Monty Hall Problem

Cameron Bale

1/25/2019

Load library.

```
library(tidyverse)

## -- Attaching packages ---------------------------------------------------------
------------ tidyverse 1.2.1 --

## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.3.1     v forcats 0.3.0

## -- Conflicts ------------------------------------------------------------------
------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

"Suppose you're on a game show, and you're given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say No. 1, and the host, who knows what's behind the other doors, opens another door, say No. 3, which has a goat. He then says to you, 'Do you want to pick door No. 2?' Is it to your advantage to take the switch?"

This is known as the Monty Hall Problem. Lets use a simulation study to decide whether it's better to switch your door or stick with your first choice.

First, let's set up a simple case: playing the game once where you keep your original choice.

```
# set prizes (you want to get 1)
prizes <- c(1, 0, 0)

# player chooses randomly
choice <- sample(prizes, 1)

# set the other door to 0 or 1 depending on players choice. This assumes the
host would only
# reveal a door with a goat behind it
other_door <- if_else(choice == 1, 0, 1)

# create dataframe with the players choice and the other door
data_frame(
  Your_Choice = choice,
  Other_Door = other_door)
```

```
## # A tibble: 1 x 2
##   Your_Choice Other_Door
##         <dbl>      <dbl>
## 1           1          0
```

Now we have simple code that simulates choosing 1 of 3 doors, the dealer removes one of the doors, and you stick with your original choice. All we have to change for the player to switch their choice is whether 'Your_Choice' in the final data_frame is assigned to the players 'choice' or to the 'other_door'.

Let's use the code to create a function that simulates playing the game.

```
# create playing function where you specify your strategy: 'keep' or 'switch'
playing <- function (keep_or_switch) {

  # set up prizes
  prizes <- c(1, 0, 0)

  # set up player's choice
  choice <- sample(prizes, 1)

  # set up the other door
  other_door <- if_else(choice == 1, 0, 1)

  # if the player keeps, show outcome in dataframe
  if (keep_or_switch == 'keep') {

    outcome <- data_frame(
      Your_Choice = choice,
      Other_Door = other_door)

    return(outcome)

  # if the player switches, show outcome in dataframe
  } else if (keep_or_switch == 'switch') {

    outcome <- data_frame (
      Your_Choice = other_door,
      Other_door = choice)

    return(outcome)

  # if invalid strategy input, give warning
  } else {

    warning("You must choose to 'keep' or 'switch' your door choice.")

  }
}
```

Let's try it out.

```
playing('switch')

## # A tibble: 1 x 2
##   Your_Choice Other_door
##         <dbl>      <dbl>
## 1           1          0
```

Now let's play the game 10,000 times using the 'keep' and the 'switch' strategy and see which one yields more wins.

```
# set number of simulations
n_sims <- 1:10000

# simulate 10,000 games under each strategy
keeping <- map_df(n_sims, function (x) playing('keep'))
switching <- map_df(n_sims, function (x) playing('switch'))

# calculate the probability of winning under each strategy
prob_win <- c(mean(keeping$Your_Choice), mean(switching$Your_Choice))

# create dataframe with strategy, probability of winning, and upper/lower
confidence bounds to
# quantify uncertainty
data_frame(Strategy = c('Keep', 'Switch'), Prob_of_Win = prob_win) %>%
  mutate(Lower = Prob_of_Win - qnorm(0.975) * sqrt(Prob_of_Win * (1 -
Prob_of_Win) / length(n_sims)),
         Upper = Prob_of_Win + qnorm(0.975) * sqrt(Prob_of_Win * (1 -
Prob_of_Win) / length(n_sims)))

## # A tibble: 2 x 4
##   Strategy Prob_of_Win Lower Upper
##   <chr>          <dbl> <dbl> <dbl>
## 1 Keep           0.333 0.324 0.342
## 2 Switch         0.666 0.657 0.676
```

The data is in: you should switch your door. The above table shows the estimate of the probability of winning under each strategy as well as lower and upper bounds on that probability to quantify the uncertainty of that estimate. Even with the uncertainty, if you switch your door you will win about 2/3 of the time.