Strategy C: James Bond Strategy

Joseph Oluwasanya (Student no: 202306564)

2023-11-04

Introduction

Here we analyse the James Bond Roulette Strategy. Firstly, a short explanation:

Bets are spread across a range of numbers in this strategy. Given a total bet size D,

- 25% of the D is placecd on numbers 13 to 18 inclusive. This is a Line bet, which pays x5, so the profit for a win is
$$(0.25 * 5)D - 0.75D = 0.5D$$
.
- 70% of D is placed on numbers 19 to 36. The payout for this bet is equal to the bet size, so the profit is
$$0.7D - 0.3D = 0.4D$$
.
- The remaining 5% of D is placed on 0 as a Straight up bet, which pays x35. The profit is
$$(0.05 * 35)D - 0.95D = 0.8D$$
.

Let X denote the random variable for winnings from roulette using this strategy. The expected value for a single role in this strategy is $\frac{-D}{37}$, calculated as follows.

$$E(X) = \frac{6}{37} \times 0.5D + \frac{18}{37} \times 0.4D + \frac{1}{37} \times 0.8D - \frac{12}{37}D$$

Here we use $D = 10$, so the expected loss for a single roll is 0.27. We will check if the simulated rolls converge to this later in Question 2.

Implementation

```r
library(MASS)
set.seed(1)


roulette <- function(n, numbers=0:36){
  # n = number of rolls
  rolls <- sample(numbers, size=n, replace=TRUE)
```

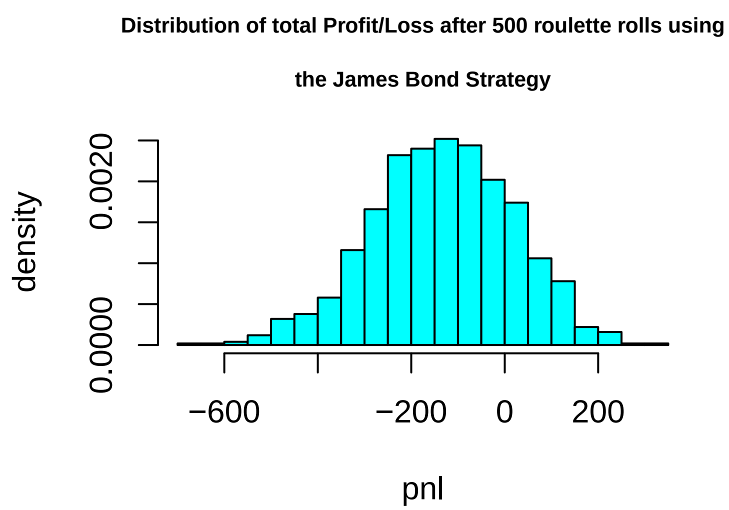
```
  return(rolls)
}

jb_profit <- function(roll, bet=10){
  # for a single roll, calculate winnings
  if (roll >= 19 & roll <= 36) return(0.7*bet - 0.3*bet)
  if (roll >=13 & roll <= 18) return(5*0.25*bet - 0.75*bet)
  if (roll == 0) return(35*0.05*bet - 0.95*bet)

  return(-bet)
}

simulate_games <- function(n=1000, rpg=500){
  # Function returns the profits from n games with rpg rolls per game
  # P is the matrix where the entry in position P_{i, j} is the payout
  # for roll i of game j
  pnl <- vector(length=n)
  P <- matrix(ncol=n, nrow=rpg)  # payout matrix, stores the payout of each roll in the game


  for (i in 1:n){
    game <- roulette(rpg)

    payout_per_roll <- sapply(game, jb_profit)
    P[,i] = payout_per_roll
    pnl[i] <- sum(payout_per_roll)
  }

  return(list(pnl, P))
}

# Similar to simulate_games but with stopping conditions
simulate_games_stop <- function(n=1000, rpg=500, stoploss=-50, stopgain=10){
  pnl <- vector(length=n)
  n_rolls <- vector(length=n)
  P <- matrix(ncol=n, nrow=rpg)  # payout matrix, stores the payout of each roll in the game


  for (i in 1:n){
    game <- roulette(rpg)
    #print(paste0("game:", i))
    payout_per_roll <- sapply(game, jb_profit)

    c_payout <- cumsum(payout_per_roll)
    # get the index of the first violation of stop conditions
    for (j in 1:rpg)
      {
      if (c_payout[j] <= stoploss | c_payout[j] >= stopgain)
        {
        payout_per_roll <- payout_per_roll[1:j]
        break
      }
    }
```

```r
    #print(paste0("stopped after ", j, " rolls"))
    pnl[i] <- sum(payout_per_roll)
    n_rolls[i] <- length(payout_per_roll)
    P[1:j,i] = payout_per_roll

  }
  result_df <- data.frame(pnl=pnl, survival=n_rolls)

  return(list(result_df, P))
}
```

Question 1

After simulating 1000 games and measuring the profits from each, we found that $\bar{x}$, expected profit/loss is around -135, and the variance σ^2 lies around 22300.

```r
results <- simulate_games()
pnl <- results[[1]]
payout_per_roll <- results[[2]]


# the expected gain/loss falls around -135
# variance is around 23000
truehist(pnl, main='Distribution of total Profit/Loss after 500 roulette rolls using\n
the James Bond Strategy', xlab='pnl', ylab='density', cex.main=0.7)
```




```r
ev <- mean(pnl); ev
```

```
## [1] -133.515
```

```r
variance <- var(pnl); variance
```

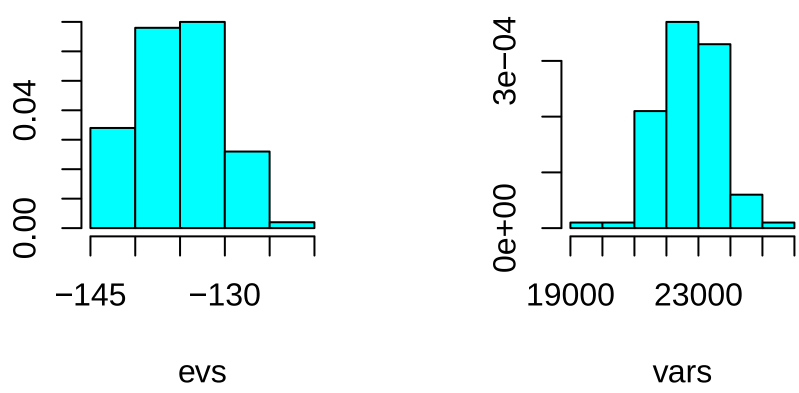
```
## [1] 22897.25
```

For more certain estimates, we tried running this simulation several times and obtained the following distributions of mean and variance. We consider medians here due to the asymmetry of the distributions.

```r
# want more certainty of estimates...
par(mfrow=c(1, 2))
n_sims <- 100
evs <- vector(length=n_sims)
vars <- vector(length=n_sims)

for (i in 1:n_sims){
  results <- simulate_games()
  evs[i] <- mean(results[[1]])
  vars[i] <- var(results[[1]])
}

truehist(evs, nbins=5); median(evs)
```

```
## [1] -135.058
```

```r
truehist(vars); median(vars)
```




```
## [1] 22794.2
```

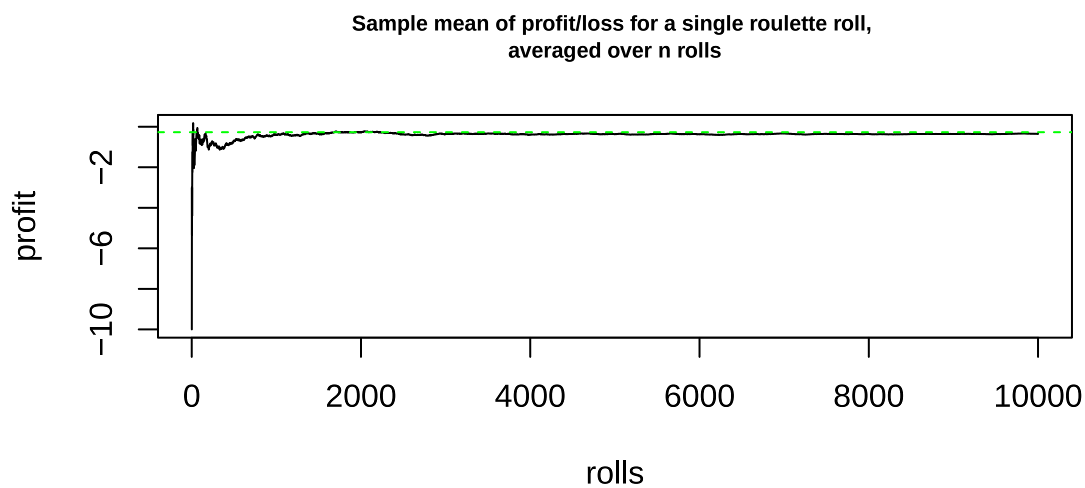
```r
mean(sqrt(vars))
```

```
## [1] 150.6694
```

Question 2

We found earlier that the expected loss for a single roulette roll using this strategy is -0.27. Here it is shown that simulation coincides, with the sample mean getting close to this value with a sample size of 10000.

```
par(mfrow=c(1, 1))
# what is the expected gain/loss on a single roll, and does it converge?
n = 10000
profit_vec <- sapply(roulette(n), jb_profit)
x_hat <- vector(length=n)
for (i in 1:n){
  x_hat[i] <- mean(profit_vec[1:i])
}

plot(1:n, x_hat[1:n], type='l',
     xlab='rolls',
     ylab= 'profit',
     main='Sample mean of profit/loss for a single roulette roll, \naveraged over n rolls',
     cex.main=0.7
)
abline(a=-0.27, b=0, col='green', lty=2)
```



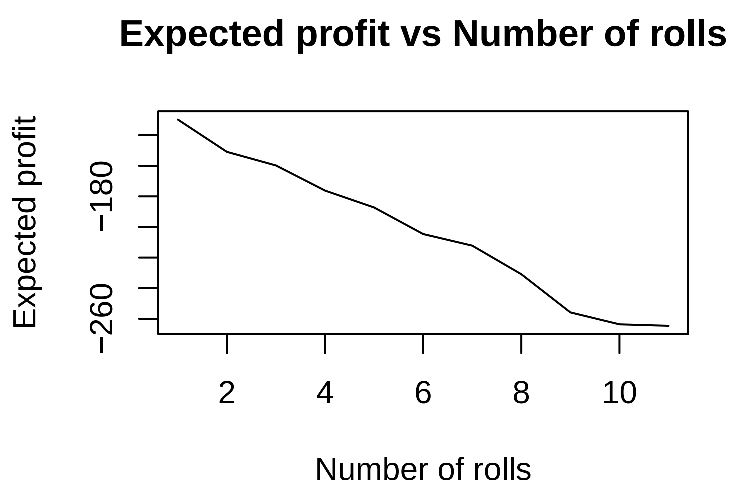

```
print(paste0("Estimate for n = ",n,": ",x_hat[n]))
```

```
## [1] "Estimate for n = 10000: -0.3506"
```

Next we investigated the effect of the number of rolls per game on the expected profit from a game. There is a clear negative trend between the number of rolls per game and expected profit from a game. We can conclude from this that games of less rolls are preferable.

```
rolls_vec <- seq(500, 1000, 50)

n <- length(rolls_vec)
```

```r
t1_ev_winnings <- vector(length=n)

for (i in 1:n){
   result <- simulate_games(rpg=rolls_vec[i])
   pnl <- result[[1]]

   # for individual expected profit histograms
   # truehist(pnl, main=paste0('Distribution of total Profit/Loss after ',
   #                            rolls_vec[i],
   #                            ' roulette rolls using\n the James Bond Strategy'),
   #          xlab='pnl', ylab='density', cex.main=0.7)
   t1_ev_winnings[i] <- mean(pnl)
}

plot(1:n, t1_ev_winnings, type='l', main='Expected profit vs Number of rolls',
     xlab = "Number of rolls", ylab='Expected profit')
```

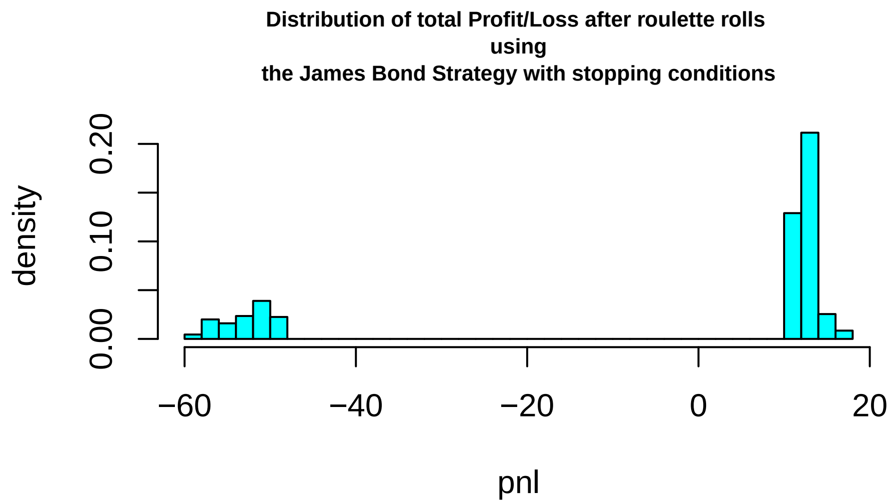


James Bond Strategy with stop conditions

If we set termination conditions, where if the gain/loss exceed certain amounts we stop, then how does this affect the expected loss for each game? For this test, each game is terminated if the total profit thus far passes the 10, or if total losses exceed -50.

We immediately see a drastic improvement in the expected loss per game, and the variance is also significantly reduced. This is because due to the stop conditions, there are much fewer rolls per game in general, as can be seen the histogram of rolls per game below.

```r
results <- simulate_games_stop(stoploss=-50, stopgain=10)
df <- results[[1]]

truehist(df$pnl, main='Distribution of total Profit/Loss after roulette rolls
using\nthe James Bond Strategy with stopping conditions',
xlab='pnl', ylab='density', cex.main=0.7, nbins=50)
```




```
ev <- mean(df$pnl); ev
```

```
## [1] -4.422
```

```
variance <- var(df$pnl); variance
```

```
## [1] 806.4884
```

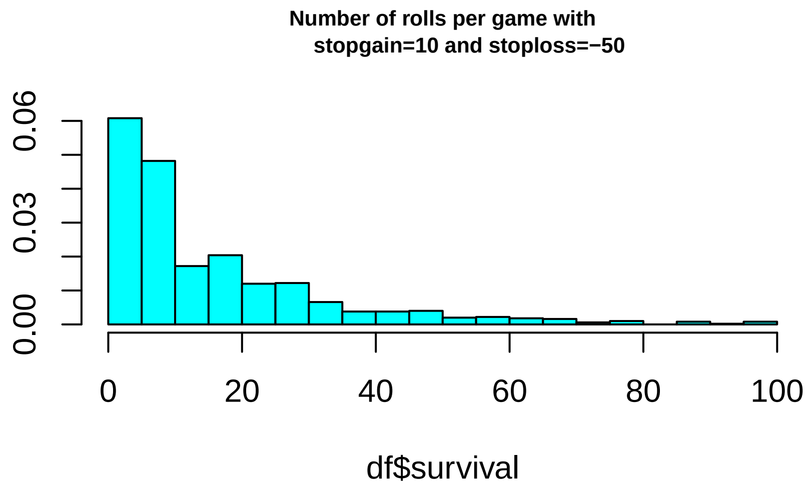
```
survival <- mean(df$survival);survival
```

```
## [1] 15.234
```

```
summary(df$survival)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00    3.00    8.00   15.23   21.00   98.00
```

```
truehist(df$survival, main='Number of rolls per game with
         stopgain=10 and stoploss=-50', cex.main=0.7)
```


Number of rolls per game with
stopgain=10 and stoploss=−50
0.00
0.03
0.06
0
20
40
60
80
100
df$survival