

Homework Assignment 3

STA 141A

Due Tuesday, May 22 by 5:00 pm

Description

In this assignment, you will be developing a simulation of the board game *Monopoly*. In part one, you will develop the following game features:

1. Dice rolls
2. Board Position
3. Going to Jail
4. Drawing cards from Chance and Community Chest

In part two, you will add more features to your existing function. These include:

1. Collecting money for passing "GO".
2. Losing money for landing on "T1" or "T2".
3. Losing money for landing on a set of colored properties.

More details on *Monopoly* can be found in the **Requirements** section of this homework. **Please read this section carefully.** Unlike the last assignment, you will have to do the problems sequentially. A good strategy is to try to build one feature at a time (e.g. make a working deck of Community Chest cards) and then combine other features later on. Don't be afraid to write test code and explore ideas in the console. Try to test the functions you write on specific use cases to verify the integrity of your game. Good luck!

Part 1: Working with the Board

Use R to find answers to all of the following questions (that is, don't do any by hand or by point-and-click). Save your code in an R script.


1. Write a function `simulate_monopoly()` that simulates n turns by a player in a game of Monopoly using two d -sided dice. The inputs to your function should be n and d . The output of your function should be a length $n + 1$ vector of positions, encoded as numbers from 0 to 39.
2. Write a function `estimate_monopoly()` that uses your simulation to estimate the long-term probabilities of ending a turn on each Monopoly square. What are the 3 most likely squares to end a turn on if you play Monopoly with 6-sided dice? What if you play with 4-sided dice? Display graphically the long-term probabilities for 3, 4, 5, and 6-sided dice.
3. Use $k = 1,000$ simulations with $n = 10,000$ rolls each to estimate the standard error for the long-term probability of ending a turn in jail. Use the standard amount of dice rolls, $d = 6$.

Part 2: Working with Money

1. Write an updated version of the `simulate_monopoly()` function called `simulate_monopoly2()`. The function should now take in numeric vectors `property` and `rent` as additional inputs. The `property` represents the indices at which a player **loses money** if they land on them. The `rent` represents the associated rent that a player pays for landing on each corresponding property. Furthermore, the

function should return the $n + 1$ board positions (coded 0-39) and the money gained or lost after each dice roll.

2. Run `simulate_monopoly2()` with $n = 100$ dice rolls and $k = 1000$ simulations for all eight colors. After each simulation is done, sum the money change at each turn to get the **total money gained/lost** per simulation. Which color seems to be the most effective when it has hotels on it? Which color is the least effective? Provide at least one informative visual showing in your report. You may add more graphics if you desire. To save time for this question, use the `properties.csv` dataset from Canvas. For more information on the features of this data, see the **Requirements** section.

3.  Now simulate a basic version of a two player game of *Monopoly*.

In this game, each player will be equipped with hotels on a specific color of properties. There will be no bankruptcy conditions (if a player goes below \$0 the game does not halt), instead, the player with the most money at the end of n dice rolls will be deemed the winner. Whenever a player lands on the opposing player's hotel, they will now **transfer the money** to the other player at that turn. Let w_i be the change in money at turn i , then the final calculation of a player's wealth is:

$$\text{Total Money} = \text{Initial Money} - \text{Hotel Costs} + \sum_{i=1}^n w_i \quad (1)$$

For each simulation, set the initial money to be \$5000. Use the hotel cost for a set of properties from `properties.csv`.

For all 28 pairwise combinations of colors, run $k = 100$ head-to-head matchups. Do this for $n = \{25, 50, 100\}$ dice rolls. Display the ordered standings with wins, losses, and win percentage for each value of n . No graphics are required here but are encouraged for motivated students. The pairs have been generated for you on Canvas to use, please see `color_combos.csv`.

Assemble your answers into a report. Please do not include any raw R output. Instead, present your results as neatly formatted¹ tables or graphics, and write something about each one. You must **cite your sources**. Your report should be **no more than 5 pages** including graphics, but excluding code and citations.

Requirements

Part 1.

The game Monopoly is played on a board with 40 squares, set up as shown below.

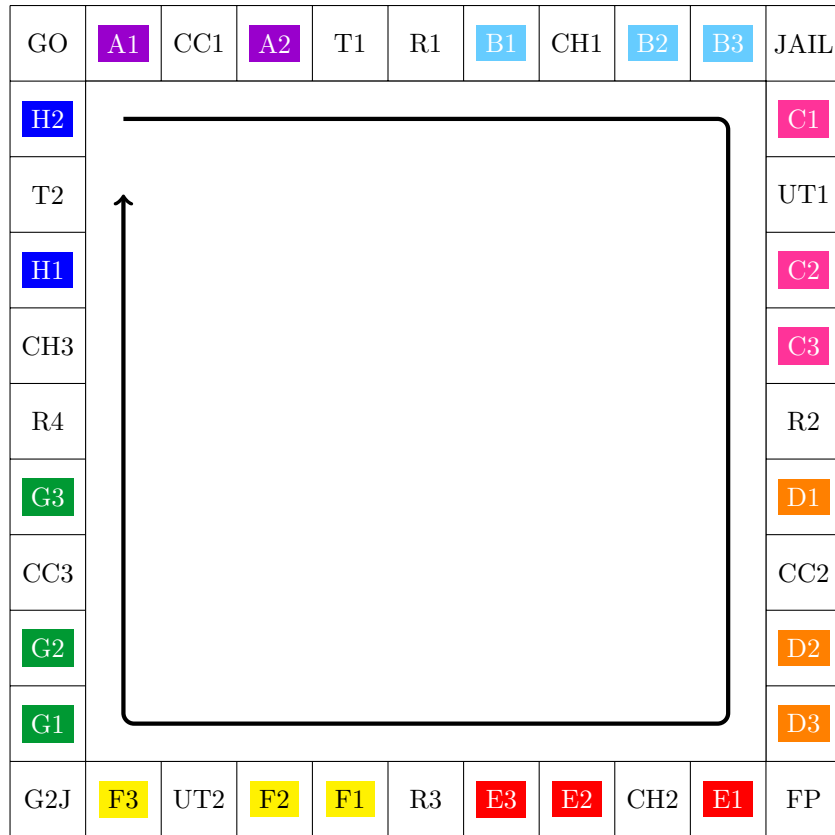
Players begin the game on the GO square. On each turn, the player rolls two 6-sided dice. The sum of the dice determines the number of squares they advance (in a clockwise direction) on that turn.

Without any further rules, we would expect players to visit each square with equal probability. However, landing on G2J (go to jail), CC (community chest), and CH (chance) changes this distribution. When a player lands on G2J, they must immediately move to the JAIL square.

At the beginning of the game, the CC and CH cards are shuffled. When a player lands on CC or CH they take a card from the top of the respective pile and, after following the instructions, it is returned to the bottom of the pile. There are 16 cards in each pile, but for this assignment we are only concerned with cards that order a movement. Any card not concerned with movement will be ignored and the player will remain on the CC/CH square. The relevant cards are:

- Community Chest (2/16)
 1. Advance to GO
 2. Go to JAIL

¹See the graphics checklist on Canvas.



- Chance (10/16)
 1. Advance to GO
 2. Go to JAIL
 3. Go to C1
 4. Go to E3
 5. Go to H2
 6. Go to R1
 7. Go to next RR (railroad)
 8. Go to next RR
 9. Go to next UT (utility)
 10. Go back 3 squares

In addition to G2J and CC/CH, if a player rolls doubles on three consecutive turns, they proceed directly to jail. A “double” is a roll where the two dice are equal, such as rolling 2 fives or 2 sixes.

Part 2.

In part 2, you are required to add a money component to the game. You will take your original `simulate_monopoly()` function and add the following features to it, making `simulate_monopoly2()`

- If a player passes or lands on the “GO” square, they will collect \$200. You do not need to make the player collect \$200 if they draw a card that places them across the board.
- If a player goes to jail by the end of the turn, they should not collect “GO” money.

- If a player lands on the income tax square ("T1"), they will lose \$200.
- If a player lands on the luxury tax square ("T2"), they will lose \$100.
- `simulate_monopoly2()` should take in a vector that includes specific property positions.
- Each property should have a corresponding "rent" for landing on it. In the spirit of *Monopoly*, the associated cost with each property is how much a player would pay as if there was a hotel on the property. So if there are hotels on {"H1", "H2"} for example, then the player would lose \$1500 if they landed on "H1" and \$2000 if they landed on "H2".
- You should include a wealth counter that calculates how much money the player gained or lost each turn.
- **Hint:** You do not need to include a player's money at the beginning, since you can add it after an n -roll simulation has ended.
- **Hint:** For question 2.3, it would be a good idea to add a variable in `simulate_monopoly2()` that has the following properties:

$$X_i = \begin{cases} 1 & \text{If player landed on a hotel property in turn } i \\ 0 & \text{Otherwise.} \end{cases}$$

- **Hint:** For 2.3, it would be best to simulate two concurrent instances of `simulate_monopoly2()`.

For questions 2.2 and 2.3, you're given two dataset. One of which being `properties.csv` that outlines the details of the monopoly properties, and the other being `color_combos.csv` which is useful for the pairwise simulations. The details of `properties.csv` is below:

- **Color** The color of the property.
- **Position** The position of the property.
- **Rent** How much money a player loses by landing on the property if they **do not own the hotel**. Correspondingly, this number represents how much money the player **who owns the hotel** gains.
- **Cost** The associated cost of building hotels on all properties of a specific color. (e.g. It costs \$2000 to build hotels on **both** "H1" and "H2")

What To Submit

Email a digital copy to spring18stat141a@gmail.com. The digital copy must contain your report (as a PDF) and your code (as one or more R scripts).

Additionally, submit a printed copy to the box in the statistics department office². The printed copy must contain your report and your code (in an appendix). Please print double-sided to save trees. It is your responsibility to make sure the graphics are legible in the printed copy!

Relevant Functions

All of the functions from previous assignments, as well as:

`seq()`, `seq.int()`, `rep()`, `sample()`, `replicate()`, `ifelse()`, `unlist()`, `do.call()`, `cbind()`, `rbind()`

Relevant Packages

The entire assignment can be completed in base R with ease, but you may use outside packages for making the visuals. Additionally, you can take a look into using `beepR` for producing sound. It may be useful to have a sound play when a large simulation is completed.

²4th floor of Mathematical Sciences Building

Domain Knowledge

[https://en.wikipedia.org/wiki/Monopoly_\(game\)](https://en.wikipedia.org/wiki/Monopoly_(game))