

Applied Statistical Programming - Spring 2022

Alex Avery

Problem Set 3

Due Wednesday, March 2, 10:00 AM (Before Class)

Instructions

1. The following questions should each be answered within an R script. Be sure to provide many comments in the script to facilitate grading. Undocumented code will not be graded.
2. Work on git. Fork the repository found at <https://github.com/johnsontr/AppliedStatisticalProgramming2022> and add your code for Problem Set 3, committing and pushing frequently. Use meaningful commit messages because these will affect your grade.
3. You may work in teams, but each student should develop their own Rmarkdown file. To be clear, there should be no copy and paste. Each keystroke in the assignment should be your own.
4. For students new to programming, this may take a while. Get started.

Let's Make a Deal¹

In the game show “Let's Make a Deal”, the candidate gets to choose one of three closed doors, and receives the prize behind the door they choose. Behind one door is a new car; behind the other two doors are goats. After the contestant selects one of the 3 doors, the host opens one of the other two doors, and reveals a goat. Now, the candidate has the option of either sticking with the door they originally selected, or switching to the only other door that is still closed. What should the candidate do, and why? What are the probabilities of winning the car if they stay versus if they switch? This question is known as the Monty Hall Problem.

Your tasks

For this problem set, you will not solve the Monty Hall Problem, but you will have to code a slightly simplified version of the “Let's Make a Deal” game. More specifically, you will set up a new class, which contains information regarding the door a player chooses, and a method that simulates a modified version of the game. You will have to do this using the S3 class system. Here are the specific instructions:

1. Define a new class: `door`. Objects of this class simply take on one numeric value: 1, 2, or 3 – indicating which door a candidate chooses.
2. Create a method for `door` objects that is called `PlayGame`. This method is supposed to do the following:
 - take the numeric value that is stored in the `door` object,
 - draw a random number between 1 and 3 that presents the door behind which the car is hidden,
 - compare the two numbers, and print a message congratulating a winning candidate that chose the correct door, or expressing sympathies for a losing candidate that chose the wrong door.
3. Write:

¹https://en.wikipedia.org/wiki/Let's_Make_a_Deal

- a construction function that allows the user to create a `door` object,
- and a validation function that checks whether the value stored in `door` is actually an integer

##CONSTRUCTOR FUNCTION##

#I am first creating the function that allows the player to choose a door.

```
choose_door <- function(x){
  if (x != 1 & x != 2 & x != 3){
    stop("Selected door must be 1, 2, or 3.") #There are only three doors, so the if
    ↪ statement stops the function if the player chooses a door outside of 1 through 3.
    ↪
  }
  selected_door <- as.integer(x) #saving the door chosen by the player into
  ↪ "selected_door" as an integer
  class(selected_door) <- "door" #assigning the selected door as the "door" class
  return(selected_door) #returning selected door
}
```

#let's test our constructor function:

```
choose_door(2)
```

```
## [1] 2
## attr(,"class")
## [1] "door"
```

```
choose_door(4)
```

```
## Error in choose_door(4): Selected door must be 1, 2, or 3.
```

##VALIDATOR FUNCTION##

```
validate_door <- function(door){
  if(is.integer(door) == "TRUE" & class(door) == "door") #the input must be a integer and
  ↪ also of the class "door"
  {
    return(door)
  } else{
    stop("Please use choose_door function to select a door")
  }
}
```

#This validator function will be fed into our play game function.

#creating a generic method for the function "play_game"

```
play_game <- function(x){
  UseMethod("play_game")
}
```

#creating our "play_game" function for the class "door".

```
play_game.door <- function(door){
  door <- validate_door(door) #Here we validate the input using our validator function
  ↪ we created earlier
  car_door <- sample(1:3, 1) #The winning door is selected from a random sample of doors
  ↪ 1 through 3.
  if(door == car_door) #here we compare the value of the input door with the car door
  ↪ sample.
```

```
{  
  print("Congrats! You won a car!")  
}  
else{  
  print("Wrong Answer! Better luck next time!")  
}  
}
```

```
##TEST##
```

```
#let's test our game!
```

```
Alex_door <- choose_door(3)  
play_game(Alex_door)
```

```
## [1] "Congrats! You won a car!"
```