# The Monty Hall problem

May 12, 2022

Suppose you're on a game show, and you're given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say No. 1, and the host, who knows what's behind the doors, opens another door, say No. 3, which has a goat. He then says to you, "Do you want to pick door No. 2?" Is it to your advantage to switch your choice?

We use a simulation to find the answer. First, we define a function to simulate a game:

```
set.seed(1)
whichDoor = function(choice, nds=3) {
    doors = rep("goat", nds)
    car = sample(1:nds, 1)
    doors[car] = "car"
    if (doors[choice] == "car") {
        host = sample((1:nds)[-choice], 1)
    } else if (nds == 3){
        host = (1:nds)[-c(choice, car)]
    } else {
        host = sample((1:nds)[-c(choice, car)], 1)
    }
    if (nds ==3) {
        switch = (1:nds)[-c(choice, host)]
    } else {
    switch = sample((1:nds)[-c(choice, host)], 1)
    }
    return(c(choice=choice, car=car, switch=switch, host=host))
}

# look at ten games
for (i in 1:10)
    print(whichDoor(sample(1:3, 1)))
```

Here are the results for the ten games:

```
  choice    car switch   host
      3      1      1      2
 choice    car switch   host
      2      1      1      3
 choice    car switch   host
```

```
     3      1      1      2
choice    car switch   host
     2      3      3      1
choice    car switch   host
     3      2      2      1
choice    car switch   host
     1      3      3      2
choice    car switch   host
     1      1      2      3
choice    car switch   host
     2      2      1      3
choice    car switch   host
     1      3      3      2
choice    car switch   host
     1      3      3      2
```

Now let's define a function to count the frequency from a larger number of the simulated games.

```
countMTH = function(n, nds=3){
    games = replicate(n, whichDoor(sample(1:3, 1), nds=nds))
    mean(games[1,] == games[2,])
    mean(games[3,] == games[2,])
    return (c(mean(games[1,] == games[2,]), mean(games[3,] == games[2,])))
    }

# simulate 100 games to approximate the probabilities
probabilities = paste(countMTH(100), collapse=", ")
```

The approximate probabilities of keeping the original choice and switching are 0.41, 0.59.