# MMS

Vanessa Closius, Jonas Tietz, Tronje Krabbe

December 29, 2018

**1.** If we simply pick one of the three shells at random, there is a one in three chance that we have picked the correct shell:

$$P = \frac{1}{3}$$

However, if the thimblerigger uncovers a shell which does not hide the ball first, and allows us to switch, things change. This is an example of the Monty Hall Problem[1] Intuitively, we would say that there is now a 50% chance that our choice was the correct shell, and that our overall odds have not changed — why would the thimblerigger revealing a loosing shell, which we have not even picked, alter our odds? However, we can actually increase our odds by switching our choice of shells to the remaining shell. See Table 1 for a concrete example of why switching is smart. The table shows all possible results, and assumes that we always pick Shell 1.

Table 1: Monty Hall Paradox

| **Shell 1** | **Shell 2** | **Shell 3** | **Result if not switching** | **Result if switching** |
|---|---|---|---|---|
| nil | nil | **ball** | loose | **win** |
| nil | **ball** | nil | loose | **win** |
| **ball** | nil | nil | **win** | loose |

We can see that our odds of winning remain as one in three if we never switch, but increase to two in three if we do switch. Thus:

$$P = \frac{2}{3}$$

**2.** Our Python implementation supports the claims we made in the previous section. Without switching, the relative frequency of winning is at about $\frac{1}{3}$. With switching, it rises to about $\frac{2}{3}$. The plot which our program generates, shown in Figure 1, shows this as well.

[1] https://en.wikipedia.org/wiki/Monty_Hall_problem

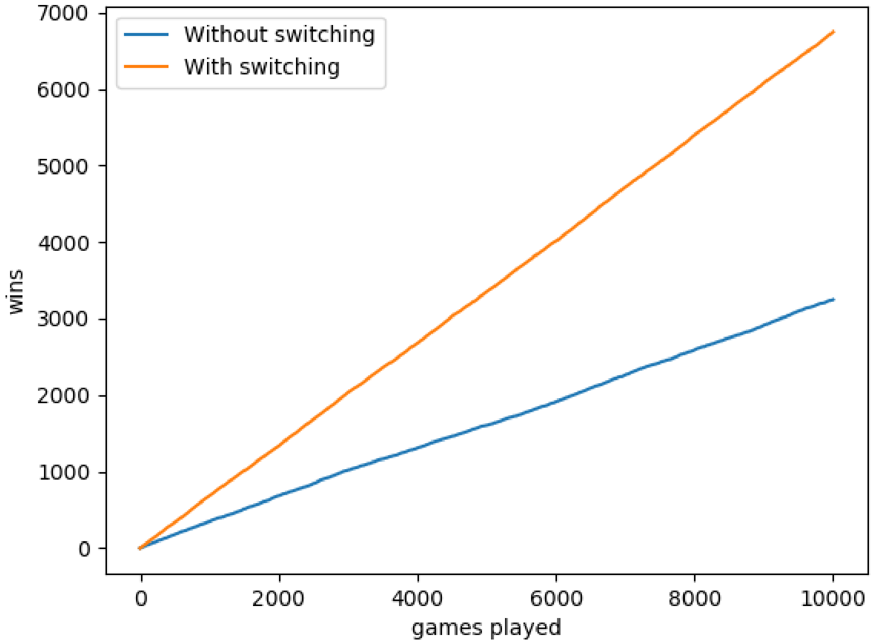


Figure 1: The relative winning frequencies plotted by our software.