Monty Hall Problem - Solutions

There are many possible ways how to solve it - you may either write a complex script which will simulate in details the way how the game is proceeding. In this case I first thought how to simplify it, while still being correct.

```
behind.doors <- matrix (ncol = 3, nrow = 1000)
item <- c('car', 'goat', 'goat')

for (i in seq (1, 1000))
  behind.doors[i,] <- sample (item, 3)

# Scenario 1: I always stick to my original decision
# How is it done: I always point to the same doors (e.g. No. 1), but as well
# could be No. 2 or No. 3
sum (behind.doors[,1] == 'car')/1000

# Scenario 2: I always change and choose the other doors
# How is it done: I always change - I first choose one door (e.g. No. 1),
# but after the host shows me the goat in the other doors, I will change.
# It means that if in the doors of the first choice (No. 1) is a car, I will lose;
# if there was a goat, for sure I will win a car.
sum (behind.doors[,1] != 'car')/1000
```

Other solutions - not using `for` loop, but function `replicate` instead:

```
item <- c('car', 'goat', 'goat')
behind.doors <- t (replicate (1000, expr = sample (item, 3)))
sum (behind.doors[,1] == 'car')/1000
sum (behind.doors[,1] != 'car')/1000
```

And finally the most complicated solution, which almost exactly mimics the whole game:

```
win.stick <- NULL
win.change <- NULL
item <- c('car', 'goat', 'goat')
# always stick to the decision
for (game in seq (1, 1000))
{
  behind.doors <- sample (item, 3)  # what is behind doors
  my.choice <- sample (1:3, 1) # which doors I choose
  what.I.got <- behind.doors[my.choice]  # what I got
  win.stick <- append (win.stick, if (what.I.got == 'car') 1 else 0)
}
sum (win.stick)/1000

# always change and choose the other door
for (game in seq (1, 1000))
```

```
{
  behind.doors <- sample (item, 3)
  my.choice <- sample (1:3, 1)
  host.can.open <- (1:3)[(1:3)!=my.choice & behind.doors != 'car']  # the host can open
    #only the door which I haven't chosen and where is not a car
  host.really.open <- if (length (host.can.open) > 1) sample (host.can.open, 1) else
    host.can.open  # if behind the door I chose was a car, the host  can randomly choose
      # which doors to open, since both have goat; if my chosen doors has a goat,
      # the host cannot choose - must open the doors where is not a car
  what.I.got <- behind.doors[c(-my.choice, -host.really.open)]
  win.change <- append (win.change, if (what.I.got == 'car') 1 else 0)
}
sum (win.change)/1000
```