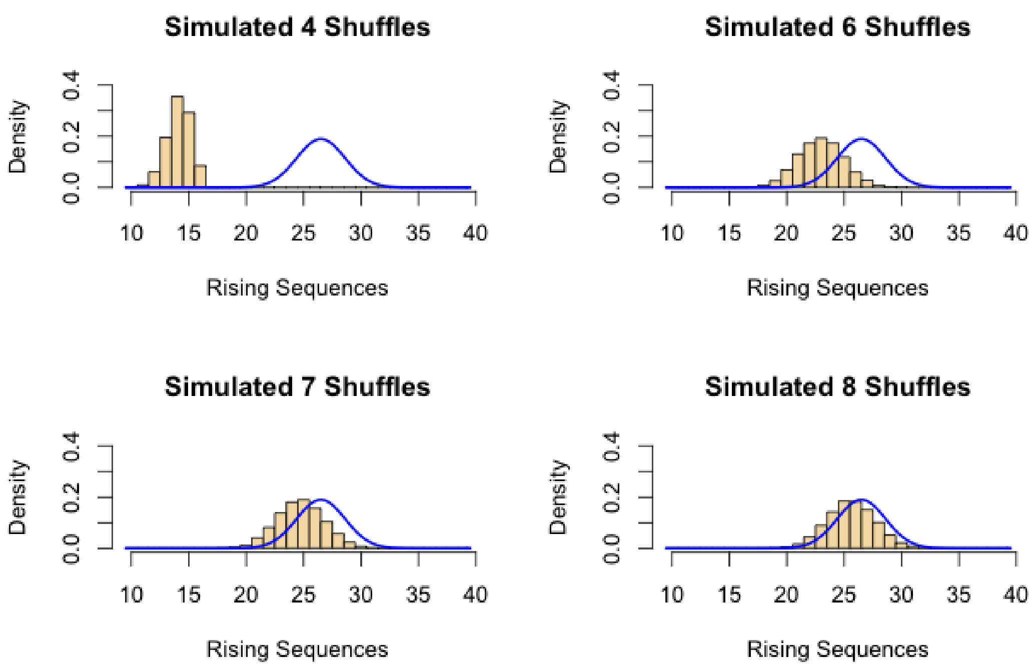
```
+   {
+     cut.nr = rbinom(1, 50, 1/2) + 1
+     ix = sort(sample(1:52, cut.nr))
+     nd = numeric(52)
+     nd[ix] = deck[1:cut.nr]
+     nd[nd==0] = deck[(cut.nr+1):52]
+     deck = nd
+   }
+   # Count Rising Sequences in Shuffled Deck
+   x[i] = sum(diff(match(1:52, deck)) < 0) + 1
+ }
+ mn = min(x); mx = max(x); cut = mn:(mx+1) - .5
+ num.mean.sd.prop[s,1]=shuffles[s];
+ num.mean.sd.prop[s,2]=mean(x); num.mean.sd.prop[s,3]=sd(x);
+ num.mean.sd.prop[s,4]=mean(x>=21 & x<=32);
+ }
> num.mean.sd.prop
```

Table 4.1: Proportion of randomized decks from 10,000 simulations vs. number shuffles

# Shuffles	Mean Rising Sequences	SD Rising Sequences	Proportion Randomized
1	2.000	0.000	0.000
2	4.000	0.000	0.000
3	7.952	0.217	0.000
4	14.102	1.079	0.000
5	19.632	1.775	0.313
6	22.963	2.008	0.894
7	24.747	2.097	0.981
8	25.606	2.084	0.994
9	26.042	2.118	0.995
10	26.276	2.100	0.997

For each number of shuffles of one through ten, Table 4.1 displays the mean number of rising sequences, the standard deviation of rising sequences, and the proportion of the 10,000 simulated decks in a randomized state based on the rising sequence count. The table demonstrates that as the number of shuffles increases, the distribution of rising sequences approaches the distribution of rising sequences seen from the completely randomized decks, normal with mean 26.5 and standard deviation of 2.1. Furthermore, as the number of shuffles increases, the proportion of randomized decks also increases toward 1. At 7 shuffles, 98% of the simulated decks have reached a random state; after 8 shuffles, more than 99% of the decks have reached a randomized state. Although the distribution of rising sequences for 8 shuffles (mean=25.6, s.d.=2.08) does not appear to exactly match the expected distribution for randomized decks (mean=26.5, s.d.=2.1), the shared coverage of the distributions is within the stated significance of 0.01 and therefore we conclude that 8 riffle shuffles produces a sufficiently randomized deck. Furthermore, any additional shuffles after the first 8 do not appear to significantly change the distribution of rising sequences, and therefore are redundant and unnecessary for further randomization. Table 4.2 clearly demonstrates this by showing the histograms of rising sequences for several degrees of shuffling plotted along with the expected distribution of randomized decks (normal with mean 26.5 and s.d. 2.1).

Table 4.2: Distribution of rising sequences for several degrees of shuffling



The results from this simulation indicate that the typical 2 to 5 shuffles most card players use are not sufficient to achieve true randomization of the deck. The players should be applying around 8 shuffles in order to randomize the deck and create equally fair playing conditions for all players in the game. Though some players may not want to apply the extra effort into shuffling their deck, the benefit of producing a new game with no influence from previous games should be compelling enough motivation to add a several more shuffles into your routine.

Bibliography:

Aldous, David and Diaconis, Perci. "Shuffling cards and stopping times", *The American Mathematical Monthly* 93(5), pgs. 333-348. 1986.

Bayer, Dave and Diaconis, Perci. "Trailing the dovetail shuffle to its lair", *The Annals of Applied Probability* 2(2), pgs.294-313. 1992.

Diaconis, Perci. "The cutoff phenomenon in finite Markov chains", *Proceedings of the National Academy of Sciences* 93(4), pgs.1659-1664. 1996.

Mann, Brad. "How many times should you shuffle a deck of cards?", *Topics in Contemporary Probability and its Applications*, pgs. 261-289. J. Laurie Snell, CRC Press, 1995.

Appendix:

Program 5.1: Random 9-card deck exploration

```
> #9-card deck exploration
> m = 100; x = numeric(m)
> for(i in 1:m)
+ {
+   deck = sample(1:9, 9)      # Randomize deck of 9 cards
+   loc = match(1:9, deck)     # Index in 'deck' of each card
+   back = diff(loc) < 0       # Ts where rising seqs end (except last)
+   x[i] = sum(back) + 1       # Count rising sequences (including last)
+ }
> mean(x); sd(x);
[1] 5.04
[1] 0.983911
> hist(x,breaks=(0:9+.5),col="wheat", xlab="Rising Sequences",
+      main="Simulated Distribution for a 9-Card Deck in Random Order")
```

Program 5.2: Random 52-card deck exploration

```
> # Simulate 100000 random decks of cards and count the number of
rising sequences for each
> m = 100000; x = numeric(m)
> for(i in 1:m)
+   {
+     deck = sample(1:52, 52)    # Randomize deck of 52 cards
+     loc = match(1:52, deck)    # Index in 'deck' of each card
+     back = diff(loc) < 0       # Where rising seqs end (except last)
+     x[i] = sum(back) + 1       # Count rising sequences (including
last)
+   }
> mean(x); sd(x);
[1] 26.50025
[1] 2.113264
> # percentage of random decks with 23 to 30 rising sequences
> mean(x)+c(qnorm(.025),qnorm(.975))*sd(x)
[1] 22.35833 30.64217
> mean(x)+c(qnorm(.005),qnorm(.995))*sd(x)
[1] 21.05684 31.94366
```

```
> mean(x>=21 & x<=32);
[1] 0.99622
> mean(x>=23 & x<=30);
[1] 0.9441
> # Histogram of distirbution of rising sequences for random decks
> cutp = seq(min(x)-1, max(x)) + .5
> hist(x, breaks=cutp, prob=T, col="wheat", xlab="Rising Sequences",
+      main="Simulated Distribution for a Deck in Random Order")
> curve(dnorm(x, 26.5, 2.1), add=T, col="blue", lwd=2)
```

Program 5.3: Simulation of riffle shuffles and randomization proportion

```
> ##Simulating riffle shuffles
> par(mfrow=c(2,2))
> shuffles=c(4,6,7,8)
> num.mean.sd.prop=matrix(nrow=length(shuffles), ncol=4)
> for (s in 1:length(shuffles))
+ {
+ m = 10000; x = numeric(m); n = shuffles[s]
+ for (i in 1:m)
+ {
+   deck = 1:52 # fresh deck
+   # Cut/Shuffle Deck n Times
+   for (j in 1:n)
+   {
+     cut.nr = rbinom(1, 50, 1/2) + 1   #Location of cut in deck
+     ix = sort(sample(1:52, cut.nr))   #Randomly select where the cards-
before-the-cut will be placed in the new deck
+     nd = numeric(52)        #Initialize temporary new deck variable
+     nd[ix] = deck[1:cut.nr]           #Place the cards-before-the-cut in
the new deck
+     nd[nd==0] = deck[(cut.nr+1):52]   #Place the cards-after-the-cut in
the new deck
+     deck = nd                         #Redefine "deck" variable with the
new deck's organization
+   }
+   # Count Rising Sequences in Shuffled Deck
+   x[i] = sum(diff(match(1:52, deck)) < 0) + 1
+ }
+ mn = min(x); mx = max(x); cut = mn:(mx+1) - .5
+ cut=10:40 - .5
+ hist(x, breaks=cut, col="wheat", prob=T, xlab="Rising
Sequences",ylim=c(0,.4),
+      main=paste("Simulated", n, "Shuffles"))
+ curve(dnorm(x, 26.5, 2.1), add=T, col="blue", lwd=2)
+ num.mean.sd.prop[s,1]=shuffles[s];
+ num.mean.sd.prop[s,2]=mean(x); num.mean.sd.prop[s,3]=sd(x);
+ num.mean.sd.prop[s,4]=mean(x>=21 & x<=32);
+ }
> par(mfrow=c(1,1))
> rbind(c("Shuffles","Mean Rising Seq","SD Rising Seq","% Random
Decks"),num.mean.sd.prop)
     [,1]       [,2]              [,3]               [,4]
[1,] "Shuffles" "Mean Rising Seq" "SD Rising Seq"    "% Random Decks"
[2,] "4"        "14.1239"         "1.06895418812297" "0"
[3,] "6"        "22.9939"         "2.01734225082798" "0.8912"
[4,] "7"        "24.7174"         "2.06348322898531" "0.9789"
[5,] "8"        "25.6154"         "2.09612075325379" "0.9936"
```