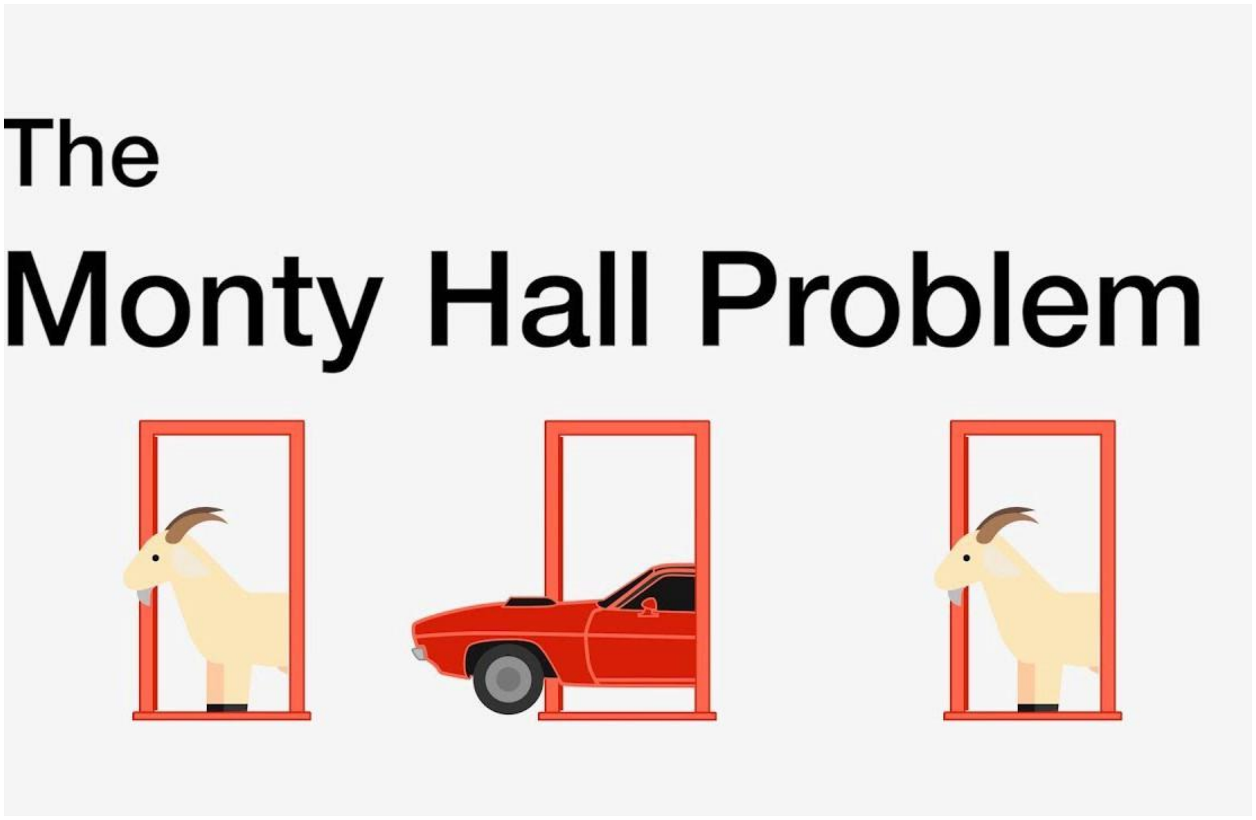


1. The Monty Hall problem is a well-known puzzle in probability derived from an American game show, Let’s Make a Deal. (The original 1960s-era show was hosted by Monty Hall, giving this puzzle its name.) Intuition leads many people to get the puzzle wrong, and when the Monty Hall problem is presented in a newspaper or discussion list, it often leads to a lengthy argument in letters-to-the-editor and on message boards.

The game is played like this:

* The game show set has three doors. A prize such as a car or vacation is behind one door, and the other two doors hide a valueless prize called a Zonk; in most discussions of the problem, the Zonk is a goat.

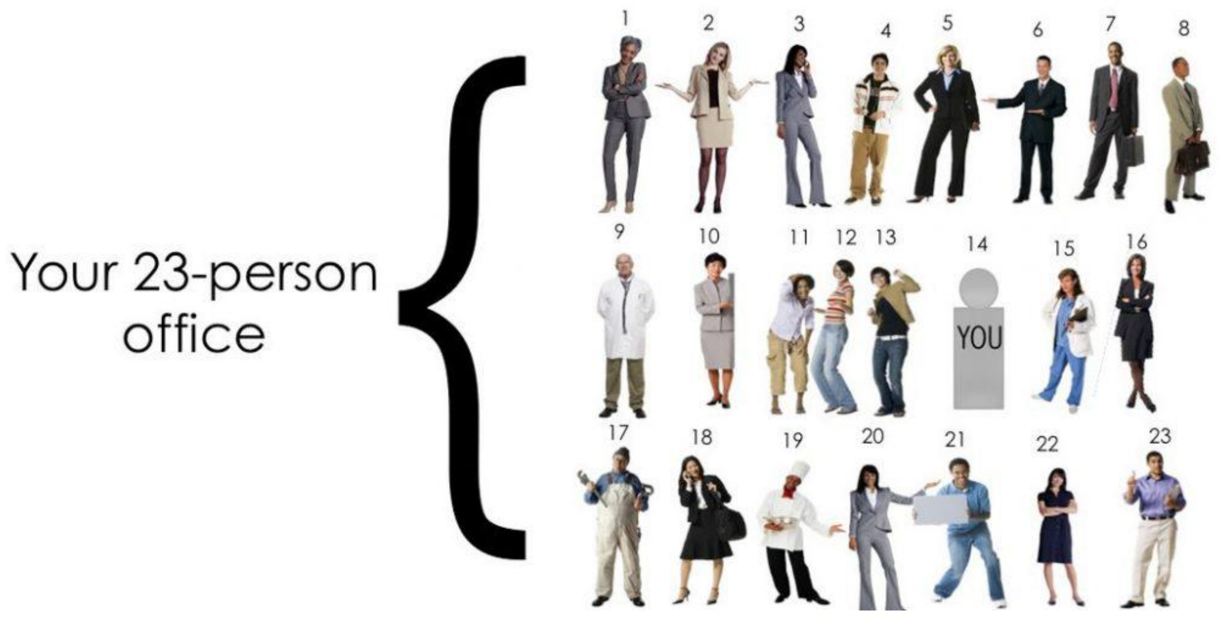
* The contestant chooses one door. We'll assume the contestant has no inside knowledge of which door holds the prize, so the contestant will just make a random choice.
* The smiling host Monty Hall opens one of the other doors, always choosing one that shows a goat, and always offers the contestant a chance to switch their choice to the remaining unopened door.
The contestant either chooses to switch doors, or opts to stick with the first choice.
* Monty calls for the remaining two doors to open, and the contestant wins whatever is behind their chosen door.
* Let's say a hypothetical contestant chooses door #2. Monty might then open door #1 and offer the chance to switch to door #3. The contestant switches to door #3, and then we see if the prize is behind #3.

The puzzle is to identify if switching increases the chance of winning the car, decreases it, or makes no difference? Use python code to play this game multiple times (1000) and analyse your observations ?

Code

```
import random
def switch_function(shown_door, number_of_doors, player_choice):
i = 1
while (i == shown_door or i== player_choice ):
i = (i+1)%(number_of_doors)
return i
def non_prize(host, number_of_doors, player_choice):
i = 1
while (i == host or i== player_choice ):
i = (i+1)%(number_of_doors)
return i
def game(switch, number_of_tests):
for interval in range(0,100):
lost_switch = 0
lost_no_switch = 0
win_switch = 0
```

```
win_no_switch = 0
doorway = [0,1,2]
number_of_doors = len(doorway)
for i in range(0,num_tests):
door_with_prize = random.randint(0, number_of_doors-1)
host = door_with_prize
player_choice = random.randint(0, number_of_doors-1)
original_player_choice = player_choice
shown_door = non_prize_door(host, number_of_doors, player_choice)
if switch == True:
player_choice = switch_function(shown_door,number_of_doors, player_choice)
if player_choice == host and switch == False:
print('Player Wins (No switch) - The player chose door: ', player_choice,' Original choice:
',original_player_choice ,', Door with prize:', door_with_prize, ', Shown Door: ',shown_door )
win_no_switch_cnt = win_no_switch_cnt + 1
elif player_choice == host and switch == True:
print('Player Wins (switch) - The player chose door: ', player_choice,' Original choice:
',original_player_choice , ', Door with prize:', door_with_prize, ', Shown Door: ',shown_door )
win_switch_cnt = win_switch_cnt +1
elif player_choice != host and switch == False:
print('Player Lost (No switch) - The player chose door: ', player_choice,' Original choice:
',original_player_choice , ', Door with prize:', door_with_prize, ', Shown Door: ',shown_door )
lose_no_switch_cnt = lose_no_switch_cnt + 1
elif player_choice != host and switch == True:
print('Player Lost (switch) - The player chose door: ', player_choice,' Original choice:
',original_player_choice , ', Door with prize:', door_with_prize, ', Shown Door: ',shown_door )
lose_switch_cnt = lose_switch_cnt + 1
else:
print('SOMETHING IS WRONG')
interval-=1
return(win_no_switch,win_switch,lose_no_switch,lose_switch, num_tests)
game(0,3)
```



2. Let's say there are 23 employees in any office party. What is the probability that at least two of the employees will have the same birthday ?

A. 0.25
B. 0.50
C. 0.75
D. Can not Say

Ans.B   0.50

```
import math
def nCr(n,r):
f = math.factorial
```

```
return f(n) / f(r) / f(n-r)
one_person_birthday = 1/365
two_person_birthday = 364/365
a = pow(two_person_birthday,nCr(23,2))
print(a)
```