Data Frames in R: Analyzing the Monty Hall Problem

Preparation

The Monty Hall Problem is a famous probability puzzle that challenges our intuition about statistics and decision-making. Named after Monty Hall, the host of the American television game show "Let's Make a Deal," this problem presents a scenario where a contestant must choose between three doors. Behind one door is a prize, while the other two conceal goats. After the contestant makes an initial choice, the host, who knows what's behind the doors, opens another door to reveal a goat. The contestant is then given the option to stick with their original choice or switch to the remaining unopened door. The question is: which is the best strategy?

Before we dive into working with R, let's gain an illustrative understanding of the problem. Watch the video below from the 1994 German television show "Geh aufs Ganze," where a contestant faces this very dilemma.

- Video "Geh aufs Ganze" (1994)

To learn about data frames in R, watch the lecture on representing data from Harvard's CS50 Introduction to Programming with R, as well as the short video on data frames:

- Video CS50R - Representing Data
- Video CS50R - Short on Data Frames

Now you should be prepared to solve the following exercises using R and RStudio.

Exercise 1: Data as Tables

Data frames represent data in a tabular format, similar to a spreadsheet. Each row corresponds to an observation, while each column represents a variable. Data frames are a fundamental data structure in R, allowing you to store, manipulate, and analyze data efficiently. Let's revisit the Monty Hall Problem using data frames.

1. Load the historical data from the game show "Let's Make a Deal" into a data frame in R. Use the provided CSV file as a source.
2. Review the structure of the data frame. Use suitable functions to get an overview of the data and answer the following questions:
   - How many rows and columns are there in the data frame?
   - What are the data types of each column?
3. Access specific columns of the data frame:
   - Extract the `prize_door` column using both the `$` operator and the `[]` notation!
   - What are the similarities and differences between the two approaches?
   - How can you extract multiple columns from a data frame?
4. Create a new column in the data frame named `won` that indicates whether the contestant won the prize. Use the following logic: If the contestant stayed and the `prize_door` matches the `contestant_choice`, or if they switched and the doors did not match, they won.
5. Filter the data frame to create a new data frame that contains only the games where the contestant won. Use the `won` column to filter.
   - How many games did the contestant win when they decided to switch?
   - How many did they win when they decided to stay?
6. Change the data type of the `decision` column to a factor and explore its levels! Why might it be useful to store categorical data as factors?
7. Access rows and columns using index notation:
   - Extract the first 10 rows of the data frame.
   - Extract the last 10 rows of the data frame.
   - Extract all columns except the last one.
8. Sort the data frame by the `prize_door` column in both ascending and descending order! What changes do you observe in the sorted data frames?
9. Reflect on the Monty Hall Problem:
   - Based on the data, which strategy (switching or staying) seems to be more successful?
   - How does the result align with your initial intuition about the problem?

Exercise 2: Create Data Frames from Scratch

In this exercise, you will conduct your own experiments with the Monty Hall Problem and create a data frame to store your results. Follow the steps below:

1. Simulate your own Monty Hall games by choosing a prize door, a contestant choice, and whether the contestant decides to switch or stay. Conduct at least 10 simulations.
2. Create a data frame using the `data.frame()` function to store your experimental data. Include the following columns:
    - `prize_door`: the door that has the prize (1, 2, or 3)
    - `contestant_choice`: the door chosen by the contestant initially (1, 2, or 3)
    - `decision`: whether the contestant decides to "stay" or "switch"
    - `won`: whether the contestant won the prize (TRUE or FALSE)
3. Perform operations on your data frame:
    - Change the data type of the `decision` column to a factor.
    - Add a new column called `game_number` that uniquely identifies each game (e.g., 1, 2, 3, ...).
4. Save your data frame to a CSV file using the `write.csv()` function. Make sure to specify the file name and set `row.names` to FALSE.
5. Reflect on your experiment:
    - Did you notice a trend in your results? Which strategy seems to be more successful?
    - How do your experimental results compare to the historical data from Exercise 1?

Exercise 3: Simulation Results as Data Frame

In this exercise, you will modify a simulation of the Monty Hall Problem that we introduced in class so that the results of each simulation run are stored as a row in a data frame. Follow the steps below:

1. Modify the Monty Hall simulation code to collect the following information as vectors for each run:
    - `prize_door`: the door with the prize (1, 2, or 3)
    - `contestant_choice`: the door chosen by the contestant initially (1, 2, or 3)
    - `decision`: whether the contestant decided to "stay" or "switch"

- `won`: whether the contestant won the prize (TRUE or FALSE)

2. Once the simulation finishes, create a data frame from the vectors to store the results of the simulation. Each row should represent a single run of the simulation.
3. Save the data frame to a CSV file using the `write.csv()` function. Be sure to specify the file name and set `row.names` to FALSE.
4. Run the simulation multiple times (e.g., 1000 times) and reflect on the results:
    - What trends do you observe when comparing the results of switching versus staying?
    - How do your simulation results compare with the theoretical probabilities discussed in class?