**The Monty Hall problem with Python**

**taken from an American reality show called Let's Make a Deal . In this section, I introduce a free Python project to simulate the Monty Hall problem .**

**An introduction to the Monty Hall problem**

**The idea behind the game confuses many people, and when the issue of Monty Hall is discussed in a newspaper or publication, it often leads to lengthy discussions in cyberspace .**

**Monty Hall's problem is as follows :**

**In this reality show, there are three doors. A prize like a car or a fun trip is behind one door, and the post two doors hides a priceless prize called Zonk . In most of the episodes of this reality show, Zonk is a goat !**

**The contestant chooses a door. We assume that he has no idea which door the main prize is behind. So the participant simply makes a random choice .**

**Smiling host Monty Hall opens one of the other doors. He always opens a door with a goat behind it and always gives the contestant the opportunity to change his choice to the unopened door (between the two remaining doors) .**

**The participant chooses between changing and sticking to their first choice .**

**Simulating the Monty Hall problem with Python**

**Simulating the Monty Hall problem is simple as a free Python project. We want to write a function that uses Python's random module to choose which door the prize is hidden behind. The initial choice of the participant, and in the case of elimination, will also be determined by the presenter .**

**An input parameter controls whether the competitor chooses to change the gate, and then the function returns a Boolean indicating whether the competitor's selection was a winning gate .**

**import random**

**from random import seed \_ randint**

**import numpy**

**def game( winningdoor , selecteddoor , change = False ):**

**assert winningdoor < 3**

**assert winningdoor >= 0**

**# Presenter removes the first door that was not selected neither winning**

**removeddoor = next ( i for i in range ( 3 ) if i != selecteddoor and i != winningdoor )**

**# Player decides to change his choice**

**if changed :**

**selecteddoor = next ( i for i in range ( 3 ) if i != selecteddoor and i != removeddoor )**

**# We suppose the player never wants to change his initial choice.**

**return selecteddoor == winningdoor**

**if \_\_name\_\_ == '\_\_main\_\_' :**

**playerdoors = numpy . random . random\_integers ( 0 , 2 , ( 1000 \* 1000 \* 1 ,))**

**winningdoors = [ d for d in playerdoors if game ( 1 , d )]**

**print ( "Winning percentage without changing choice: " , len ( winningdoors ) / len ( playerdoors ))**

**winningdoors = [ d for d in playerdoors if game ( 1 , d , change = True )]**

**print ( "Winning percentage while changing choice: " , len ( winningdoors ) / len ( playerdoors ))**

Amirshayan Jalili

[Shayan138190@gmail.com](mailto:Shayan138190@gmail.com)

<https://github.com/Amirshayan2002>