

10-29_Monte_Hall_Problem_Simulation

September 27, 2023

```
[1]: import numpy as np
```

```
[2]: def keep_door():  
    prize = np.random.randint(1, 4)  
    pick = np.random.randint(1, 4)  
    return prize == pick
```

```
[17]: won = np.zeros(10000)  
for i in range(won.size):  
    won[i] = keep_door()
```

```
[18]: won.dtype
```

```
[18]: dtype('float64')
```

```
[19]: np.mean(won)
```

```
[19]: 0.3354
```

```
[5]: host = [True, True, True]  
host[prize - 1] = False  
host[pick - 1] = False  
host
```

```
-----  
NameError                                Traceback (most recent call last)  
/tmp/ipykernel_4665/2474663275.py in <module>  
     1 host = [True, True, True]  
----> 2 host[prize - 1] = False  
     3 host[pick - 1] = False  
     4 host  
  
NameError: name 'prize' is not defined
```

```
[6]: for door, state in enumerate(host):  
    print(door, state)
```

```
0 True
1 True
2 True
```

```
[7]: pick = 2
      open_door = 1
      doors = [True, True, True]
      doors[pick - 1] = False
      doors[open_door - 1] = False
      print(doors)
      doors.index(True) + 1
```

```
[False, False, True]
```

```
[7]: 3
```

```
[37]: prize = 1
       pick = 1
       host = [True, True, True]
       host[prize - 1] = False
       host[pick - 1] = False
       open_door = 0
       for door, state in enumerate(host):
           print(door + 1, state)
           if state:
               open_door = door + 1
               print("open", open_door)
               break
       doors = [True, True, True]
       doors[pick - 1] = False
       doors[open_door - 1] = False
       pick = doors.index(True) + 1
       prize == pick
```

```
1 False
2 True
open 2
```

```
[37]: False
```

```
[8]: def change_door():
      prize = np.random.randint(1, 4)
      pick = np.random.randint(1, 4)
      host = [True, True, True]
      host[prize - 1] = False
      host[pick - 1] = False
      open_door = 0
      for door, state in enumerate(host):
```

```
    if state:
        open_door = door + 1
        break
doors = [True, True, True]
doors[pick - 1] = False
doors[open_door - 1] = False
pick = doors.index(True) + 1
return pick == prize
```

```
[9]: won = np.zeros(10000)
for i in range(won.size):
    won[i] = change_door()
```

```
[10]: np.mean(won)
```

```
[10]: 0.669
```

```
[11]: x = np.arange(1, 20)
print(x)
for i in x:
    print(i)
    if i > 10:
        break
print('done')
```

```
[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19]
1
2
3
4
5
6
7
8
9
10
11
done
```

```
[ ]:
```