Monty Hall Problem

Foundations of Statistical Inference (PLSC 503)

2/14/2019

The Monty Hall Problem[1] is a very famous puzzle that you may have learned about in a probability class, or seen in a movie. The puzzle was originally stated as follows:

> *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?*

As you may know from the movie "21"[2], the correct answer is to switch. This puzzle generated a bunch of controversy. When Marilyn vos Savant presented the correct solution in her column for *Parade* magazine in 1990, thousands of people wrote in to tell her she was wrong[3].

To avoid confusion for the problem, make the following standard assumptions:

1. The host (Monty) must always open a door that was not picked by the contestant.
2. Monty must always open a door that has a goat behind it and never the car! If he has a choice to make between two goat doors, he chooses at random.
3. Monty must always offer the chance to switch between the originally chosen door and the remaining closed door.

Show that contestants who switch have a better chance of winning the car than contestants who stick with their initial choice. First, provide an analytic solution. Second, convice yourself of this using simulation. For the simulation solution, you should write a function called `monty_hall` that takes two arguments: `sims` for the number of simulations to run and `strategy` for one of three strategy choices: "stay", "switch" or "random". The "random" strategy should just pick one of the three doors at random. The input to `sims` should be an integer (≥ 1000) and the `strategy` argument should be a character. In the function, assume that your initial choice of door is random.

There is no "right" way to do the simulation. You don't need to use any random number generators. The only function you need to use is `sample()`. I would recommend using a for loop and taking advantage of nested if statements. This may not be the most efficient way, but I think it is the easiest. There are actually two methods (that I know of) for the analytic solution. I will show both in section.

[1] https://en.wikipedia.org/wiki/Monty_Hall_problem
[2] clip: https://www.youtube.com/watch?v=Zr_xWfThjJ0
[3] Have a look at some of the comments she received here: http://marilynvossavant.com/game-show-problem/

A sketch:

```r
monty_hall <- function(strategy = "switch", sims = 1000, ...) {
  # Initialize door vector.

  # Initialize vector to keep track of wins.

  # Inner loop.
  for(i in 1:sims) {
  # Randomly determine winning door.

  # Randomly determine your first choice.

  # The host reveals one of the doors you did not pick (which contains a
  #   goat). [Caution: what happens if you get lucky and pick the winning
  #   door first?]

  # Result if you decide to stay.

  # Or if you decide to switch.

  # Or if you pick a random strategy.

  # Tally the outcome for this sim. End loop.
  }

  # Return the win percentage, averaged across all sims.
  return(...)
}
```