Lab #3

Data Science I

For this Lab we will be doing Exercise 10.4 from our textbook ‘Modern Data Science with R’.

The Monty Hall Problem: The Monty Hall problem illustrates a simple setting where intuition is often misleading. The situation is based on the TV game show ‘Let’s Make a Deal’. First Monty (the host) puts a prize behind one of three doors. Then the player chooses a door. Next, (without moving the prize) Monty opens an unselected door, revealing that the prize is not behind it. The player may then switch to the other nonselected door. Should the player switch?

Many people see that there are now two doors to choose between and feel that since Monty can always open a nonprize door, there is still equal probability for each door. If that were the case, the player might as well keep the original door. A correct intuitive route is to observe that Monty’s door is fixed. The probability that the player has the right door is 1/3 before Monty opens the non-prize door, and remains 1/3 after that door is open. This means that the probability that the prize is behind one of the doors is 2/3, both before and after Monty opens the nonprize door. After Monty opens the nonprize door, the player gets a 2/3 chance of winning by switching to the remaining door. If the player wants to win, they should switch doors.

One way to prove to yourself that switching improves your chances of winning is through simulation. Write a function to perform the Monty Hall simulation multiple times to calculate the percent of times the game was won.

Here are some tips:

[1] First write a function that does one play of the Monty Hall simulation. Have an argument ‘switch’ which tells if the contestant uses the strategy of switching the door (switch = TRUE) or nor (switch = FALSE).

[a] First use sample() to select the door that the contestant picks and the door that the prize is under
[b] Next use an if else statement to show what door is revealed if (1) the contestant guessed the correct door or (2) if the contestant did not guess the correct door
[c] Use an if statement to switch guesses of the door if switch = TRUE
[d] return a logical (TRUE/FALSE) indicating if the game was won or not

[2] Next write a function that performs the simulation multiple times. Have an argument for the number of times to run the simulation (n) and the strategy (switch). Check out the replicate() function which may be useful.

Please turn in an html of your R markdown document with the functions and a test of the functions with 1000 plays with/without using the switching strategy. Don’t forget to put your github repo at the top of your R markdown!

MAKE SURE YOU SET A SEED SO THE WORK IS REPRODUCIBLE!

Rubric:

Code Style (5 points) Is code organized well and commented?

Submission (5 points) Was the lab submitted as an html document on Canvas? Does the html document contain a link for a GitHub repository that contains your R package?

Function (30 points) Were the directions in the lab followed to create a function that simulates the Monty Hall problem?

Simulation (10 points) Did you set a seed for the simulation? Did you show your functions working for switch = TRUE and switch = FALSE