Applied Statistical Programming - Spring 2022

Amaan Charaniya

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

```
# Question 1
door <- 1  #this is storing a numeric value for door
class(door) <- "door"  #Then assigning it a class 'door'

# Question 2 Here I'm just creating a generic method for the PlayGame function
PlayGame <- function(door) {
    UseMethod("PlayGame")
}

# This is the specific method for a door class It takes an input of a door that
# I defined above
PlayGame.door <- function(door) {
    car_door <- sample(c(1, 2, 3), 1, replace = TRUE)  #It samples a number and assigns it to the car
    if (door == car_door) {
        # First we check if our given door is the car door
        print("Congrats you won a new car!")  #Print a nice encouraging message if it s
    } else {
        print("You are a loser.")  #Crush spirits otherwise, sympathies make the candidate weak.
    }
}
set.seed(1000)
# I tested my method here with the door I created above. Below I create other
# doors and test as well.
PlayGame.door(door)
```

```
## [1] "You are a loser."
```

```
# Question 3 Here I am defining a new function called 'new_door' that accepts a
# single value
new_door <- function(number) {
    door <- number  #I'm storing the imputed number as an object defined in this function
    class(door) <- "door"  #This is my assignment of the class 'door' to this new object
    validate <- function(door) {
        # I added a validation function within this to save time. I could've
        # also written this separately. I opted to check if the imputed number
        # was one of our accepted door values, if not I'm throwing an error.
        if (!(door %in% c(1, 2, 3))) {
            stop("Please enter 1, 2, or 3 as a value for this door")
        }
    }
    validate(door)  #just running the validation function
    return(door)  #returning the output which should include the door value and class 'door'
}

door <- new_door(1)
door
```

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

```
PlayGame.door(new_door(2))
```

```
## [1] "Congrats you won a new car!"
```

```
PlayGame.door(new_door(3))
```

```
## [1] "Congrats you won a new car!"
```