Analysis of switching strategy for Monty Hall problem

Tejovardhan Medamarti

31 January 2017

Overview and summary

The Monty Hall problem is a brain teaser, in the form of a probability puzzle (Gruber, Krauss and others), loosely based on the American television game show **Let's Make a Deal** and named after its original host, Monty Hall. The problem was originally posed in a letter by Steve Selvin to the American Statistician in 1975 (Selvin 1975a), (Selvin 1975b).

Our objective in this assignment is to write an R-script to simulate the conditional probabilities and variances of the distributions for winning a car vs. a goat for the Monty Hall Problem. Include the cases of switching or not switching. Compare these cases to determine the best strategy.

Note: Following packages are required to run the below report. -plyr -ggplot2 -scales

```
rm(list=ls())
require (plyr)
```

```
## Loading required package: plyr
```

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
require(scales)
```

```
## Loading required package: scales
```

Data Preparation:

```
MontyHall.Problem.Simulation <- function (noofgates, r, strategy)
{
  dfmontymatrix <- data.frame(matrix(0 ,ncol = 6, nrow = r))
  cnames <- c("Gatechosen", "PrizeGate","MontySelection","Strategy", "Result", "Noofsimulations")
  colnames(dfmontymatrix) <- cnames
  #noofgates <- 1:3

  dfmontymatrix[, "Noofsimulations"] <- r  # No of simulations.
  for (ri in 1: r)
  {
    indexprize <- sample(noofgates, 1, replace = TRUE)
    Gatechosen <- sample(noofgates, 1, replace = TRUE)

    if(indexprize != Gatechosen)
      {
      MontySelection <- noofgates[-c(indexprize, Gatechosen)]
    } else
      {
          MontySelection <- sample(noofgates[-c(indexprize, Gatechosen)], 1)
      }
```

```
    dfmontymatrix[ri, "Gatechosen"] <- Gatechosen
    dfmontymatrix[ri, "PrizeGate"] <- indexprize
    dfmontymatrix[ri, "MontySelection"] <- MontySelection
    dfmontymatrix[ri, "Strategy"] <- sample(strategy, 1, replace = TRUE)

    if (dfmontymatrix[ri,"Strategy"] == "stay")
    {
      if (dfmontymatrix[ri,"Gatechosen"] == dfmontymatrix[ri, "PrizeGate"])
      {
        dfmontymatrix[ri, "Result"] = "win"
      }
      else
      {
        dfmontymatrix[ri, "Result"] = "loss"
      }
    } else
    {

      if (dfmontymatrix[ri,"Gatechosen"] != dfmontymatrix[ri, "PrizeGate"])
      {
        dfmontymatrix[ri, "Result"] = "win"
      }
      else
      {
        dfmontymatrix[ri, "Result"] = "loss"
      }
    }
  }

  return (with (dfmontymatrix, table(Noofsimulations, Strategy, Result)))

}
```

Data simulation:

Lets simulate the data with datapoints (1000, 10000, 20000)

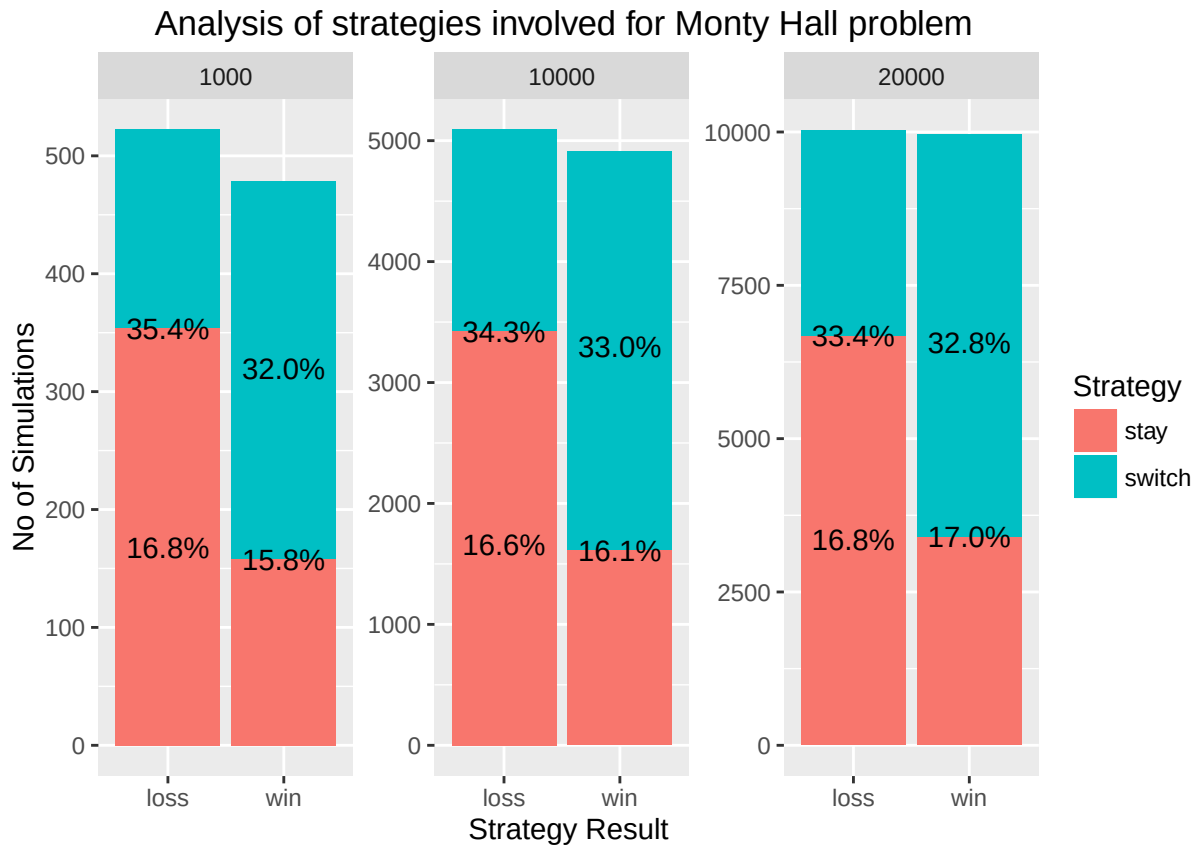
```
r <- c( 1000, 10000, 20000)
noofgates <-1:3
strategy <- c("stay", "switch")
sampledata <- lapply(r, function(r) MontyHall.Problem.Simulation(noofgates, r, strategy))
final.output <- ldply (sampledata, data.frame)
final.output$Noofsimulations <- as.integer(as.character(final.output$Noofsimulations))
final.output <- mutate(final.output,
                       Percentage = percent(Freq/ Noofsimulations))

print (final.output)
```

```
##    Noofsimulations Strategy Result Freq Percentage
## 1             1000     stay   loss  354      35.4%
## 2             1000   switch   loss  168      16.8%
## 3             1000     stay    win  158      15.8%
## 4             1000   switch    win  320      32.0%
```

```
## 5            10000     stay   loss 3426      34.3%
## 6            10000   switch   loss 1663      16.6%
## 7            10000     stay    win 1612      16.1%
## 8            10000   switch    win 3299      33.0%
## 9            20000     stay   loss 6680      33.4%
## 10           20000   switch   loss 3354      16.8%
## 11           20000     stay    win 3402      17.0%
## 12           20000   switch    win 6564      32.8%
```

Visualization:

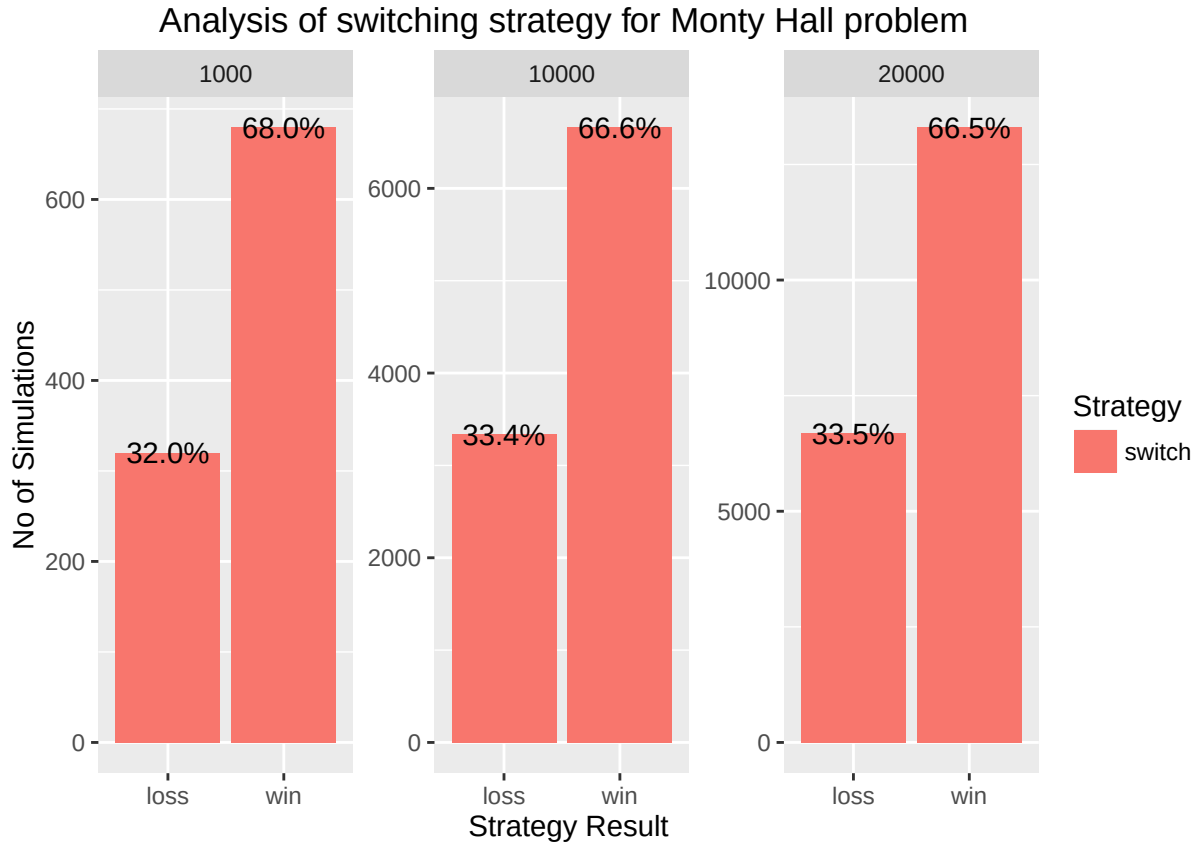
```
ggplot(final.output, aes(Result,Freq, fill = Strategy ))+
  geom_bar( stat = "identity")+
  geom_text(aes(label = Percentage))+
  facet_wrap(~Noofsimulations, scales = "free" )+
  labs(title = "Analysis of strategies involved for Monty Hall problem ", y = "No of Simulations", x =
```



Lets do only for strategy = "switch"

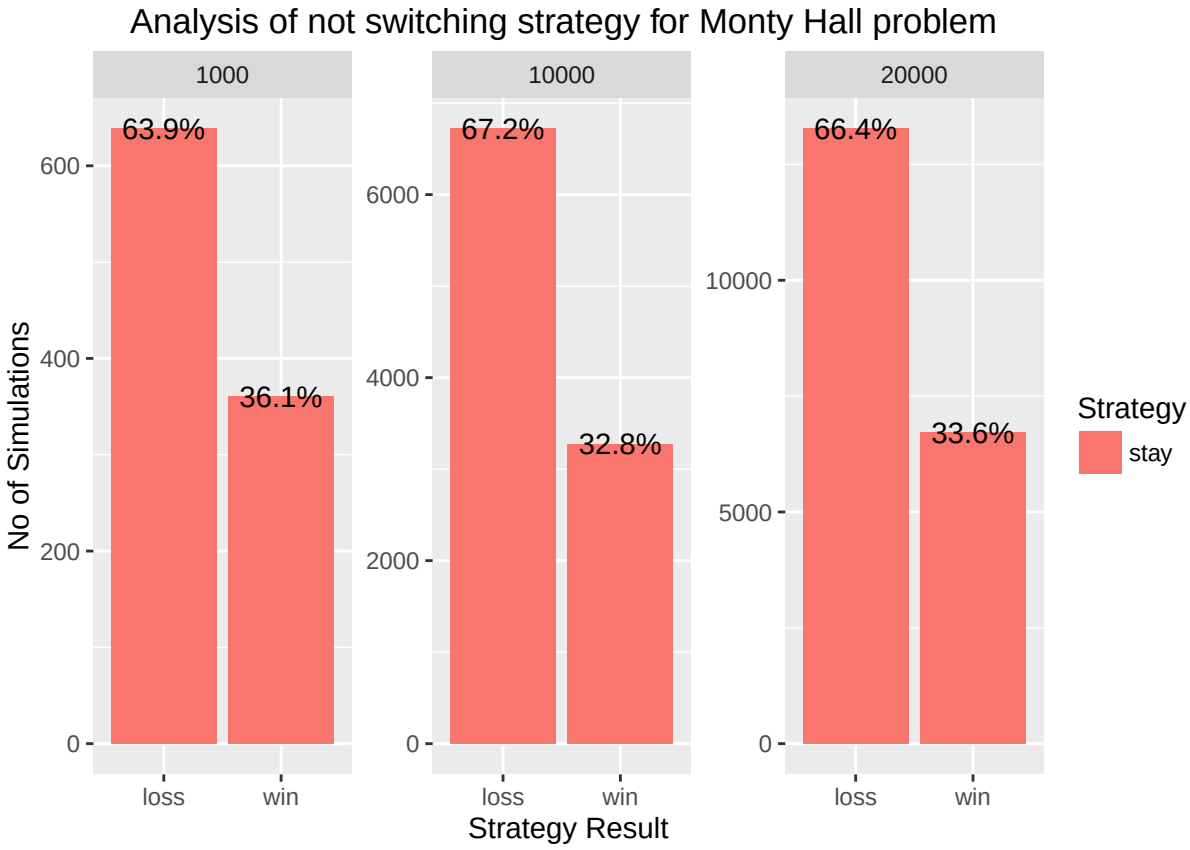
```
strategy <-  "switch"

sampledata <- lapply(r, function(r) MontyHall.Problem.Simulation(noofgates, r, strategy))

final.output <- ldply (sampledata, data.frame)
final.output$Noofsimulations <- as.integer(as.character(final.output$Noofsimulations))
final.output <- mutate(final.output,
```

```
                      Percentage = percent(Freq/ Noofsimulations))

ggplot(final.output, aes(Result,Freq, fill = Strategy ))+
  geom_bar( stat = "identity")+
  geom_text(aes(label = Percentage))+
  facet_wrap(~Noofsimulations, scales = "free" )+
  labs(title = "Analysis of switching strategy for Monty Hall problem ", y = "No of Simulations", x = "S
```



Lets do only for strategy = "stay"

```
strategy <-  "stay"

sampledata <- lapply(r, function(r) MontyHall.Problem.Simulation(noofgates, r, strategy))

final.output <- ldply (sampledata, data.frame)
final.output$Noofsimulations <- as.integer(as.character(final.output$Noofsimulations))
final.output <- mutate(final.output,
                       Percentage = percent(Freq/ Noofsimulations))

ggplot(final.output, aes(Result,Freq, fill = Strategy ))+
  geom_bar( stat = "identity")+
  geom_text(aes(label = Percentage))+
  facet_wrap(~Noofsimulations, scales = "free" )+
  labs(title = "Analysis of not switching strategy for Monty Hall problem ", y = "No of Simulations", x
```



Conclusion:

Based on the above visualizations, Switching strategy has more chances to win than staying.