

CASE STUDY 037

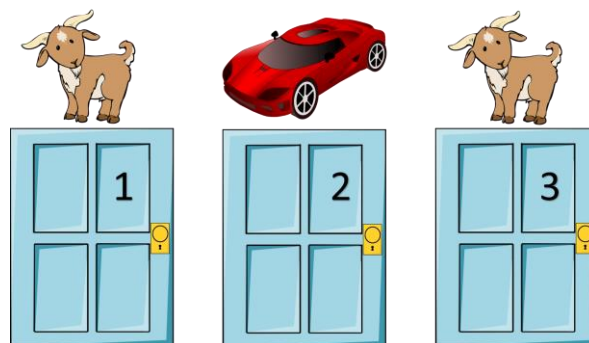
[Python]

Monty Hall Probabilities Trial

Difficulty Level: 3 of 3

The Monty Hall problem is a brain teaser, in the form of a probability puzzle, loosely based on the American television game show Let's Make a Deal and named after its original host, Monty Hall.

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 number 1, and the host, who knows what's behind the doors, opens another door, say number 3, which has a goat. He then says to you, "Do you want to pick door number 2?" Is it to your advantage to switch your choice?



It turns out that the probabilities of the contestant win if he changes the door is the double of not to change, roughly 67% and 33% chance respectively.

This case study will prove this converging probabilities in the long run (long number of trials) through random experiments, comparing the outcome (winning cases) of changing or not the door.

To build our experiment, we will need to create functions to represent the game's rules:

1) `newGame()`:

Create a function that randomly chooses one of the three initial positions of the game, and returns a dict like:

```
{ 'door1': 'car', 'door2': 'goat', 'door3': 'goat' }
```

This function doesn't have to receive any input parameter.

2) `guestChoice()`:

Create a function that randomly chooses one door, as the guest does in the show, and return the number of the door, like: **'door1'**

This function doesn't have to receive any input parameter.

3) `openOneDoor(game, chosen_door):`

Create a function that receives a created game and the door chosen by the guest. The host of the show should open one door – randomly, once again – but he can't open the door containing the car neither the same door the guest had chosen. The function should return the modified dict of the game, presenting the value 'open' to the door opened, like:

```
{'door1': 'car', 'door2': 'open', 'door3': 'goat'}
```

4) `guestChange(game, chosen_door):`

Create a function that receives a created game and the door chosen by the guest. At this stage of the game, the guest should decide if he will change to the other closed door or not. The function should return the number of the door, like: **'door1'**

For comparison, let's keep the first choice in a different variable.

5) `checkResult(game, chosen_door):`

Create a function that receives a created game and a door, returning **'win'** in case the door contains a car or **'lose'** in case the door contains a goat.

To run the experiment, we need to define the number of trials and repeat the following steps (each game):

- Create a new game
- Simulate a guest choice
- Open one door
- Change the door
- Check the result for both choices and keep track of the number of winning for both cases.
- Print the percentage of winning for both cases.

Good luck!

Difficulty note: this is a difficult assignment. Do not be surprised that there will be lots of nuances we have not covered off in the courses. But just like in the Real Life – there will be things training has not prepared you for and you will need to do research to find how to solve the problems at hand. If you get stuck, check the clues file.