Applied Statistical Programming - Spring 2022

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.

\section*{Let's Make a Deal\footnote{\url{https://en.wikipedia.org/wiki/Let's_Make_a/_Deal}}}

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.

\subsection*{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:
   - 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

```
# 1. Create a function that choose a door. Input= the door it is selected.
# Output= the door selected with a class. Create the function
door <- function(choice) {
    door_choice <- list(door_selected = choice)  #Store in an object a list the choice of door selected
    class(door_choice) <- "door"  #Store the attribute class to the door choice.
    return(door_choice)  #Return the door choice.
}

# Test the first function
door(3)
```

```
## $door_selected
## [1] 3
##
## attr(,"class")
## [1] "door"
```

```
# 2 Create a method, this one is for door object. The argument is the door
# choice Generic function:
PlayGame <- function(door) {
    UseMethod("PlayGame")
}
# Method function:
PlayGame.door <- function(door) {
    door_selected <- door$door_selected  #Store the door selected in the first function
    car <- sample(1:3, 1)  #Choose a number of door for the car prize
    if (car == door_selected) {
        # Print Congrats in case is the same door
        print("Congrats you just won a new CAR!")
    } else {
        # Print Sorry in case is different door to the one which has the prize
        print("Sorry, you loose. Keep trying")
    }
}
# Test the second function
PlayGame.door(door(2))
```

```
## [1] "Sorry, you loose. Keep trying"
```

```
# 3 This function is the validator. This function is going to create an object
# only if the door object is a 1,2, or 3. The output will be the door with the
# class.

new_door <- function(door_choice) {
    if (!(door_choice %in% c(1, 2, 3))) {
        # In case the selection is not between 1,2,3, it will print an error
        stop("Error: You need to choose a door between 1 to 3")
    }
    return(door(door_choice))
}

# Test the third function
new_door(2)
```

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

```
# Test the game:
set.seed(1234)
PlayGame.door(new_door(2))
```

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