Some find it a common knowledge, some find it weird. As a professor I usually teach about Monty Hall problem and year after year I see puzzling looks from students regarding the solution.

Image taken from [http://media.graytvinc.com/images/690\*388/mon+tyhall.jpg](http://media.graytvinc.com/images/690*388/mon+tyhall.jpg)

The original and most simple scenario of the Monty Hall problem is this: You are in a prize contest and in front of you there are three doors (A, B and C). Behind one of the doors is a prize (Car), while behind others is a loss (Goat). You first choose a door (let’s say door A). The contest host then opens another door behind which is a goat (let’s say door B), and then he ask you will you stay behind your original choice or will you switch the door. The question behind this is what is the better strategy?

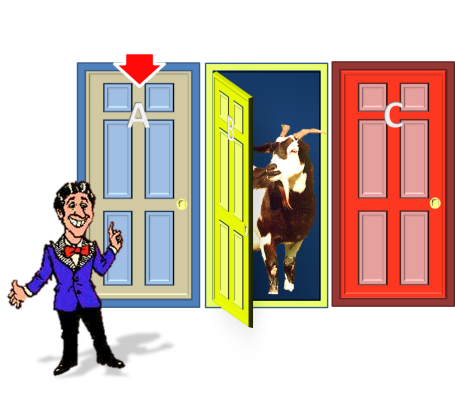


image taken from <https://rohanurich.files.wordpress.com/2013/03/mhp-agc2.png>

The basis of the answer lies in related and unrelated events. The most common answer is that it doesn’t matter which strategy you choose because it is 50/50 chance – but it is not. The 50/50 assumption is based on the idea that the first choice (one of three doors) and the second choice (stay or switch door) are unrelated events, like flipping a coin two times. But in reality, those are related events, and the second event depends on the first event.

At the first step, when you choose one of three doors, the probability that you picked the right door is 33%, or in other words, there is 66,67% that you are on the wrong door. The fact that that in the second step you are given a choice between your door and the other one doesn’t change the fact that you are most likely starting with the wrong door. Therefore, it is better to switch door in the second step.

**Simulation using R**

To explore this a bit further and to have a nice exercise with R, a small simulation of games is created.

First we load the necessary packages

library(ggplot2)

library(scales)

Then we create the possible door combinations

#create door combinations

a<-c(123,132,213,231,312,321)

So what I did was to generate three-digit numbers. The first number will always say behind which door is a car, and two other numbers will say where are goats.

Now let’s prepare the vectors for the simulation

#create results vectors

car=integer(length=100000)

goat1=integer(length=100000)

goat2=integer(length=100000)

initial\_choice=integer(length=100000)

open\_door=integer(length=100000)

who\_wins=character(length=100000)

﻿

Now we are ready for the simulation

#create 100.000 games

for (i in 1:100000){

#set up a situation

doors<-sample(a,1) #randomly pick a door combination

car[i]<-doors %/% 100 #the first number is which door is the right door

goat1[i]<-(doors-car[i]\*100)%/%10 #where is the first wrong door

goat2[i]<-doors-car[i]\*100-goat1[i]\*10 #where is the second wrong door

#have a person select a random door

initial\_choice[i]<-sample(c(1,2,3),1)

#now we open the wrong door

if (initial\_choice[i]==car[i]){

open\_door[i]=sample(c(goat1[i],goat2[i]),1) #if the person is initially on the right door we randomly select one of the two wrong doors

} else if (initial\_choice[i]==goat1[i]) {

open\_door[i]=goat2[i]

} else {open\_door[i]=goat1[i]} #if the person is initially on the wrong door, we open the other wrong door

#stayer remains by his initial choice and switcher changes his choice

if (initial\_choice[i]==car[i]){who\_wins[i]="Stayer"} else {who\_wins[i]="Switcher"}

}

monty\_hall=data.frame(car, goat1,goat2,initial\_choice,open\_door,who\_wins)

And now we got a nice analysis of 100.000 games. To put the most important result into chart we use ggplot2

ggplot(data=monty\_hall, aes(who\_wins, fill=who\_wins)) +

geom\_bar(aes(y = (..count..)/sum(..count..))) + #crude but effective

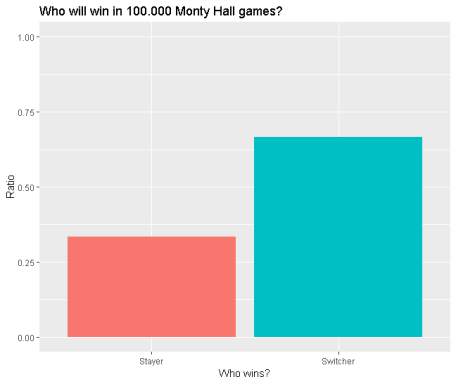
ylim(0,1)+

ylab("Ratio")+

xlab("Who wins?")+

theme(legend.position = "none")

And  
now we got a nice analysis of 100.000 games. To put the most important result  
into chart we use ggplot2



So it is definitely  
better to switch door!

For more reading  
refer to <https://en.wikipedia.org/wiki/Monty_Hall_problem>

Happy coding