The Monty Hall Game

Assignment- 1

Suprio Dubey

2013036

25/10/2022

Question

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 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?
Q1) What is the probability to win switching door, knowing that the host knows where it is the price and open always the door with a goat?
Q2) What is the probability to win switching door, knowing that the host choose randomly which door to open?
Q3) What is the probability to win switching door, knowing that the host choose sometimes randomly and sometimes only goats, which door to open?
Q4) Can you devise a simulation to test your answers for questions Q1, Q2 and Q3?

Answer1

The first question is to see if the probability of our chances of winning a car is higher if we switch the door knowing that the host knows where the prize is and always open the door with a goat.
We can start by seeing that before the host opens the door there is an equal probability i.e $\frac{1}{3}$ that the car is in each door.After, I chose the door **1**, the probabilty of the car being in the other two doors which were not chosen is still $\frac{2}{3}$ and does not change. Now, the host knows where the car is so he will only open the door which has a goat. So, when the host opens door **3** and we find a goat, we eliminate the chance of the goat being at door **3** so the probability of the the car being in door **2** is now $\frac{2}{3}$. Therefore, we see the probability of finding the car in door **2** is higher $\frac{2}{3}$ after the goat was revealed than at door **1** which is still $\frac{1}{3}$.So switching choices will help win the prize, since it has a higher probability of $\frac{2}{3}$

ANSWER 2

Now in the second question we see that the host is unaware of where the prize is. Suppose I chose door **1**.

1. If the car is behind door **1** it doesnt matter whihch door the host opens. we win if we stayand we lose if we switch.

2. If the car is behind door **2** $\frac{1}{2}$ of the time the host finds the car giving us the probability of finding the goat as $\frac{1}{6}$.Here we lose if we stay and win if we switch.

3. This will be simillar to case no. 2 by changing the door 2 with 3

We see that ignoring the cases where the found the car there are two cases, where we win if we stay and two cases where we win, if we switch. Hence the probability is $\frac{1}{2}$.

ANSWER 3

In this case we get an average out of the two previous probabilities i.e $(\frac{1}{3} + \frac{1}{2})/2 = \frac{5}{12}$. If we switch our winning probability switches to $(\frac{2}{3} + \frac{1}{2})/2 = \frac{7}{12}$.

Answer 4 Python Code for the simulation of the problem

```
#!/usr/bin/python
import random
import numpy as np
import matplotlib.pyplot as plt
import random





def host_opendoor(player_door, prize_door,num_door, case):
  if(case == 0):

    if (player_door == prize_door):

      option = num_door.pop(num_door.index(player_door))
      hostoption = random.choice(num_door)
      rem = num_door.pop(num_door.index(hostoption))
    else:

      option = num_door.pop(num_door.index(player_door))

      for i in num_door:

        if i != prize_door:
          host_option = i
      hostoption = num_door.pop(num_door.index(host_option))
  if(case ==1):
    option = num_door.pop(num_door.index(player_door))
    hostoption = random.choice(num_door)




  return(hostoption,player_door,prize_door, num_door)




def result(sw, num_test):
  win_switch = 0
  win_no_switch = 0
  lose_switch = 0
  lose_no_switch = 0
  win_r_s=0
```

```
lose_r_s =0
win_r_ns =0
lose_r_ns =0
dont_count=0

for i in range(num_test):
    case=random.randint(0,1)# when case = 0 only we can check question 1.For case = 1 only we can check question 2.

    num_door = [0,1,2]

    prize_door = random.choice((num_door))

    player_door = random.choice((num_door))
    hostdoor,playerdoor,prizedoor,nd = host_opendoor(player_door,prize_door,num_door, case)
    if(case == 0):
        if(sw == True):
            playerdoor = nd[0]

        if(sw == True and playerdoor == prizedoor ):
            win_switch +=1
        if(sw == True and playerdoor != prizedoor ):
            lose_switch +=1
        if(sw == False and playerdoor == prizedoor ):
            win_no_switch +=1
        if(sw == False and playerdoor != prizedoor ):
            lose_no_switch +=1
    if(case ==1):
        #print(nd)
        #print(hostdoor)
        if(sw == True):
            nd.pop(nd.index(hostdoor))
            playerdoor = nd[0]

        #print(playerdoor)
        #print(prizedoor)
        if(hostdoor != prizedoor and playerdoor == prizedoor and sw == True):
            win_r_s += 1
        if(hostdoor != prizedoor and playerdoor != prizedoor and sw == True):
            lose_r_s += 1
        if(hostdoor != prizedoor and playerdoor == prizedoor and sw == False):
            win_r_ns += 1
        if(hostdoor != prizedoor and playerdoor != prizedoor and sw == False):
            lose_r_ns += 1
```

```
      if(hostdoor == prizedoor):
        dont_count +=1


  print((win_switch+win_r_s)/(num_test-dont_count),(win_no_switch+
    win_r_ns)/(num_test-dont_count))#lose_switch,lose_no_switch,
    win_r_s , lose_r_s,win_r_ns, lose_r_ns, dont_count)

result(False,10000)
result(True,10000)
```