Quantitative Exercise

Marine Collery and Benjamin Seregi

10/30/2017

```
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
```

Introduction

We investigate the Monty Hall problem. There are three doors, behind one door there is a car, behind the others there is a goat. We assume that the distribution of this is uniform. You can pick a door and win either a goat or a car. Once you made your decision, the game host will show you one door with a goat, after that the host will give you the chance to change your mind and select another door. Question is: what is the best strategy? Sticking to your first choice or changing your mind?

Dataset

The dataset is generated by a C program that saves the samples in a CSV file.

Setting the directory to the data directory.

```
getwd()
```

```
## [1] "/home/ben/Desktop/research-methodology/paper"
```

```
setwd("/home/ben/Desktop/research-methodology/paper")

getwd()
```

```
## [1] "/home/ben/Desktop/research-methodology/paper"
```

Reading in data frame.

```
mdf <- read.csv("samples-100000.csv",header=T,sep=',',stringsAsFactors = F)

# check what variables are there
names(mdf)
```

```
## [1] "CarDoor"  "Selected" "S"        "NS"
```

```
summary(mdf)
```

```
##    CarDoor            Selected              S            
##  Length:100000      Length:100000      Length:100000     
##  Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character  
##       NS           
##  Length:100000     
##  Class :character  
##  Mode  :character  
```

We can then factorize the data.

```r
#factorize the data

mdf$Sf=as.factor(mdf$S)
mdf$NSf=as.factor(mdf$NS)
#summary(mdf)
```

The function countWin will count when the player wins and has changed his mind and when he has win without changing his mind.

```r
#Count all the wins in 2 sets : Switch or No Switch
countWin <- function(df) {
    # df is a dataframe

    l1 <- length(df$NS)
    sumWinS <- 0
    sumWinNS <- 0

    for (i in 1:l1) {

        if (df$S[i]=='W' & df$NS[i]=='L') {
          sumWinS <- sumWinS+1
          df$sumWinSwitched[i] <- sumWinS

          #Not changed:
          df$sumWinNotSwitched[i] <- sumWinNS
        }
        else if (df$S[i]=='L' & df$NS[i]=='W'){
          sumWinNS <- sumWinNS+1
          df$sumWinNotSwitched[i] <- sumWinNS

          #Not changed:
          df$sumWinSwitched[i] <- sumWinS
        }
    }
    return(df)

}

mdf <- countWin(mdf)

summary(mdf)
```

```
##    CarDoor            Selected              S            
##  Length:100000      Length:100000      Length:100000     
##  Class  :character  Class  :character  Class  :character 
##  Mode   :character  Mode   :character  Mode   :character 
##                                                          
##                                                          
##                                                          
##       NS            Sf        NSf       sumWinNotSwitched sumWinSwitched
##  Length:100000      L:33212   L:66788   Min.   :    1     Min.   :    0  
##  Class  :character  W:66788   W:33212   1st Qu.: 8424     1st Qu.:16577  
##  Mode   :character                      Median :16736     Median :33264  
##                                         Mean   :16690     Mean   :33310  
##                                         3rd Qu.:25012     3rd Qu.:49988  
##                                            Max.   :33212     Max.   :66788
```

Variables

We only have very few variables for this problem that are relevant for the analysis. Those variables are:

- S : (equal to “W” or “L”) this variable represents whether the player has Win or Lose IF he has switched (S).
- NS : (equal to “W” or “L”) this variable represents whether the player has Win or Lose IF he has not switched (NS).

NS and S are dependent of each other: if one is “W” the other is “L”. We could have kept only one of them but keeping both facilitates the comprehension and the display of the result.

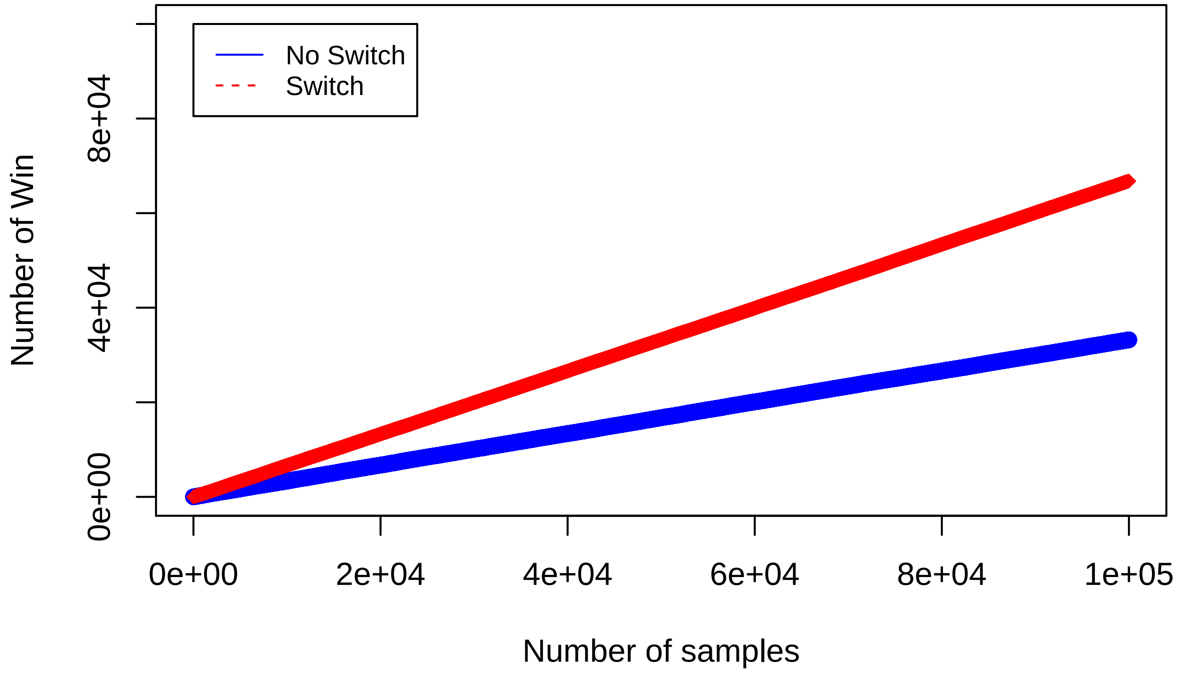
The other attributes of the data frame either facilitate the analysis (sumWinSwitched and sumWinNotSwitched) or are required to understand the problem (CarDoor and Selected). The last ones are required for the generation.

Analysis

```r
#Plot Win without switch
plot(mdf$sumWinNotSwitched, xlim=c(0.0,length(mdf$NS)), ylim=c(0.0,length(mdf$NS)), xlab='Number of samp

#Add Win with switch
lines(mdf$sumWinSwitched, pch=18, col="red", type="b", lty=2)

#Legend
legend(0, length(mdf$NS)-1, legend=c("No Switch", "Switch"),
       col=c("blue", "red"), lty=1:2, cex=0.8)
```



According to the theory, the probability of winning with a switch is 2/3, and it can be calculated using conditional probability (refer to https://en.wikipedia.org/wiki/Monty_Hall_problem#Solutions_using_conditional_probability_and_other_solutions).

```
probWinWithSwitch = mdf$sumWinSwitched[length(mdf$NS)]/(length(mdf$NS))
```

The probability that we calculated using the usual combinatorial probability is

```
probWinWithSwitch
```

```
## [1] 0.66788
```

We can then conclude that our samples have the expected properties.