Simulating the Monty Hall Problem

https://advancedsqlpuzzles.com

For this puzzle we are going to run a simulation of the Monty Hall problem. If you are unfamiliar with the problem, read the Wikipedia article here.

“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?”

For this simulation, lets add a catch by parametrizing all the variables.

- The number of doors
- The number of goats/prizes available
- The number of doors the contestant must choose
- The number of doors the host must reveal
- The number of prize doors the contestant must choose to be considered a winner

The most difficult part of this simulation will be writing the SQL statement to ensure you have a valid set of parameter values; as you will need to guarantee you have a winning possibility of doors/goats available given 1) the number of doors the contestant can choose, 2) the number of doors the contestant can switch, 3) the number of goats/prize doors available, and 4) the number of doors the host must reveal.

In my script, when the contestant switches doors, they must switch all their doors for new doors, and not a partial amount.

It is worth noting that the Monty Hall problem is a variant of an urn probability problem, with the difference being a winning possibility must occur in the Monty Hall problem. Here is a link to the Wikipedia article on urn problems.

“In probability and statistics, an urn problem is an idealized mental exercise in which some objects of real interest (such as atoms, people, cars, etc.) are represented as colored balls in an urn or other container. One pretends to remove one or more balls from the urn; the goal is to determine the probability of drawing one color or another, or some other properties.”

When running the simulation, you need to understand the law of large numbers. Here is a link to the Wikipedia article on the law of large numbers.

“In probability theory, the law of large numbers (LLN) is a theorem that describes the result of performing the same experiment a large number of times. According to the law, the average of the results obtained from a large number of trials should be close to the expected value and will tend to become closer to the expected value as more trials are performed.”

To ensure I have coded the simulation correctly, I run the simulation 10,000 times under the traditional 3 door scenario. In this scenario there is an 1/3 chance of winning before switching, and a 2/3 chance of winning after switching, which my results closely replicate. To see a mathematical proof using Bayes theorem, see the Monty Hall Wikipedia article.

Unfortunately, I am not a probability master and do not have the ability to create the mathematical proofs given different scenarios, so I can only validate my script against the above traditional scenario. If we create a scenario where there are 100 doors, 30 doors are winning doors, the other 70 doors are goats, the contestant must choose 20 doors, the host reveals 10 doors, and the contestant much ultimately choose 10 winning doors to be considered a winner, I get a 3% chance of winning before switching doors and a 8% chance of winning after switching doors.

Here is a link to my solution. Note this links to my GitHub repository.

GitHub - The Monty Hall Simulation

Whew, I think that is it.

Happy coding!