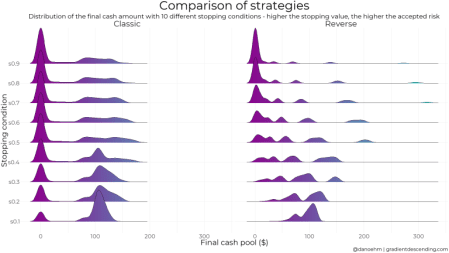
[](http://gradientdescending.com/martingale-strategies-dont-work-but-we-knew-that-simulation-analysis-in-r/)

It’s generally accepted that Martingale betting strategies don’t work. But people still gravitate towards them because they are intuitive. I was curious to find out how they actually perform.

*Disclaimer: I do not encourage gambling, nor do I gamble myself but the games are good examples of stochastic processes.*

**At a glance**

* The classic and Reverse Martingale strategies don’t work. They will almost certainly have you leaving with less money than what you started with, or more often than not no money at all.
* The reverse strategy is generally less risky, higher potential to win big but lower returns on average.
* Going all in on the first bet is the best way to double your money and save your time.

**Classic Martingale**

The Classic Martingale strategy is as follows. Make an initial bet $x. If you win, bet $x on the next round. If you lose, double your previous bet. In a nutshell you bet $2^{n}x where nis the number of losses in a row. The intention is to make your money back on the next win.

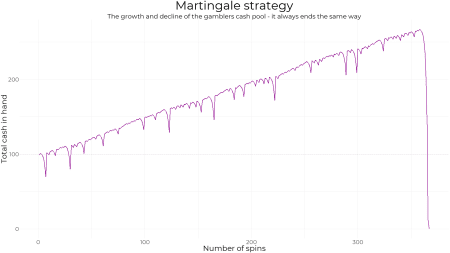
Assume we start with $100 and our initial bet is $1. We keep playing until there is not enough money in the cash pool to make the next bet. Also assume we are playing roulette on an American table and place bets on red or black which both have the probability 18/38. It only takes a streak of 6 losses before the game is over because we don’t have enough cash to double our bet for a 7th time. If we see a streak of 10 or more it really starts to get out of control.

| **trials** | **bet** | **cumulative\_loss** |
| --- | --- | --- |
| 1 | 1 | 1 |
| 2 | 2 | 3 |
| 3 | 4 | 7 |
| 4 | 8 | 15 |
| 5 | 16 | 31 |
| 6 | 32 | 63 |
| 7 | 64 | 127 |
| 8 | 128 | 255 |
| 9 | 256 | 511 |
| 10 | 512 | 1023 |

The probability of losing 6 in a row is P(\text{Streak of 6 losses}) = (1-p)^6 = 0.011. Sounds unlikely, but it will occur more often than you think. With each win we will win $1, so once we have won 27 times we’ll have enough cash in order to afford a losing streak of 6 and bet on the 7th.

It’s more likely we’ll have a few wins and losses before observing a long losing streak that takes us out of the game. The question is how many trials (spins of the roulette wheel) will we place bets on before we lose our money and play stops? A slight variation I’ve applied is, if there is not enough money left to double the bet, we will simply bet the remainder in the cash pool, in other words go all-in. More in line with what someone might do.

This simulation was a relatively lucky one, winning over $170 and almost 400 trials, however one bad streak and it’s all over. You can plot as many simulations as you like, some are shorter and some are longer but they all end the same way.

[](https://i1.wp.com/gradientdescending.com/plots/example-vanilla.png)

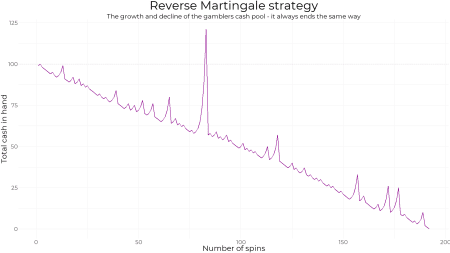
This is a typical pattern for the classic strategy.

**Reverse Martingale**

Instead of doubling the bet upon a loss, double the bet upon a win. Not surprisingly this ends the same way as the classic strategy.

The players cash amount slowly decreases by $1 on each loss. Occasionally there is a big win, rather than a big loss. With this strategy you can watch your money dwindle away rather than vanish in front of your eyes.

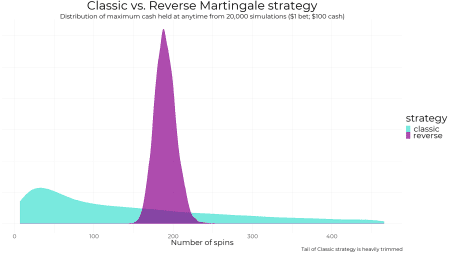
This plot is using the same win-lose sequence as the one above. In this case the maximum cash held throughout the game is higher with the classic strategy than the reverse.

[](https://i0.wp.com/gradientdescending.com/plots/example-reverse.png)

**Number of trials for each strategy**

These strategies was simulated 20,000 times. The distribution of the number of trials shows how long a typical game will last until bankruptcy. The classic strategy has a very long tail, so potentially could be playing for a very long time. The maximum number of trials is this simulation was 64254. But you could also be playing for a very short time.

The reverse strategy has a slightly higher median number of trials but much less variable than the classic strategy meaning you can be sure to play between 166 and 217 spins.

[](https://i0.wp.com/gradientdescending.com/plots/classic-vs-reverse-spins-1-100.png)

|  | **min** | **2.5%** | **10%** | **50%** | **90%** | **97.5%** | **max** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| classic | 7 | 10 | 22 | 170 | 1132 | 2759 | 64254 |
| reverse | 152 | 167 | 172 | 191 | 207 | 213 | 226 |

**Doubling your money**

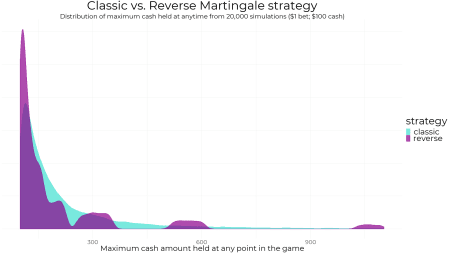
Assume the goal is to double your money. What is the probability you’ll double your money before going bust? After 20,000 simulations for both strategies, the probability you will double your money using…

* The Classic Martingale strategy is 0.35
* The Reverse Martingale strategy is 0.28

The Classic Martingale strategy tends to do better on average, but only slightly. Neither of these strategies are better than simply playing once and going all-in which is 0.47

**Maximum cash**

The distribution of the maximum amount of cash held by the player at any given time during the game shows the classic strategy has the slight edge over the reverse strategy on average. Although the reverse strategy has the potential for massive wins if you score a good winning streak.

[](https://i1.wp.com/gradientdescending.com/plots/classic-vs-reverse-max-cash-1-100.png)

|  | **min** | **2.5%** | **10%** | **50%** | **90%** | **97.5%** | **max** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| classic | 100 | 101 | 107 | 158 | 425 | 926 | 19894 |
| reverse | 100 | 100 | 102 | 133 | 534 | 2062 | 131163 |

However, keep in mind these simulations all resulted in total loss at the end of the game. Perhaps the key is to know when to stop?

**Stopping conditions**

Rather than stopping once you reach a specified amount which may not happen, stop when the bet reaches a specified amount.

We could specify a streak length, however a better idea is to specify a ratio of bet to total cash. This way the is stopping condition is dynamic. For example, if there is a winning streak we’ll have more money in which to bet.

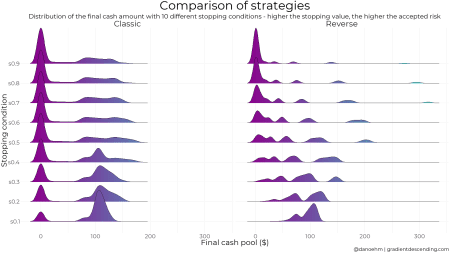
Essentially by using this ratio we are fixing a certain level risk rather than betting amount. The ratio is calculated as

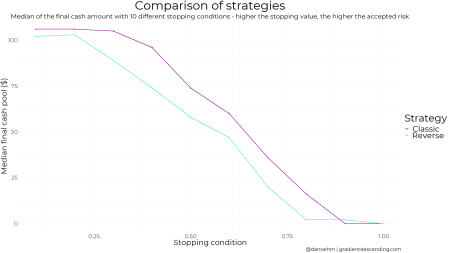
\[\text{stopping ratio} = \frac{\text{bet amount}}{\text{total cash amount}}\]

If we fix a stopping ratio of 0.1, we could place 4 bets before the ratio exceeds 0.1. If our initial cash pool was $200 we could place 5 bets until the ratio exceeds 0.1.

If we stop as soon as this ratio is reached it means we’re on a losing streak so it makes sense in the case of the classic strategy to bet again until our next win and walk away.

In the case of the reverse it makes sense to leave as soon as the ratio is met since we’re on a winning streak.

[](https://i2.wp.com/gradientdescending.com/plots/stopping-conditions-1.png)

[](https://i0.wp.com/gradientdescending.com/plots/stopping-conditions-median-plot.png)

There are clear differences and similarities between the two strategies.

* The median final amount for the low risk stopping condition is actually slightly above 100 for both strategies, meaning it is likely you will break even or take home a couple of bucks.
* As the risk goes up the median value decreases where the classic strategy tends to have a higher median than the reverse strategy.
* Higher the risk, the higher potential for bigger wins.
* The reverse strategy has some interesting features. The ‘bumps’ occur when the risk is great enough to allow the next bet when you’re on a streak. However the chance of this streak is low enough that it doesn’t improve the strategy.

Overall the stopping strategies help to minimise loss rather than lock in wins, so on average you will still lose your money. The Martingales don’t work.

**Code bits**

The code for the plots in this post can be found on [github](https://github.com/doehm/martingale). The Martingale function is below.

# libraries

library(tidyverse)

# martingale function

martingale <- function(bet, cash, p, stop\_condn = Inf, stop\_factor = Inf, reverse = FALSE, plot = TRUE, stop\_on\_next\_win = TRUE){

bet\_vec <- vector(mode = "numeric")

cash\_vec <- vector(mode = "numeric")

outcome <- vector(mode = "numeric")

winnings <- vector(mode = "numeric")

total\_cash <- vector(mode = "numeric")

trial <- 0

total\_cash[1] <- cash

while(total\_cash[max(trial, 1)] > 0){

# iterate through trials

trial <- trial + 1

# update cash pool

if(trial == 1){

cash\_vec[trial] <- cash

}else{

cash\_vec[trial] <- total\_cash[trial-1]

}

# set bet

if(!reverse){

if(outcome[trial - 1] == 1 || trial == 1){

bet\_vec[trial] <- bet

}else{

bet\_vec[trial] <- min(2\*bet\_vec[trial-1], cash\_vec[trial]) # if there isn't enough to double the bet just bet what is left

}

}else{

if(outcome[trial - 1] == 0 || trial == 1){

bet\_vec[trial] <- bet

}else{

bet\_vec[trial] <- min(2\*bet\_vec[trial-1], cash\_vec[trial]) # if there isn't enough to double the bet just bet what is left

}

}

# stop condition

if(bet\_vec[trial]/cash\_vec[trial] > stop\_condn){  
 if(stop\_on\_next\_win & !reverse){  
 stop\_trigger <- TRUE  
 }else{  
 outcome[trial] <- NA  
 winnings[trial] <- NA  
 total\_cash[trial] <- cash\_vec[trial]  
 break  
 }  
 }

outcome[trial] <- sample(c(0,1), 1, prob = c(1-p, p))

winnings[trial] <- bet\_vec[trial]\*outcome[trial] - bet\_vec[trial]\*(1-outcome[trial])

total\_cash[trial] <- cash\_vec[trial] + winnings[trial]

# stop condition

if(total\_cash[trial] >= stop\_factor\*cash) break

}

# make the plot

g1 <- NULL

if(plot){

df <- data.frame(trials = 1:trial, cash = total\_cash)

gg <- ggplot() +

geom\_line(data = df, mapping = aes(x = trials, y = cash), col = "darkmagenta", lty = 1, size = 1) +

geom\_hline(yintercept = cash\_vec[1], col = "grey", lty = 2) +

theme\_minimal() +

labs(

x = "Number of spins",

y = "Total cash in hand",

title = ifelse(reverse, "Reverse Martingale strategy", "Martingale strategy"),

subtitle = "The growth and decline of the gamblers cash pool - it always ends the same way"

) +

ylim(0, NA)

print(gg)

}

return(list(

bet = bet\_vec,

cash = cash\_vec,

outcome = outcome,

total\_cash = total\_cash,

trials = trial,

plot = gg))

}

# run the simulation and plot the output

# try different parameters to see the effect

martingale(1, 100, 18/38, reverse = FALSE, plot = TRUE, stop\_condn = 1,  
 stop\_on\_next\_win = TRUE)

Plot.R

|  |
| --- |
| library(tidyverse) |
|  | library(showtext) |
|  | library(magick) |
|  | library(png) |
|  | library(gridExtra) |
|  | library(parallel) |
|  | library(ggridges) |
|  |  |
|  | # theme setup |
|  | font\_add\_google("Montserrat", "montserrat") |
|  | showtext\_auto() |
|  |  |
|  | my\_theme <- function(scale, drop.y = FALSE){ |
|  | out <- theme\_minimal() + |
|  | theme( |
|  | text = element\_text(family = "montserrat", size = 32\*scale), |
|  | plot.title = element\_text(hjust = 0.5), |
|  | axis.text = element\_text(size = 18\*scale), |
|  | axis.title = element\_text(size = 24\*scale), |
|  | plot.caption = element\_text(size = 16\*scale), |
|  | plot.subtitle = element\_text(size = 20\*scale, hjust = 0.5, margin = margin(t = 10)) |
|  | ) |
|  |  |
|  | if(drop.y){ |
|  | out <- theme\_minimal() + |
|  | theme( |
|  | text = element\_text(family = "montserrat", size = 32\*scale), |
|  | plot.title = element\_text(hjust = 0.5), |
|  | axis.text = element\_text(size = 18\*scale), |
|  | axis.title = element\_text(size = 24\*scale), |
|  | plot.caption = element\_text(size = 16\*scale), |
|  | plot.subtitle = element\_text(size = 20\*scale, hjust = 0.5, margin = margin(t = 10)), |
|  | axis.title.y = element\_blank(), |
|  | axis.text.y = element\_blank(), |
|  | axis.ticks.y = element\_blank() |
|  | ) |
|  | } |
|  | return(out) |
|  | } |
|  |  |
|  | martingale(1, 100, 18/38, reverse = FALSE, plot = TRUE) |
|  | martingale(1, 100, 18/38, reverse = TRUE, plot = TRUE, stop\_condn = 1) |
|  |  |
|  |  |
|  | # $1 bet; $100 cash |
|  | sim <- sapply(1:20000, function(x) martingale(bet = 1, cash = 100, p = 18/38, reverse = FALSE)$trials) |
|  | scale <- 1 |
|  | data.frame(trials = sim) %>% |
|  | dplyr::filter(trials < quantile(sim, 0.975)) %>% |
|  | ggplot(aes(x = trials)) + |
|  | geom\_density(fill = "turquoise", col = "turquoise", alpha = 0.7) + |
|  | labs( |
|  | x = "Number of spins", |
|  | y = "Density", |
|  | title = "Martingale strategy", |
|  | subtitle = "Distribution of number of trials from 20,000 simulations until bankruptcy ($1 bet; $100 cash)", |
|  | caption = "@danoehm | gradientdescending.com" |
|  | ) + |
|  | my\_theme(scale, drop.y = TRUE) |
|  | quantile(sim, c(0.025, 0.1, 0.5, 0.9, 0.975)) |
|  |  |
|  | # reverse martingale |
|  | sim\_reverse <- sapply(1:20000, function(x) martingale(bet = 1, cash = 100, p = 18/38, reverse = TRUE)$trials) |
|  | scale <- 1 |
|  | data.frame(trials = sim\_reverse) %>% |
|  | dplyr::filter(trials < quantile(sim\_reverse, 0.999)) %>% |
|  | ggplot(aes(x = trials)) + |
|  | geom\_density(fill = "turquoise", col = "turquoise", alpha = 0.7) + |
|  | labs( |
|  | x = "Number of spins", |
|  | y = "Density", |
|  | title = "Reverse Martingale strategy", |
|  | subtitle = "Distribution of number of trials from 20,000 simulations until bankruptcy ($1 bet; $100 cash)", |
|  | caption = "@danoehm | gradientdescending.com" |
|  | ) + |
|  | my\_theme(scale, drop.y = TRUE) |
|  | quantile(sim\_reverse, c(0.025, 0.1, 0.5, 0.9, 0.975)) |
|  |  |
|  | # dual plot |
|  | sim\_max\_cash <- sapply(1:20000, function(x) martingale(bet = 1, cash = 100, p = 18/38, reverse = FALSE)$cash %>% max) |
|  | sim\_max\_cash\_reverse <- sapply(1:20000, function(x) martingale(bet = 1, cash = 100, p = 18/38, reverse = TRUE)$cash %>% max) |
|  | save(sim\_max\_cash, sim\_max\_cash\_reverse, file = "./martingale/sim-max-cash.Rdata") |
|  | scale <- 1 |
|  | data.frame( |
|  | final\_cash\_amount = c(sim\_max\_cash, sim\_max\_cash\_reverse), |
|  | strategy = c(rep("classic", 20000), rep("reverse", 20000)) |
|  | ) %>% |
|  | dplyr::filter(final\_cash\_amount < quantile(final\_cash\_amount, 0.975)) %>% |
|  | ggplot(aes(x = final\_cash\_amount, fill = strategy, col = strategy)) + |
|  | geom\_density(alpha = 0.7) + |
|  | labs( |
|  | x = "Number of spins", |
|  | y = "Density", |
|  | title = "Classic vs. Reverse Martingale strategy", |
|  | subtitle = "Distribution of maximum cash held at anytime from 20,000 simulations ($1 bet; $100 cash)" |
|  | # caption = "Tail of Classic strategy heavily trimmed" |
|  | ) + |
|  | my\_theme(scale, drop.y = TRUE) + |
|  | scale\_fill\_manual(values = c("turquoise", "darkmagenta")) + |
|  | scale\_color\_manual(values = c("turquoise", "darkmagenta")) |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | # vanilla |
|  | clust <- makeCluster(10) |
|  | clusterExport(clust, varlist = c("martingale", "%>%")) |
|  | stop\_seq <- seq(0.1, 1, 0.1) |
|  | final\_cash\_amount\_list\_v <- lapply( |
|  | stop\_seq, |
|  | function(j) parSapply(clust, 1:20000, function(x) martingale(bet = 1, cash = 100, p = 18/38, stop\_condn = j, reverse = FALSE)$total\_cash %>% tail(1)) |
|  | ) |
|  |  |
|  | # reverse |
|  | final\_cash\_amount\_list\_r <- lapply( |
|  | stop\_seq, |
|  | function(j) parSapply(clust, 1:20000, function(x) martingale(bet = 1, cash = 100, p = 18/38, stop\_condn = j, reverse = TRUE)$total\_cash %>% tail(1)) |
|  | ) |
|  |  |
|  | # disconnect |
|  | stopCluster(clust) |
|  |  |
|  | # fix names |
|  | names(final\_cash\_amount\_list\_v) <- paste0("s", stop\_seq) |
|  | names(final\_cash\_amount\_list\_r) <- paste0("s", stop\_seq) |
|  |  |
|  | # plot ridges |
|  | ridges\_plot <- rbind( |
|  | as\_data\_frame(final\_cash\_amount\_list\_v) %>% |
|  | gather(key = "stopping", value = "final\_cash") %>% |
|  | mutate( |
|  | stopping = factor(stopping), |
|  | Strategy = "Classic" |
|  | ), |
|  | as\_data\_frame(final\_cash\_amount\_list\_r) %>% |
|  | gather(key = "stopping", value = "final\_cash") %>% |
|  | mutate( |
|  | stopping = factor(stopping), |
|  | Strategy = "Reverse" |
|  | ) |
|  | ) %>% |
|  | group\_by(stopping) %>% |
|  | dplyr::filter(final\_cash < quantile(final\_cash, 0.9)) %>% |
|  | ggplot(aes(x = final\_cash, y = stopping, fill = ..x..)) + |
|  | geom\_density\_ridges\_gradient() + |
|  | scale\_fill\_gradientn(colours = colorRampPalette(c("darkmagenta", "turquoise"))(32)) + |
|  | my\_theme(1) + |
|  | labs( |
|  | title = "Comparison of strategies", |
|  | subtitle = "Distribution of the final cash amount with 10 different stopping conditions - higher the number, the higher the accepted risk", |
|  | x = "Final cash pool ($)", |
|  | y = "Stopping condition", |
|  | caption = "@danoehm | gradientdescending.com" |
|  | ) + |
|  | theme(legend.position = "none") + |
|  | facet\_grid(. ~ Strategy) |
|  |  |
|  | ridges\_plot |
|  |  |
|  |  |
|  | sim <- parSapply(clust, 1:20000, function(x) martingale(bet = 1, cash = 100, p = 18/38)$trials) |
|  | save(sim, file = "./martingale/sim\_trials.Rdata") |
|  |  |
|  | sim\_reverse <- parSapply(clust, 1:20000, function(x) martingale(bet = 1, cash = 100, p = 18/38, reverse = TRUE)$trials) |
|  | save(sim\_reverse, file = "./martingale/sim\_trials\_reverse.Rdata") |
|  |  |
|  | # double money |
|  | N <- 20000 |
|  | clust <- makeCluster(10) |
|  | clusterExport(clust, varlist = c("martingale", "%>%")) |
|  | sim\_double\_money <- parSapply(clust, 1:N, function(x) martingale(bet = 1, cash = 100, p = 18/38, reverse = FALSE, stop\_factor = 2)$total\_cash %>% tail(1)) |
|  | sim\_double\_money\_reverse <- parSapply(clust, 1:N, function(x) martingale(bet = 1, cash = 100, p = 18/38, reverse = TRUE, stop\_factor = 2)$total\_cash %>% tail(1)) |
|  | stopCluster(clust) |
|  |  |
|  | sum(sim\_double\_money >= 200)/length(sim\_double\_money) |
|  | sum(sim\_double\_money\_reverse >= 200)/length(sim\_double\_money\_reverse) |
|  | mean(sim\_double\_money) |
|  | mean(sim\_double\_money\_reverse) |
|  |  |
|  | save(sim\_double\_money, sim\_double\_money\_reverse, file = "./martingale/sim\_double\_money.Rdata") |
|  |  |
|  | g\_double\_money <- data.frame( |
|  | final\_cash\_amount = c(sim\_double\_money, sim\_double\_money\_reverse), |
|  | strategy = c(rep("vanilla", N), rep("reverse", N)) |
|  | ) %>% |
|  | ggplot(aes(x = final\_cash\_amount, fill = strategy)) + |
|  | geom\_histogram(position = "identity", alpha = 0.7) + |
|  | labs( |
|  | x = "Final cash amount ($)", |
|  | title = "Distribution of final cash amounts", |
|  | subtitle = "Comparison of Vanilla vs Reverse Martingale strategies stopping when cash is doubled or bust ($1 bet; $100 cash)", |
|  | caption = "@danoehm | gradientdescending.com" |
|  | ) + |
|  | my\_theme(1, drop.y = TRUE) + |
|  | scale\_fill\_manual(values = c("turquoise", "darkmagenta")) |
|  | g\_double\_money |
|  | png("./martingale/double-money.png", width = 1920, height = 1080) |
|  | g\_double\_money |
|  | dev.off() |