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.

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,
    - 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] https://en.wikipedia.org/wiki/Let's_Make_a_Deal

```r
# CONSTRUCTOR AND VALIDATOR ------------------------------------------------

# Constructor
new_door <- function(x) {
  # Just in case someone does not input an integer. Probably could run the
  # validate function within this function to make things smoother.
  if (!is.integer(x)) {
    x <- as.integer(x)
    warning(paste("Input was not an integer, converting to the integer", x))
  }

  class(x) <- "door"
  return(x)
}

# Validator
validate_door <- function(x) {
  if (!is.integer(x)) {
    stop("Choice is not an integer, please input an integer")
  }

  if (!(x %in% c(1L, 2L, 3L))) {
    stop("Choice must be an inetger between 1 and 3")
  }

  return(TRUE)
}

# METHOD -------------------------------------------------------------------

# General Method
PlayGame <- function(x) {
  UseMethod("PlayGame")
}

# Class Method
PlayGame.door <- function(x) {

  # Validate the input before running code
  validate_door(x)

  # Made the random door choose from integers just in case some numerical
  # error could show up otherwise
  winner <- sample(c(1L, 2L, 3L), 1)

  if (x == winner) {
    print("Congratulations, you won!")
  } else {
    print("Sorry, better luck next time!")
  }
}

# create consistent results
```

```r
set.seed(12345)

# Testing functions
test_1 <- new_door(1L)
test_2 <- new_door(3L)

new_door(1)
```

```
## Warning in new_door(1): Input was not an integer, converting to the integer 1
```

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

```r
validate_door(test_1)
```

```
## [1] TRUE
```

```r
# The first one should have the losing message and the second one should have
# the winning message
PlayGame(test_1)
```

```
## [1] "Sorry, better luck next time!"
```

```r
PlayGame(test_2)
```

```
## [1] "Congratulations, you won!"
```