Roulette Simulation

Sriram Kannan

09-05-2021

Contents

Introduction

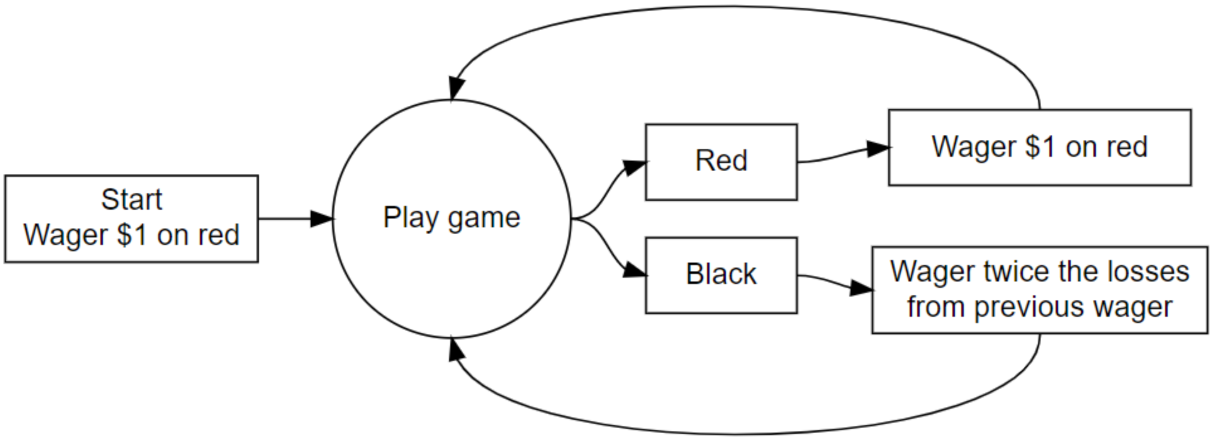
In this blog post, we explore Martingale's strategy and its viability through simulation and based on certain constraints applicable in a real world scenario(albeit with different parameter values depending on the setting).

Background

A brief blurb about Roulette in General :

A roulette table composed of 38 evenly sized pockets on a wheel. The pockets are colored red, black, or green. The pockets are also numbered. Roulette is a game of chance in which a pocket is randomly selected. Gamblers may wager on several aspects of the outcome. For example, one may place a wager that the randomly selected pocket will be red or odd numbered or will be a specific number. In our scenario, all one needs to know is that there are 38 pockets of which 2 are green, 18 are red, and 18 are black. The payout for a bet on black (or red) is $1 for each $1 wagered. This means that if a gambler bets $1 on black and the randomly selected pocket is black, then the gambler will get the original $1 wager and an additional $1 as winnings.

Here's a pictorial representation of Martingale's strategy. The Martingale strategy appears to always end in positive earnings, regardless of how unlucky a string of spins may be.



Constraints for the Simulation :

Stopping rule

A player will use the above strategy and play until

- The player has W(Winning Threshold for Stopping) dollars
- The player goes bankrupt
- The player completes L wagers (or plays)

In this scenario W is 300$ (Starting Budget + Winnings) abd L is 1000 plays.

Budget

The player starts with B(Starting Budget B - 200$) dollars. The player cannot wager more money than he/she has.

Maximum wager

Some casinos have a maximum bet. Call this parameter M. If the strategy directs the player to wager more than M dollars, then the player will only wager M dollars(100$ in this scenario).

Methods

```
#The below chunks of code show the modular processes involved in simulation of the roulette game based
```

```
#This chunk of code simulates a single spin of the roulette
single_spin <- function(){
  possible_outcomes <- c(rep("red",18), rep("black",18), rep("green",2))
  sample(possible_outcomes, 1)
}
```

```
#This chunk of code calculates the wager for the upcoming turn based on the previous wager as well as t
martingale_wager <- function(
  previous_wager
  , previous_outcome
  , max_wager
  , current_budget
```

```
){
  if(previous_outcome == "red") return(1)
  min(2*previous_wager, max_wager, current_budget)
}
```

```
#This chunk of code simulates one turn of the roulette outputting specific parameters to be used as par
one_play <- function(previous_ledger_entry, max_wager){
  # Create a copy of the input object that will become the output object
  out <- previous_ledger_entry
  out[1, "game_index"] <- previous_ledger_entry[1, "game_index"] + 1
  out[1, "starting_budget"] <- previous_ledger_entry[1, "ending_budget"]
  out[1, "wager"] <- martingale_wager(
    previous_wager = previous_ledger_entry[1, "wager"]
    , previous_outcome = previous_ledger_entry[1, "outcome"]
    , max_wager = max_wager
    , current_budget = out[1, "starting_budget"]
  )
  out[1, "outcome"] <- single_spin()
  out[1, "ending_budget"] <- out[1, "starting_budget"] +
    ifelse(out[1, "outcome"] == "red", +1, -1)*out[1, "wager"]
  return(out)
}
```

```
#This chunk of code establishes the stopping condition
stopping_rule <- function(
  ledger_entry
  , winning_threshold
){
  ending_budget <- ledger_entry[1, "ending_budget"]
  if(ending_budget <= 0) return(TRUE)
  if(ending_budget >= winning_threshold) return(TRUE)
  FALSE
}
```

```
#This chunk of code simulates the entire series of roulette spins until a certain stopping condition is
one_series <- function(
  max_games, starting_budget, winning_threshold, max_wager
){
  # Initialize ledger
  ledger <- data.frame(
      game_index = 0:max_games
    , starting_budget = NA_integer_
    , wager = NA_integer_
    , outcome = NA_character_
    , ending_budget = NA_integer_
  )
  ledger[1, "wager"] <- 1
  ledger[1, "outcome"] <- "red"
  ledger[1, "ending_budget"] <- starting_budget
  for(i in 2:nrow(ledger)){
    ledger[i,] <- one_play(ledger[i-1,], max_wager)
    if(stopping_rule(ledger[i,], winning_threshold)) break
  }
```

```
  # Return non-empty portion of ledger
  ledger[2:i, ]
}
```

Results

```
#The below code chunk gives the profit over a full series of roulette spins which is a property of inte
profit <- function(ledger){
  n <- nrow(ledger)
  profit <- ledger[n, "ending_budget"] - ledger[1, "starting_budget"]
  return(profit)
}
```

```
#All of the below function inputs are based on constraints described above
a <- one_series(1000,200,300,100)
a
```

```
##    game_index starting_budget wager outcome ending_budget
## 2           1             200     1     red           201
## 3           2             201     1     red           202
## 4           3             202     1     red           203
## 5           4             203     1     red           204
## 6           5             204     1   black           203
## 7           6             203     2   black           201
## 8           7             201     4     red           205
## 9           8             205     1     red           206
## 10          9             206     1     red           207
## 11         10             207     1   black           206
## 12         11             206     2   black           204
## 13         12             204     4   black           200
## 14         13             200     8   black           192
## 15         14             192    16     red           208
## 16         15             208     1     red           209
## 17         16             209     1     red           210
## 18         17             210     1   black           209
## 19         18             209     2   black           207
## 20         19             207     4     red           211
## 21         20             211     1     red           212
## 22         21             212     1   black           211
## 23         22             211     2   black           209
## 24         23             209     4     red           213
## 25         24             213     1     red           214
## 26         25             214     1     red           215
## 27         26             215     1     red           216
## 28         27             216     1     red           217
## 29         28             217     1   black           216
## 30         29             216     2     red           218
## 31         30             218     1   black           217
## 32         31             217     2   black           215
## 33         32             215     4   black           211
```

```
## 34          33                211     8     red              219
## 35          34                219     1   green              218
## 36          35                218     2   black              216
## 37          36                216     4     red              220
## 38          37                220     1     red              221
## 39          38                221     1     red              222
## 40          39                222     1     red              223
## 41          40                223     1     red              224
## 42          41                224     1     red              225
## 43          42                225     1     red              226
## 44          43                226     1     red              227
## 45          44                227     1     red              228
## 46          45                228     1   black              227
## 47          46                227     2     red              229
## 48          47                229     1   black              228
## 49          48                228     2   black              226
## 50          49                226     4   black              222
## 51          50                222     8     red              230
## 52          51                230     1   black              229
## 53          52                229     2   green              227
## 54          53                227     4   black              223
## 55          54                223     8     red              231
## 56          55                231     1   green              230
## 57          56                230     2   black              228
## 58          57                228     4   black              224
## 59          58                224     8   black              216
## 60          59                216    16   black              200
## 61          60                200    32   black              168
## 62          61                168    64   black              104
## 63          62                104   100   black                4
## 64          63                  4     4   black                0
```

```
profit(a)
```

```
## [1] -200
```

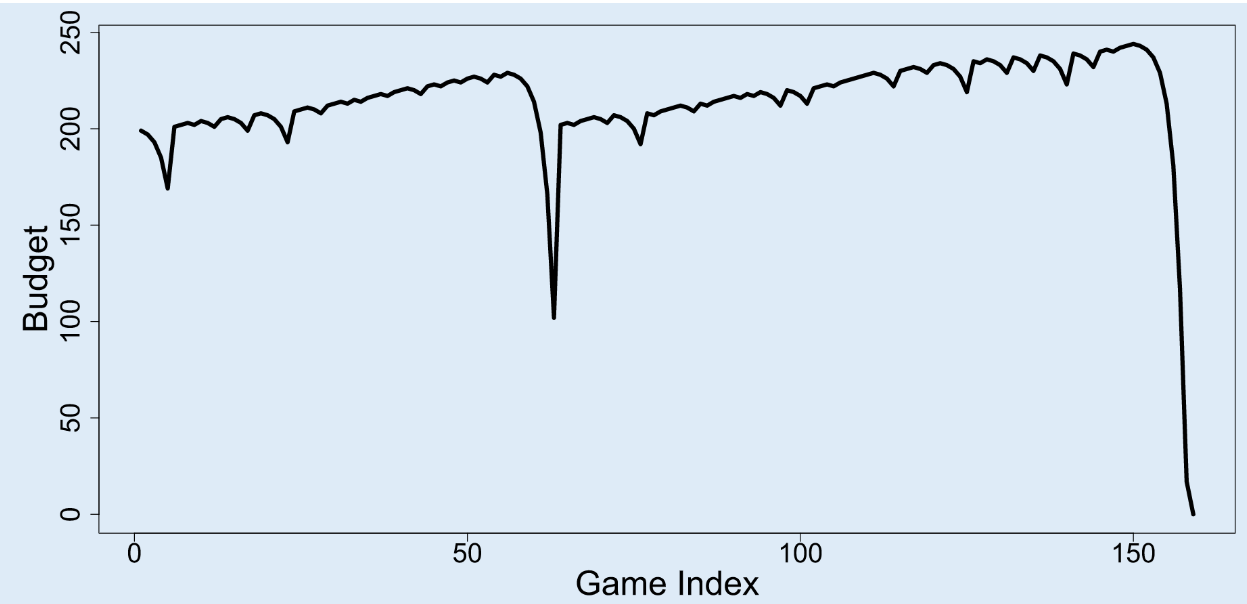
The results are not always positive in the sense that the chances of winning and losing seem to be random and the Martingale strategy **under these constraints is not** a guaranteed success. A specific observation is that a vast majority of profit results tend to be either -200 or 100. This is due to the either one of the stopping conditions being achieved - 1000 turns (extremely low chance of happening due to the small difference between starting amount and the winning threshold and the starting amount itself being low enough that 8-9 continuous losses would cause the player to lose the entire initial capital) or achieving the winning threshold or losing the initial capital/budget.

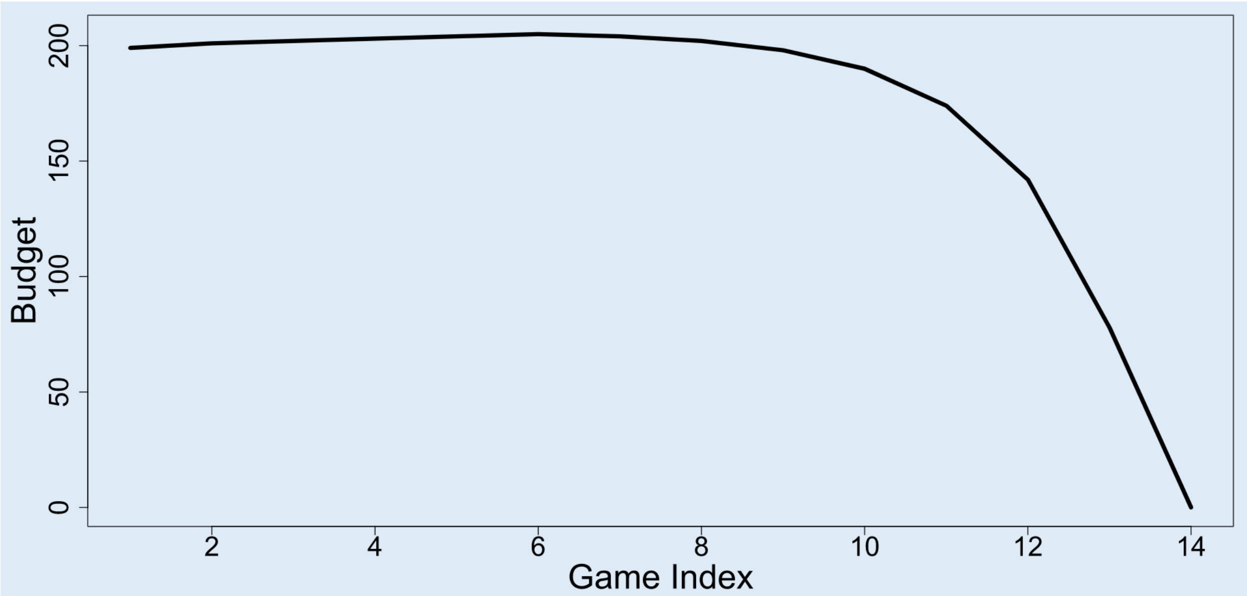
```
library(magrittr)

svg(filename = "pattern.svg", width=16, height =8.25)
par(cex.axis=2.0, cex.lab = 2.5, mar = c(8,8,2,2), bg = rgb(222, 235, 247, max = 255))
#set.seed(1)
#Constraints chosen based on above stipulations.
ledger <- one_series(1000,200,300,100)
plot(ledger[,c(1,5)], type = "l", lwd = 5, xlab = "Game Index", ylab = "Budget")
dev.off()
```

```
## pdf
##   2
```

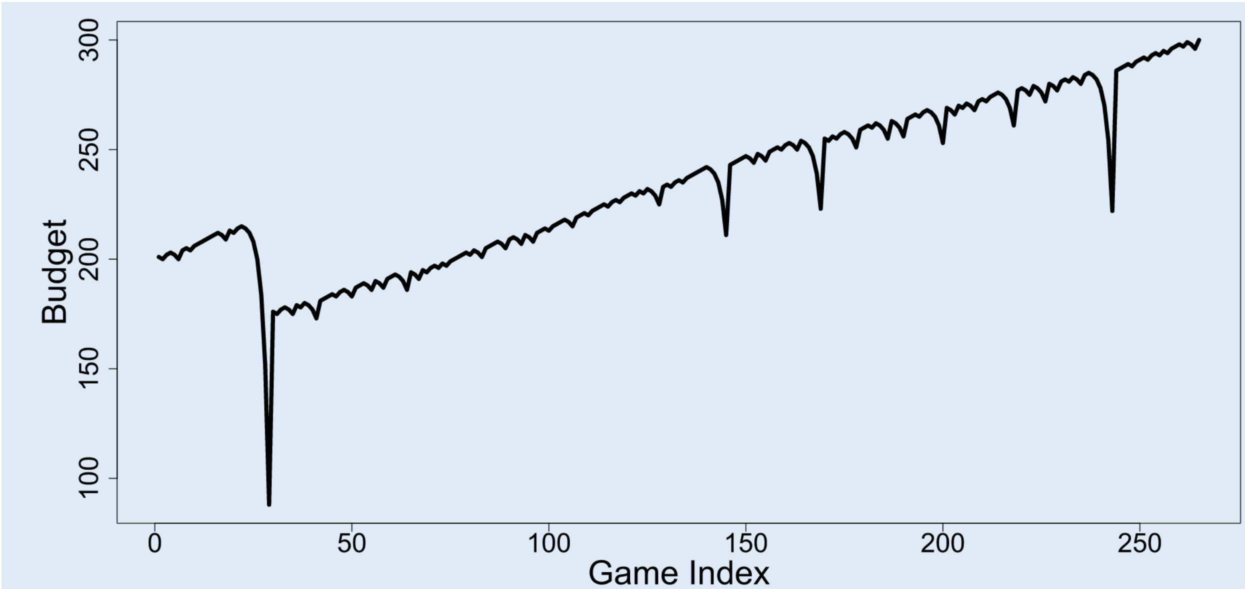
The below figures show both gamblers winning and losing over the series of wagers based on above constraints.

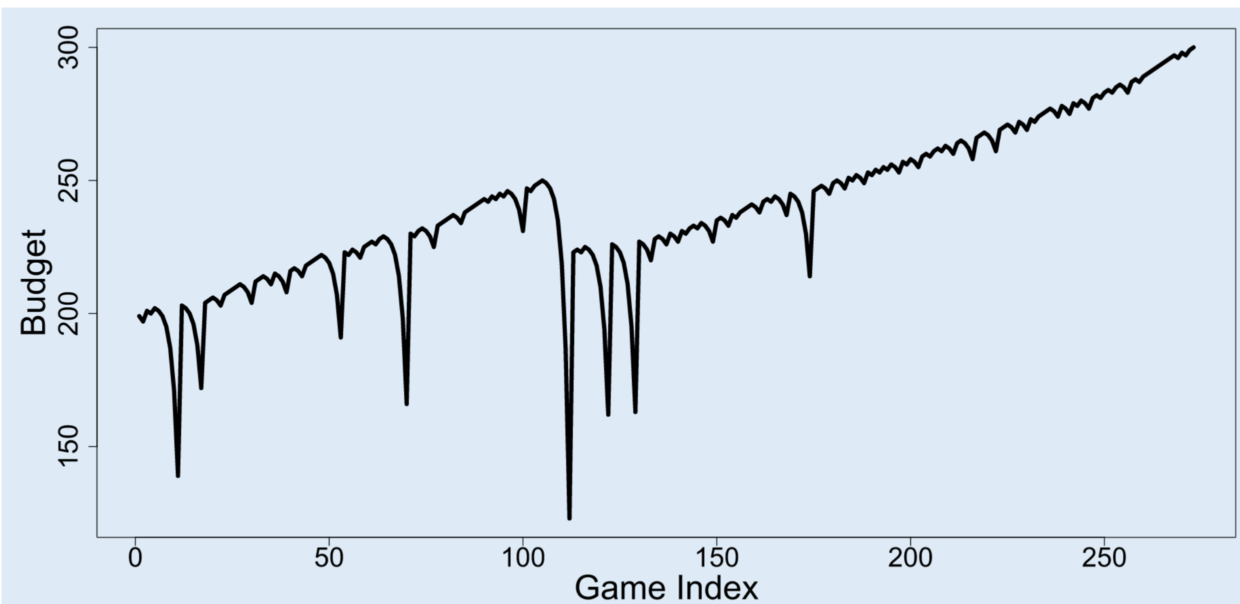
Lost Wager





Won Wager





Conclusion and Discussion

A massive limitation with this simulation is the chosen constraint value. It can be concluded that the Martingale's strategy cannot guarantee a win with the above constraints. But, modification of said constraints can increase the probability of positive results. It can also be effectively concluded that the Martingale's strategy **guarantees** positive net earnings when there are no constraints and that the gambler has full control as to how many turns he gets to play, has an infinite (or a very large capital to spend) and there isn't a maximum bet restriction. Modifying some of these constraints can shift the odds in the gambler's favor though.

For instance,

- The budget of the gambler is directly proportional to the chance of a positive outcome.
- The number of chances to play is also directly proportional to the chance of a positive outcome. (In our scenario this played a very minor role due to how the stopping conditions were set up.)

- The maximum bet amount is inversely (although not in any linear fashion) proportional to the chance of a positive outcome.

```
#This chunk of code returns positive results alone
PResult <- function(r, max_games, starting_budget, winning_threshold, max_wager)
{
rep <- c(replicate(r, profit(one_series(max_games, starting_budget, winning_threshold, max_wager))))
return (rep[rep>0])
}
```

```
#This chunk of code shows how changing parameters based on the discussion above affects the outcome of
PResult(15, 1000, 200, 300, 100)
```

```
## [1] 100 100 100 100 100 100 100
```

```
PResult(15, 1000, 200, 300, 2000)
```

```
## [1] 100 100 100 100 100 100 100 100
```

```
PResult(15, 1500, 1000, 2000, 2000)
```

```
## [1] 717 704 733
```

```
PResult(15, 1000, 25000, 30000, 50000)
```

```
##  [1] 461 465 479 467 444 488 479 444 493 443 468 489 462 490
```

There tends to be a shift towards outcomes being positive. Hence, Martingale's strategy while not a slam dunk in every scenario, can be edged towards a positive outcome based on conditions at play. However, it is still based of probability and not an absolute certainty. Even with conducive conditions, one could get unlucky. This can be observed in the above example in the 3rd case whwere it had large amounts of profits but the number of positive results were lower than the first case.