Applied Statistical Programming - Spring 2022

Cecilia Sui

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 Deal[1]

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,

---

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

- 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:

- 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

```r
# Create Constructor
new_door <- function(door_num){
  # check user input
  if (!is.integer(door_num)){
    door_num <- as.integer(door_num)
    warning(sprintf("Input is not an integer. \n
                    Converted input to integer: %s", door_num))
  }

  # create object
  output <- list(chosen = door_num,
                 winning = FALSE)

  # change class to door
  class(output) <- "door"
  # return object
  return(output)
}


# Validator
validate_door <- function(x){
  # constructor already takes care of integer type,
  # so we do not need to check for the data type

  # Check for integer value
  if (!(x$chosen %in% c(1L, 2L, 3L))){
    stop("Door number must be either 1, 2, or 3.")
  }

  return(x)
}


# Create method PlayGame
PlayGame.door <- function(x) {
  # validate parameter object
  validate_door(x)

  # choose door number with the car
  car <- sample(c(1L, 2L, 3L), 1)
  print(paste("The winning car is behind door:", car))

  # update object
  if (car == x$chosen) {
    x$winning <- TRUE
```

```r
  }

  # print appropriate messages
  print(ifelse(x$winning,
               "Congratulations! You have won the car!",
               "Sorry! You chose the wrong door. Maybe try again? "))

}



# Testing
# set seed
set.seed(123)

# Create an object
mydoor <- new_door(3.5)
```

```
## Warning in new_door(3.5): Input is not an integer.
##
##                        Converted input to integer: 3
```

```r
mydoor$chosen
```

```
## [1] 3
```

```r
mydoor$winning
```

```
## [1] FALSE
```

```r
# validate the object
validate_door(mydoor)
```

```
## $chosen
## [1] 3
##
## $winning
## [1] FALSE
##
## attr(,"class")
## [1] "door"
```

```r
# play a game
PlayGame.door(mydoor)
```

```
## [1] "The winning car is behind door: 3"
## [1] "Congratulations! You have won the car!"
```