Final Project

Daniel Abravanel

11/30/2020

Introduction

This is a probability and simulation project handled in R, concerned with the intricacies of playing Blackjack 1 on 1 against a dealer. There are several instances in which we can add complexity, but we will ignore the following concerns by the magic of coding that is not exhibited in real life:

1. The deck will be shuffled perfectly for every play
2. There will not be other players whose cards we can see
3. The dealer shall show us one card until our turn is over, and then show their other card
4. The dealer is required to ask for a new card (henceforth called a "hit") if their cards sum up to 16 or less
5. The dealer is required to stop asking for a card (henceforth called a "stay") if their cards sum up to 17 or more
6. There will be no splitting of hands or any other fancy strategies

Of course, with the complexities of real life we will usually see some mild correlation between card series' in successive plays (non-thorough shuffling), several decks with which to mitigate the effects of said shuffling, some decision changes upon observing the cards other players receive (Bayesian probability) and so on and so forth. Assuming those interested in the paper are not interested in counting cards in the near future, we will keep this project vanilla and pursue the following topics:

1. What are the probabilities of winning, losing or tying if we were to enforce a cutoff for a certain value? In other words, what are our odds of winning if we were to base our **stay** ONLY on our own cards (i.e. not knowing what the dealer is holding)?
2. What are the probabilities of winning, losing or tying if we know the value of the cards we hold regardless of what the dealer has? In other words, probability of winning given our hand.
3. What are the probabilities of beating the dealer if we know both our own cards as well as the dealer's first card (which is the most realistic option)? In other words, probability of winning given our hand AND the dealer's first card.

My hypothesis for this simulation project is that it will lead us to the conventional answer that over all hands, the dealer has a slight advantage over the player. For the extremes in both directions (i.e. hitting too high or staying too low) the player's chances should be much lower, but following a specific strategy should allow the player to reach an almost equal footing as the dealer. Given the way the dealer plays, we could likely find that the player's strategy should ALWAYS depend on solely their own cards when playing against a dealer alone, and that the optimal strategy should exactly mimic the dealer. However, perhaps we will be able to find some answers in (3) that are illuminating to us, knowing fully that the dealer always hits below 17. In other words, if the dealer were to show a card that most likely will put them in the 13-16 range in sum, perhaps it is better to always stay and allow the dealer a probable bust.

######Imports

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(ggplot2)
```

####Starter Code

```r
# Compile the deck -- do not need J, Q or K because they are the same as 10 in BlackJack
Deck <- c(2,2,2,2,3,3,3,3,4,4,4,4,5,5,5,5,6,6,6,6,7,7,7,7,8,8,8,8,9,9,9,9,
          10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,"Ace","Ace","Ace","Ace")


# Function for Shuffling the Deck
shuffle_deck <- function(deck) {
  return (sample(deck))
}


# Function for dealing two cards to player from the Deck
deal_player <- function(deck) {
  player <- c(deck[1], deck[3])
  deck <- deck[-c(1,3)]

  return(player)
}

#Function for dealing two cards to the dealer from the Deck -- we are removing
#cards 1&2 from the deck because we already dealt and removed player's cards!

deal_dealer <- function(deck) {
  dealer <- c(deck[2], deck[4])
  deck <- deck[-c(1,2)]

  return(dealer)
}

#Function for pre-deciding a player's cards
decided_player <- function (deck, a, b) {
  player <- c(a,b)
  deck <- deck[-c(match(a, deck),match(b, deck))]

  return(player)
}

#Sort Aces -- this function will check whether each Ace should
#be 1 or 11 for purposes of hitting or not

ace_check <- function(hand) {
  if ("Ace" %in% hand) {
  current_sum <- 0
  for (a in hand) {
    if (is.numeric(a)) {
      current_sum <- current_sum + a
      }
    }
  if(current_sum < 11) {
    hand[match("Ace", hand)] <- 11
  } else hand[match("Ace", hand)] <- 1

  return (ace_check(hand))

  } else return (as.numeric(hand))
}

#a function to determine whether to play or not based on the
#current sum, insert ace_check at start

play <- function(hand, deck, cutoff) {
  hand <- ace_check(hand)
  if(sum(hand) > cutoff) {
    return (hand)
  } else {
    hand <- append(hand, deck[1])
    deck <- deck[-1]
  }
  play(hand, deck, cutoff)
}

#Run through of one game of BlackJack based on a fixed
#cutoff strategy for the player

Final <- function(deck, cutoff) {
  shuffled <- shuffle_deck(deck)
  player_hand <- deal_player(shuffled)
  dealer_hand <- deal_dealer(shuffled)
  player_hand <- play(player_hand, shuffled, cutoff)
  dealer_hand <- play(dealer_hand, shuffled, 16)

  return (if(sum(player_hand) > 21) {"L"}
             else if (sum(dealer_hand) > 21) {"W"}
             else if (sum(player_hand) < sum(dealer_hand)) {"L"}
             else if (sum(player_hand) > sum(dealer_hand) ) {"W"}
             else {"T"} )
}

#Run through a game of BlackJack knowing what your hand will be from the start
Final_Decided <- function(deck, a, b, cutoff) {
  shuffled <- shuffle_deck(deck)
  player_hand <- decided_player(shuffled, a, b)
  dealer_hand <- deal_dealer(shuffled)
  player_hand <- play(player_hand, shuffled, cutoff)
  dealer_hand <- play(dealer_hand, shuffled, 16)

  return (if(sum(player_hand) > 21) {"L"}
             else if (sum(dealer_hand) > 21) {"W"}
             else if (sum(player_hand) < sum(dealer_hand)) {"L"}
             else if (sum(player_hand) > sum(dealer_hand) ) {"W"}
             else {"T"} )

}
```

Question 1: Distribution of Wins from Holding Based on a Strict Cutoff Now that we have the above starter code for dealing a game of BlackJack, we can begin simulating MANY games to answer our key questions

In this first one, we will simulate what happens if we enforce a cutoff BEFORE WE EVEN START PLAYING

```
#Set the dimensions
i <- 1:10000 #10000 iterations per stay
j <- 13:20 #stays from 13 to 20

results <- matrix(nrow = length(i),ncol =  length(j), dimnames = list(i,j))


for (a in i) {
  for (b in j) {
    results[a,b - j[1] + 1] <- Final(Deck, b)
  }
}

distributions <- matrix(nrow = length(j), ncol = 3, dimnames = list(j, c("W","T","L")))


for(coli in 1:ncol(results)) {
  distributions[coli,] <- c(sum(results[,coli] == "W") / length(results[,coli]),
                            sum(results[,coli] == "T") / length(results[,coli]),
                            sum(results[,coli] == "L") / length(results[,coli]) )

}

distributions
```

```
##         W      T      L
## 13 0.3771 0.0631 0.5598
## 14 0.3860 0.0780 0.5360
## 15 0.3880 0.0926 0.5194
## 16 0.3748 0.0949 0.5303
## 17 0.3512 0.0745 0.5743
## 18 0.3158 0.0601 0.6241
## 19 0.2483 0.0469 0.7048
## 20 0.1094 0.0113 0.8793
```

As seen in the above results, we might not be well off from jumping into a game of BlackJack with one strict cutoff rule. It turns out that we cannot even break a 40% win rate with any cutoff, but surprisingly we get closest to success by staying on anything above 15 (Loss only 52%), whereas I hypothesized that above 16 would have provided the best chance of winning. Another surprising result is that staying only on 20 will yield wins approximately 11% of the time.

####Question 2: Distribution of Wins from Holding Based on a Given Initial Sum

This next situation is when we are dealt an initial sum and must make a decision given the initial hand (this is more of a Bayesian approach: chance of winning (A), given staying with the hand we were dealt (B)).

```
#Set the dimensions
i <- 1:10000 #10000 iterations per stay
j <- 13:20 #stays from 13 to 20

results2 <- matrix(nrow = length(i),ncol =  length(j), dimnames = list(i,j))


for (a in i) {
  for (b in j) {
    results2[a,b - j[1] + 1] <- Final_Decided(Deck, 10, b - 10, b - 1)
  }
}

distributions2 <- matrix(nrow = length(j), ncol = 3, dimnames = list(j, c("W","T","L")))


for(coli in 1:ncol(results2)) {
  distributions2[coli,] <- c(sum(results2[,coli] == "W") / length(results2[,coli]),
                             sum(results2[,coli] == "T") / length(results2[,coli]),
                             sum(results2[,coli] == "L") / length(results2[,coli]) )
}
distributions2
```

```
##         W      T      L
## 13 0.3005 0.0000 0.6995
## 14 0.2934 0.0000 0.7066
## 15 0.3053 0.0000 0.6947
## 16 0.2937 0.0000 0.7063
## 17 0.2844 0.1398 0.5758
## 18 0.4356 0.1448 0.4196
## 19 0.5814 0.1367 0.2819
## 20 0.7112 0.1710 0.1178
```

These second results are actually very interesting and intuitive. What we can see out of the 10,000 games we fixed to start with a certain sum, our chances of winning remain pretty much stagnant through 17. In other words, because the dealer has a strict cutoff of 17 and above, they will always hit if they are below it. Because of this, on the overall spectrum of all possible dealer hands, the dealer has an approximate 30% chance of busting! We can surmise from this that if we get any combination below 18, we still have a 30% chance of winning if we do not hit. This is wonderful news, especially in our simulation where we don't know see the dealer's cards before we make our decisions. Of course, though, our chances of winning rise with the sum we initially start with. On 17, we get an approximate 14% chance of also tying with the dealer, meaning that we still are most likely to lose the match! Even the dealer does not hit on 17, but our probability of winning while staying are improbable! The last callout is that even if we hold on to 20 (if you do not hold on to 20 you are either foolish, cheating, or a sorcerer), we only have an approximate 72% chance of winning, 17% chance of tying, and 11% chance of losing. This game is not for the weak!

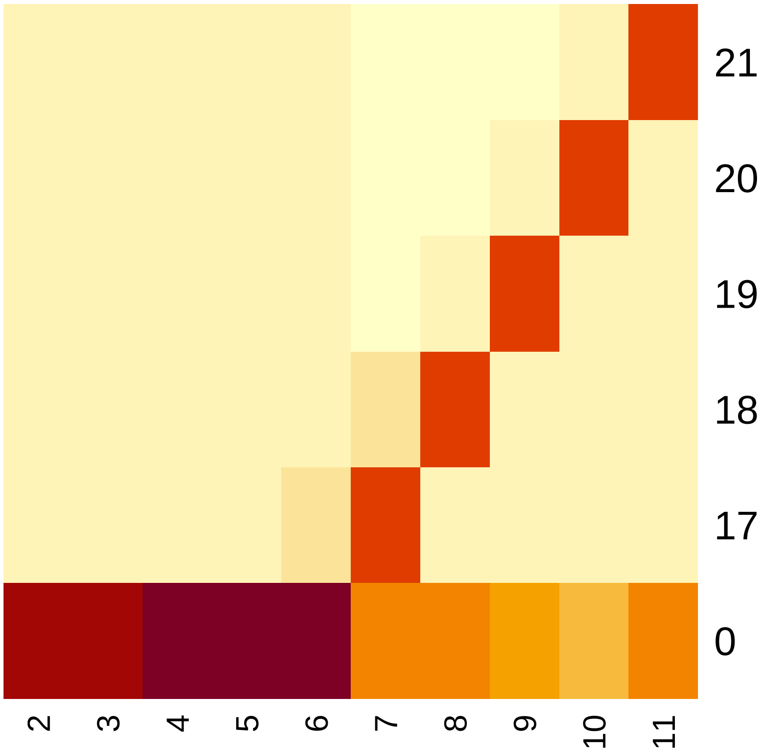
####Question 3: Distribution of Wins based on What the Dealer Holds Initially

This last simulation is most closely related to a Bayesian simulation, by which we want to predict our chances of winning given the first card we see in the dealer's hand. The use case of this would be that perhaps we would play more or less aggressively based on what the dealer's sum is most likely to be, given the first card which we can now see. If a player thinks a dealer has a much higher chance of busting, they will play more conservatively, or even stay regardless of what their sum is. We should break this out into two segments: 1) The general distributions the dealer will achieve given their first card, and 2) The specific likelihoods of beating the dealer based on a combination of our sum vs. their initial card

```
#Set the dimensions
i <- 1:10000 #10000 iterations
j <- c(2:11) #first card


deal_dealer_fixed <- function (deck, j) {
  hand <- (deal_dealer(deck))
  hand[1] <- j
  return (hand)
}

Final_Dealer_Dist <- function(deck,j) {
  shuffled <- shuffle_deck(deck)
  dealer_hand <- deal_dealer_fixed(shuffled, j)
  dealer_hand <- play(dealer_hand, shuffled, 16)
  dealer_first <- dealer_hand[1]
  solution <- if(sum(dealer_hand) > 21) {0} else {sum(dealer_hand)}

  return (c(solution, dealer_first))
}

results3 <- matrix(nrow = 0, ncol = 2)

for (a in i) {
  for (b in j) {
    results3 <- rbind(results3, Final_Dealer_Dist(Deck, b))
  }
}


#distribution of dealer results versus first card
(table2 <- table(results3[,1], results3[,2]) / 10000)
```

```
##
##            2      3      4      5      6      7      8      9     10     11
##   0  0.4098 0.4294 0.4463 0.4695 0.4631 0.2888 0.2750 0.2535 0.2193 0.2763
##   17 0.1215 0.1138 0.1173 0.1009 0.1340 0.3591 0.1134 0.1073 0.1068 0.0951
##   18 0.1228 0.1194 0.1109 0.1205 0.1012 0.1345 0.3573 0.1212 0.1157 0.0984
##   19 0.1215 0.1135 0.1133 0.1060 0.1074 0.0763 0.1257 0.3483 0.1108 0.0928
##   20 0.1110 0.1187 0.1055 0.1034 0.0963 0.0752 0.0621 0.1126 0.3386 0.1041
##   21 0.1134 0.1052 0.1067 0.0997 0.0980 0.0661 0.0665 0.0571 0.1088 0.3333
```

```
heatmap(table2, Rowv = NA, Colv = NA, scale = "none")
```



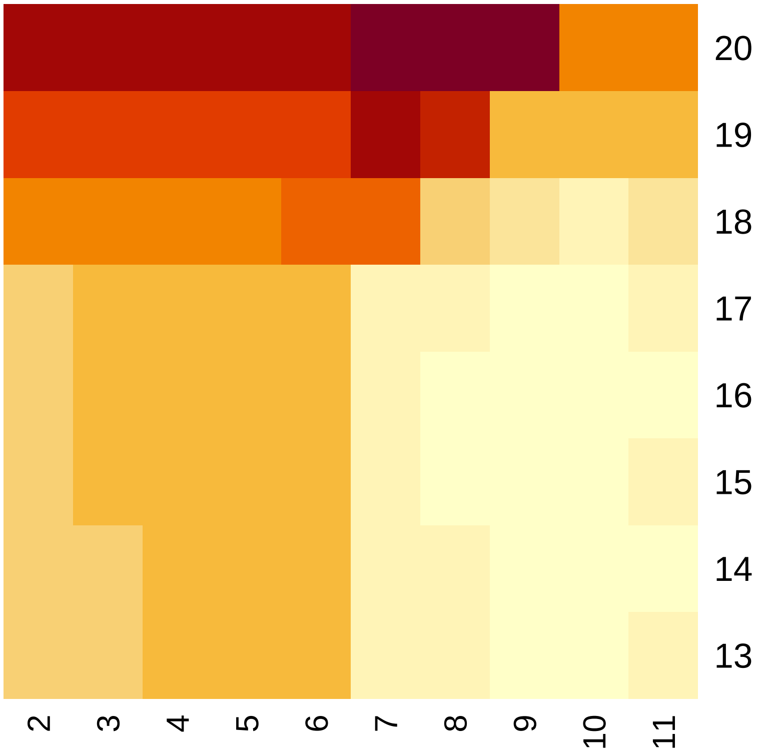
The above results are very illuminating. We saw earlier that on an overall basis, we can expect the dealer to bust approximately 30% of the time. As we can see from the table and associated heatmap above, this 30% (the '0' result) is split evenly across two groups: below 7 and above 7, with 7 representing the 30% in the middle. Hence, we can see that seeing the dealer's first card as 2-6, we are between 40-48% likely to watch the dealer bust that round, especially 5 and 6. Therefore, it could be a good strategy to hold and play safe when either of those cards is on the table. Thereafter, we are MOST likely to see lower number of busts from 7 to 10 ("Ace" going slightly higher than 10), and the highest likelihood of final sum for each first number is the first number + 10 as the second card (which makes sense because there are 16 10's in the deck in total). Therefore, the higher the dealer's initial card is, the more aggressively the player should bet to try to beat the dealer.

Now we must find out what happens to our decision-making when we know (1) our own 2 cards, and (2) the dealer's first card:

```
#Set the dimensions
i <- 1:2000 #2000 iterations per box
j <- 13:20 #player initial sum
k <- c(2:11) #dealer first card

#Define Function through which to iterate where we know both
  #our hand
  #dealer's first card

Final_Final <- function(deck, a, b, cutoff, dealer_first) {
  shuffled <- shuffle_deck(deck)
  player_hand <- decided_player(shuffled, a, b)
  dealer_hand <- deal_dealer_fixed(shuffled, dealer_first)
  player_hand <- play(player_hand, shuffled, cutoff)
  dealer_hand <- play(dealer_hand, shuffled, 16)

  return (if(sum(player_hand) > 21) {0}
             else if (sum(dealer_hand) > 21) {1}
             else if (sum(player_hand) < sum(dealer_hand)) {0}
             else if (sum(player_hand) > sum(dealer_hand) ) {1}
             else {0} )

}


results4 <- matrix(0L, nrow = length(j), ncol = length(k), dimnames = list(j,k))

for (a in i) {
  for (b in 1:length(j)) {
    for (c in 1:length(k)) {

      results4[b,c] <- results4[b,c] + Final_Final(Deck, 10, j[b] - 10, j[b] - 1, k[c])

    }
  }
}

(results4 = results4 / 2000)
```

```
##         2      3      4      5      6      7      8      9     10     11
## 13 0.4010 0.4295 0.4495 0.4645 0.4655 0.2725 0.2790 0.2565 0.2165 0.2780
## 14 0.4210 0.4335 0.4645 0.4645 0.4480 0.2780 0.2745 0.2590 0.2230 0.2630
## 15 0.4185 0.4375 0.4515 0.4645 0.4620 0.2945 0.2695 0.2465 0.2230 0.2815
## 16 0.3940 0.4385 0.4550 0.4595 0.4640 0.2710 0.2700 0.2505 0.2305 0.2580
## 17 0.4085 0.4405 0.4500 0.4615 0.4815 0.2940 0.2725 0.2530 0.2315 0.2770
## 18 0.5475 0.5560 0.5705 0.5740 0.6055 0.6395 0.3985 0.3540 0.3080 0.3605
## 19 0.6710 0.6620 0.6710 0.6865 0.7050 0.7750 0.7485 0.4835 0.4350 0.4705
## 20 0.7775 0.7780 0.7850 0.8000 0.8130 0.8565 0.8700 0.8225 0.5500 0.5580
```

```
heatmap(results4, Rowv = NA, Colv = NA, scale = "none")
```



As we can see above, our heatmap shows us the results of how our initial cards match up to the dealer's first card. If we find ourselves on the lighter side of the heatmap (i.e. the dealer has a higher initial card while we have a lower initial sum, we should play more aggressively in order to compete, because we most likely will not win the game without hitting at least once). On the other hand, our best chance of winning comes when we hold a high sum (18-20) and the dealer's first card is a 7 or 8. However, if our sum is below 17 WHILE the dealer's first card is below 7, we should consider staying because the dealer will most likely bust.

Summary In this project we built out a BlackJack simulation from scratch, walked through several strategy combinations, and found some interesting insights. The experiments gradually ramped up from pure frequentist to frequentist-bayesian mix, as we began to simulate what the odds would be given new bits of information. While they were not modern bayesian tests since I did not utilize R packages such as rstanarm or the respective continuous/marginal distributions to update priors, etc., we still looked at the probability spread of winning given new bits of information about our hand and the dealer's hand. I believe this analysis will be a useful springboard for future, more complex BlackJack analyses in which one could analyze probabilities based on what other people at the table are holding.

Below are our findings from this project:

1. A pre-determined cutoff of 15 is better than 16 by approximately 1% (39% vs. 38%), but will only get the wins + ties to 48%
2. A dealer will bust 30% of the time if the player stays automatically
3. When dealt a 20, there is only a 72% chance of winning, 17% chance of tying, and 11% chance the dealer will get a Blackjack
4. Dealer has an almost 50% chance of busting if their first card is a 5 or 6, which would make it wise to play conservatively
5. The best chance of beating the dealer is when we stay on high (18-20) and the dealer holds 7 or 8, because their next card is very likely to be 7-10, which would make the dealer stay on the lower end or hit and most likely bust.

Lastly, our hypotheses at the start were mixed. We were correct that if we picked a hard cutoff rule for ourselves at the start of the match, we would probably not beat the dealer (in fact we were at a slight disadvantage because we go first), but it was interesting to find out that 15, not 16, was the optimal number with the seed in our simulation. In this sense, if we were to pursue this strategy, we would not mimic the dealer. Our second hypothesis – that we most likely will find that our play should depend solely on our own cards – was definitely incorrect. The dealer's hand, as shown in summary points (4) and (5), has a tremendous weighting on our Bayesian playing strategy. And the best chance of winning was actually when the dealer had an equal chance of busting as well as staying at the lower end of the winning cycle (when their first card is 7 or 8).