ETC 2420/5242 Lab 3 2016

Di Cook

SOLUTION

Question 1 (3 pts)

Here is a function that will randomly sample one value from a choice of {`pig`, `pig`, `car`}

```
ppc_sample <- function(){
  sample(c("pig","pig","car"), 1)
}
```

- Run this function 20 times and report the proportion of times that `car` is selected. (You could put this in a loop and compute the proportion of times `car` shows.) For me it was 8 times.
- How many times would you expect `car` to be selected out of 20 runs? 6.67, so between 6 and 7 times
- Modify the function so that it takes one argument, which will be the player's initial choice of door.

```
ppc_choice <- function(choice=1){
  doors <- sample(c("pig","pig","car"))
  doors[choice]
}
```

Question 2 (2 pts)

- Modify your function so that it shows the player the door that the host selects.

```
ppc_host <- function(choice=1) {
  doors <- sample(c("pig","pig","car"))
  notchosen <- c(1:3)[-choice]
  if (doors[choice] == "pig") {
    if (doors[notchosen[1]] =="pig")
      host_choice = paste(doors[notchosen[1]],
          paste("door", notchosen[1], sep=" "), sep="-")
    else
      host_choice = paste(doors[notchosen[2]],
          paste("door", notchosen[2], sep=" "), sep="-")
  }
  else {
    d <- sample(notchosen, 1)
    host_choice = paste(doors[d], paste("door", d, sep=" "), sep="-")
  }
  return(host_choice)
}
```

Question 3 (2 pts)

- Modify your function so that it includes the player strategy as an input, and returns the appropriate result.

```r
ppc_play <- function(choice=1, strategy="switch") {
  doors <- sample(c("pig","pig","car"))
  notchosen <- c(1:3)[-choice]

  host_choice <- sample(notchosen, 1) # player choice is car
  if (doors[choice] == "pig") { # player chice is pig
    host_choice <- ifelse (doors[notchosen[1]] == "pig",
            notchosen[1], notchosen[2])
  }

  # Now employ strategy
  final_door <- doors[choice] # Strategy is to stay with original choice
  if (strategy == "switch") {
    d <- notchosen[which(notchosen != host_choice)]
    final_door <- doors[d]
  }

  return(final_door)
}
```

Question 4 (2 pts)

- Run your function 100 times, for each strategy and report the proportion of times that the player wins the car. (You might want to do this in a loop to automate it. Also, if you have had trouble writing your function, you can use the code at the bottom of this Rmd document, which simulates the game - it can be found on the web, too.)

For the stay strategy, the result was

```r
ncar <- 0
for (i in 1:100) {
  if (ppc_play(strategy="stay") == "car")
    ncar <- ncar+1
}
ncar/100
# [1] 0.34
```

And for the switch strategy, the result is

```r
ncar <- 0
for (i in 1:100) {
  if (ppc_play() == "car")
    ncar <- ncar+1
}
ncar/100
# [1] 0.7
```

Question 5 (1 pt)

- What is different about Monty Hall, and the popular TV game show "Deal or No Deal"? Players are given the choice to switch cases, if they play to the very end of the game. Do they get an advantage if they switch cases?

The difference between the games is that in Monty Hall the host knows what is behind the door, and with Deal or No Deal the host does not know what is in the case. It is advantageous to switch in Monty Hall because the host is the person who decides what to show you, and they won't show the door with the prize. In Deal or No Deal, there is no difference in outcome whether you switch or stay, both strategies have the same chance of getting the bigger prize.