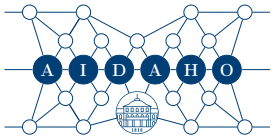
Introduction to Applied Data Science
Prof. Dr. Thomas Dimpfl, Dr. Johannes Bleher, & Sophia Koch
Department of Business Mathematics
and Data Science



Exercise Sheet 1

—

The Monty Hall Problem

Background

Suppose a contestant is presented with three doors, one of which conceals a valuable car, while the other two hide goats. The contestant chooses one door, and then the host, who knows what's behind each door, opens one of the remaining two doors to reveal a goat. The contestant is then given the option to either stick with their initial choice or switch to the other unopened door. Should the contestant reconsider his/her decision and opt to switch doors, or should he/she stay with his/her initial choice?

In the following task, you will set up the Monty Hall problem and compute the winning probabilities for the strategies "stay" and "switch".

Task 1: The Monty Hall Problem

1. Define a vector that contains the numbers 1 to 3 which represent the three doors. Call the vector `doors`.

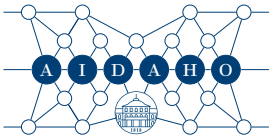
2. Take the role of the host and draw randomly a number from the `doors` vector. This will be the door with the car behind. Call the variable `car`.
   Hint: *Use the function* `sample()`.

3. Take the role of the player and draw randomly a number from the `doors` vector. Call the variable `tipp`.

4. Identify the door that the host opens which has a goat behind. Remember the door cannot reveal the car **and** cannot be the door chosen by the player. Call the variable `opendoor`.
   What happens if the player has chosen the door with the car behind?
   Hint: *Use an if-statement to define the variable* `opendoor` *in this specific case.*

5. Define a vector `remaindoors` that captures the doors which are still closed.

6. Now the player has to decide if he stays with the chosen door or if he switches. Define a new variable `strategy` in the beginning of your code and assign the string "switch" to it.



Use this variable to change your variable `tipp` if the player decides to switch the doors. If you change `strategy` to "stay", the `tipp` stays the same as before. Move on with the strategy "switch".

7. Check if the player has chosen the correct door and won the car, or if the player has chosen a door with a goat behind. Assign the result to the variable `won`.

8. To compute the winning probability of the strategy "switch", repeat the game 1,000 times and store the results in a vector. Compute the winning probability for the strategy "switch". Print the winning probability to the console.
   Hint: *Use a loop. Turn the variable* `won` *of Task 1.7 into a vector and define it before initiating the loop to store your results. The winning probability equals the relative frequency of the times when the player has chosen the door with the car behind.*

9. Repeat Task 1.8 after changing the `strategy` variable to "stay". Which strategy should you choose?

10. Write the function `MHProblem` which has the input argument `strategy` and returns the result of playing the game once. Call your function within the loop of Task 1.8. Try your function for both strategies and compare your results to Task 1.8 and 1.9.