

The Monty Hall problem and the effect of goats

Vince Knight

November 3, 2017

1 Introduction: The Monty Hall Problem

The Monty hall problem is a probabilistic problem based on a game show [1]. In this show a contestant was presented with three doors. Behind one of the doors was a car and behind the others was a goat. The contestant would choose a door (which would thus either be hiding a goat or a car) and the host would then open a door showing the location of the goat. At this stage the contestant had a strategic choice:

- Stick: if their original choice had the car they won!
- Switch: take whatever was behind the remaining door.

This puzzled many mathematicians for years as it might seem that the contestant is given a false choice and the probability of winning is in fact the same whether or not they switch. This is however not true: if the contestant switched they doubled their probability of winning!

In this work, we will hopefully clarify this by considering what happens if we modify the game: let us have n goats behind n doors. Everything else remains the same: the host still opens one of the doors behind which is a goat.

In Section 2 Python will be used to simulate both strategies and find the probability of winning. Then in Section 3 a mathematical formula for the probability of winning will be obtained and verified.

2 Simulating more goats

Here are two python functions that simulate sticking or switching with a given number of goats:

```
import random

def stick(number_goats=2):
    """A function to simulate a play of the game when we stick"""
    doors = ['Car'] + number_goats * ['Goat']

    initial_choice = random.choice(doors) # make a choice
    return initial_choice == 'Car'

def switch(number_goats=2):
    """A function to simulate a play of the game when we swap"""
    doors = ['Car'] + number_goats * ['Goat']

    initial_choice = random.choice(doors) # make a choice

    doors.remove(initial_choice) # Switching: remove initial choice
    doors.remove('Goat') # The game show host shows us a goat

    new_choice = random.choice(doors) # We choose our one remaining option

    return new_choice == 'Car'
```

Using this we can simulate the probabilities for the case of 2 goats (the original TV show!):

```
>>> repetitions = 10000
>>> random.seed(0)
>>> prob_win_stick = sum([stick() for rep in range(repetitions)]) / repetitions
```

```
>>> prob_win_switch = sum([switch() for rep in range(repetitions)]) / repetitions
>>> prob_win_stick, prob_win_switch
(0.3346, 0.6636)
```

This indeed looks like the contestant is twice as likely to win if they switch.

3 Analytic formulae

We can compute the probability p_n of winning when switching when there are n goats. In order to win when switching:

- The initial choice must not be the car. If there are n goats, there are a total of $n+1$ doors. Thus, the probability of finding the car is $\frac{1}{n+1}$. Thus, the probability of **not** finding the car is $1 - \frac{1}{n+1}$.
- Switching implies choosing the car from the remaining doors. There have however been 2 doors removed (the original choice and the one that the host opens). Thus the probability of choosing the car is $\frac{1}{n-1}$.

Thus:

$$p_n = \left(1 - \frac{1}{n+1}\right) \left(\frac{1}{n-1}\right) \quad (1)$$

$$p_n = \left(\frac{n+1-1}{n+1}\right) \left(\frac{1}{n-1}\right) \quad (2)$$

$$p_n = \frac{n}{(n+1)(n-1)} \quad (3)$$

$$p_n = \frac{n}{(n^2 - 1)} \quad (4)$$

Using Python to verify the algebra:

```
>>> import sympy as sym
>>> n = sym.symbols('n')
>>> p_n = (1 - 1 / (n + 1)) * (1 / (n - 1))
>>> p_n.simplify()
n/(n**2 - 1)
```

The ratio of how much better it is to switch is thus given by:

$$\alpha_n = \frac{p_n}{1/(n+1)} = \frac{n}{n-1}$$

Finally, let us compare this formulae to simulated values.

```
def ratio(repetitions=50000, number_goats=2):
    """Obtain the ratio of win probabilities"""
    prob_win_stick = sum([stick(number_goats=number_goats)
                          for rep in range(repetitions)]) / repetitions
    prob_win_switch = sum([switch(number_goats=number_goats)
                           for rep in range(repetitions)]) / repetitions
    return prob_win_switch / prob_win_stick
```

Here is some Python code that plots this ratio (both simulated and using the formulae) shown in Figure 1.

```
>>> random.seed(0)
>>> goats = range(2, 25 + 1)
>>> ratios = [ratio(number_goats=n) for n in goats]
>>> theoretic_ratio = [(n / (n - 1)) for n in goats]
>>> plt.figure()
>>> plt.scatter(goats, ratios, label="simulated")
>>> plt.plot(goats, theoretic_ratio, color="C1", label="theoretic")
>>> plt.xlabel("Number of goats")
>>> plt.ylabel("Ratio")
>>> plt.legend()
>>> plt.savefig("simulated_v_expected_ratio_of_win_probability.pdf");
```

Finally let us confirm that as the number of goats grows the effect of switching gets smaller and smaller:

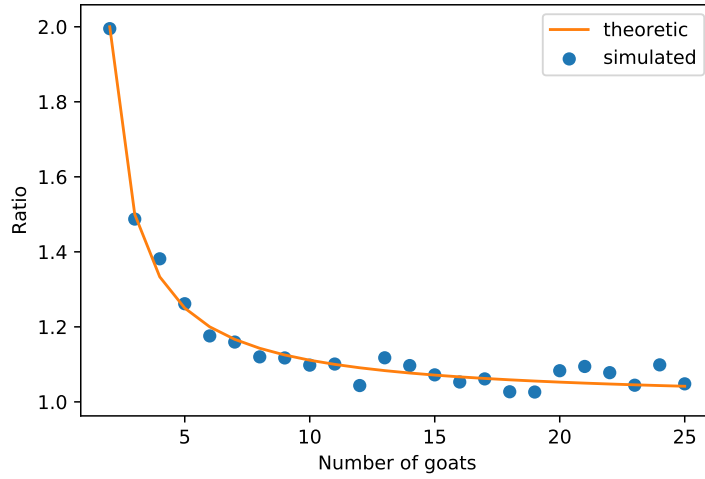


Figure 1: The simulated and expected effect of goats on the win probability

$$\lim_{n \rightarrow \infty} \alpha_n = \lim_{n \rightarrow \infty} \frac{n}{n-1} \quad (5)$$

$$\lim_{n \rightarrow \infty} \alpha_n = \lim_{n \rightarrow \infty} \frac{1}{1-1/n} \quad (6)$$

$$\lim_{n \rightarrow \infty} \alpha_n = 1 \quad (7)$$

Using Python to verify this calculation:

```
>>> import sympy as sym
>>> alpha_n = p_n / (1 / (n + 1))
>>> sym.limit(alpha_n, n, sym.oo)
1
```

4 Conclusion

This report has used mathematics and Python to shed some light on the Monty hall problem. The modified version of the problem considers the effect of having a given number of goats in the game.

An analytical formula is obtained for the probability of winning if the player switches.

As the number of goats increases the beneficial effect of switching is reduced: this makes intuitive sense as the increase of goats implies that the host makes very little change to the problem by showing a door with a goat.

Other avenues for investigation could include seeing the effect of adding more cars and also what happens when the host opens more doors with goats.

References

- [1] Jason Rosenhouse. *The Monty Hall problem: the remarkable story of Math's most contentious brain teaser*. Oxford University Press, 2009.