

# Monopoly

Yuki Kitamura

2024-07-16

This is a simulation of a classic board game, Monopoly. The goal is to find out which spaces on the board get landed on the most. I will not simulate the entire game. I will simulate only the movement of pieces, and will keep track of which squares the pieces land on.

```
source("Monopoly Script.R") # edit with your file name
print(Player)
```

```
## <Player> object generator
##   Public:
##     pos: 1
##     verbose: TRUE
##     count: 0
##     chance_n: 0
##     chest_n: 0
##     jail_n: 0
##     move_fwd: function (n)
##       chance: function (draw, card)
##       chest: function (draw, card)
##       double: function (rolls)
##       jail: function (rolls)
##       end: function ()
##     initialize: function (verbose = FALSE, pos = 1)
##     clone: function (deep = FALSE)
##   Parent env: <environment: R_GlobalEnv>
##   Locked objects: TRUE
##   Locked class: FALSE
##   Portable: TRUE
```

```
print(take_turn)
```

```
## function (player, spacetracker)
## {
##   dice_rolls <- dice$roll()
##   player$double(dice_rolls)
##   player$jail(dice_rolls)
##   player$move_fwd(sum(dice_rolls))
##   spacetracker$tally(player$pos)
##   player$chance(chance$draw(), chance$deck[chance$deck_order[chance$position -
##     1], 1])
##   if (player$chance_n != 0) {
```

```

##         spacetracker$tally(player$pos)
##     }
##     player$chest(community$draw(), community$deck[community$deck_order[community$position -
##         1], 1])
##     if (player$chest_n != 0) {
##         spacetracker$tally(player$pos)
##     }
##     double_check <- function(dice_rolls) {
##         if (dice_rolls[1] == dice_rolls[2] & player$jail_n ==
##             0) {
##             if (player$verbose) {
##                 cat("\nPlayer rolled doubles, so they take another turn.\n")
##             }
##             dice_rolls <- dice$roll()
##             player$double(dice_rolls)
##             if (player$count < 3) {
##                 player$move_fwd(sum(dice_rolls))
##                 spacetracker$tally(player$pos)
##                 player$chance(chance$draw(), chance$deck[chance$deck_order[chance$position -
##                     1], 1])
##                 if (player$chance_n != 0) {
##                     spacetracker$tally(player$pos)
##                 }
##                 player$chest(community$draw(), community$deck[community$deck_order[community$position -
##                     1], 1])
##                 if (player$chest_n != 0) {
##                     spacetracker$tally(player$pos)
##                 }
##                 double_check(dice_rolls)
##             }
##             else {
##                 spacetracker$tally(player$pos)
##             }
##         }
##     }
##     if (player$count >= 1) {
##         double_check(dice_rolls)
##     }
##     player$end()
## }

```

## Part 1: Test Cases

Demonstrating that I coded the rules by showing the output of several test cases using preset dice. The output should be very verbose. It should announce the player roles, where the player moves, what spaces get tallies, if they rolled doubles, etc.

### Test Case 1: Space: Go to Jail

```

dice <- PresetDice$new(
  rolls = c(3,4), # Preset the dice rolls

```

```

    verbose = TRUE
  )

  set.seed(16)
  player1 <- Player$new(verbose = TRUE, pos = 24)
  monopoly <- SpaceTracker$new(verbose = TRUE)

  for (i in 1:1) {
    cat("--- Turn", i, "---\n")
    take_turn(player1, monopoly)
    cat("\n")
  }

```

```

## --- Turn 1 ---
## Dice Rolled: 3 4
## Player starts at 24: Indiana Avenue.
## Player moves forward 7.
## Player is now at 31: Go to jail.
## Player goes to jail.
## Added tally to 11: Jail.

```

```

print(setNames(monopoly$counts, 1:40))

```

```

##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
##  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 27 28 29 30 31 32 33 34 35 36 37 38 39 40
##  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

```

**Test Case 2: Chance Card and Doubles Tests: Advance to Go, Reading Railroad, Nearest Railroad, Nearest Utility, No Movement**

```

dice <- PresetDice$new(
  rolls = c(3,4, 4,3, 1,1, 3,4, 5,3),
  verbose = TRUE
)
set.seed(135)
chance <- CardDeck$new(chancedeck, verbose = TRUE)
community <- CardDeck$new(communitydeck, verbose = TRUE)
player1 <- Player$new(verbose = TRUE)
monopoly <- SpaceTracker$new(verbose = TRUE)

for (i in 1:4) {
  cat("--- Turn", i, "---\n")
  take_turn(player1, monopoly)
  cat("\n")
}

```

```

## --- Turn 1 ---
## Dice Rolled: 3 4
## Player starts at 1: Go.

```

```

## Player moves forward 7.
## Player is now at 8: Chance.
## Added tally to 8: Chance.
## Darw a Chance card.
## Card: Advance to Go
## Player moves to 1: Go.
## Added tally to 1: Go.
##
## --- Turn 2 ---
## Dice Rolled: 4 3
## Player starts at 1: Go.
## Player moves forward 7.
## Player is now at 8: Chance.
## Added tally to 8: Chance.
## Darw a Chance card.
## Card: Take a ride on the Reading Railroad
## Player moves to 6: Reading Railroad.
## Added tally to 6: Reading Railroad.
##
## --- Turn 3 ---
## Dice Rolled: 1 1
## Doubles count is now 1.
## Player starts at 6: Reading Railroad.
## Player moves forward 2.
## Player is now at 8: Chance.
## Added tally to 8: Chance.
## Darw a Chance card.
## Card: Advance token to the nearest Railroad
## Player moves to 16: Pennsylvania Railroad.
## Added tally to 16: Pennsylvania Railroad.
##
## Player rolled doubles, so they take another turn.
## Dice Rolled: 3 4
## Player starts at 16: Pennsylvania Railroad.
## Player moves forward 7.
## Player is now at 23: Chance.
## Added tally to 23: Chance.
## Darw a Chance card.
## Card: Advance token to nearest Utility
## Player moves to 29: Water Works.
## Added tally to 29: Water Works.
##
## --- Turn 4 ---
## Dice Rolled: 5 3
## Player starts at 29: Water Works.
## Player moves forward 8.
## Player is now at 37: Chance.
## Added tally to 37: Chance.
## Darw a Chance card.
## Card: Bank pays you dividend of $50

print(setNames(monopoly$counts, 1:40))

## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

```

```
## 1 0 0 0 0 1 0 3 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0
## 27 28 29 30 31 32 33 34 35 36 37 38 39 40
## 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0
```

### Test Case 3: Multiple doubles. Community Chest.

```
dice <- PresetDice$new(
  rolls = c(3,3, 2,2, 2,1, 3,1), verbose = TRUE)

player1 <- Player$new(verbose = TRUE)
monopoly <- SpaceTracker$new(verbose = TRUE)
for (i in 1:2) {
  cat("--- Turn", i, "---\n")
  take_turn(player1, monopoly)
  cat("\n")
}
```

```
## --- Turn 1 ---
## Dice Rolled: 3 3
## Doubles count is now 1.
## Player starts at 1: Go.
## Player moves forward 6.
## Player is now at 7: Oriental Avenue.
## Added tally to 7: Oriental Avenue.
##
## Player rolled doubles, so they take another turn.
## Dice Rolled: 2 2
## Doubles count is now 2.
## Player starts at 7: Oriental Avenue.
## Player moves forward 4.
## Player is now at 11: Jail.
## Added tally to 11: Jail.
##
## Player rolled doubles, so they take another turn.
## Dice Rolled: 2 1
## Player starts at 11: Jail.
## Player moves forward 3.
## Player is now at 14: States Avenue.
## Added tally to 14: States Avenue.
##
## --- Turn 2 ---
## Dice Rolled: 3 1
## Player starts at 14: States Avenue.
## Player moves forward 4.
## Player is now at 18: Community Chest.
## Added tally to 18: Community Chest.
## Draw a Community Chest card.
## Card: Life insurance matures. Collect $100
```

```
print(setNames(monopoly$counts, 1:40))
```

```
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
```

```
## 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0
## 27 28 29 30 31 32 33 34 35 36 37 38 39 40
## 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Test Case 4: Doubles three times. Three turns in jail.

```
dice <- PresetDice$new(
  rolls = c(3,3, 3,3, 3,3, 5,6, 5,6, 5,6),
  verbose = TRUE
)

player1 <- Player$new(verbose = TRUE)
monopoly <- SpaceTracker$new(verbose = TRUE)

for (i in 1:4) {
  cat("--- Turn", i, "---\n")
  take_turn(player1, monopoly)
  cat("\n")
}
```

```
## --- Turn 1 ---
## Dice Rolled: 3 3
## Doubles count is now 1.
## Player starts at 1: Go.
## Player moves forward 6.
## Player is now at 7: Oriental Avenue.
## Added tally to 7: Oriental Avenue.
##
## Player rolled doubles, so they take another turn.
## Dice Rolled: 3 3
## Doubles count is now 2.
## Player starts at 7: Oriental Avenue.
## Player moves forward 6.
## Player is now at 13: Electric Company.
## Added tally to 13: Electric Company.
##
## Player rolled doubles, so they take another turn.
## Dice Rolled: 3 3
## Doubles count is now 3.
## Player goes to jail.
## Added tally to 11: Jail.
##
## --- Turn 2 ---
## Dice Rolled: 5 6
## Player stays in jail.
## Added tally to 11: Jail.
##
## --- Turn 3 ---
## Dice Rolled: 5 6
## Player stays in jail.
## Added tally to 11: Jail.
##
```

```
## --- Turn 4 ---
## Dice Rolled: 5 6
## Player's third turn in jail. Player must exit jail.
## Player exits jail.
## Player starts at 11: Jail.
## Player moves forward 11.
## Player is now at 22: Kentucky Avenue.
## Added tally to 22: Kentucky Avenue.
```

```
print(setNames(monopoly$counts, 1:40))
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
##  0  0  0  0  0  0  1  0  0  0  3  0  1  0  0  0  0  0  0  0  0  1  0  0  0  0
## 27 28 29 30 31 32 33 34 35 36 37 38 39 40
##  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

**Test Case 5:** After going to Jail, player's turn ends immediately. Rolling doubles while in Jail gets player out of jail.

```
dice <- PresetDice$new(
  rolls = c(3,3, 1,2, 3,3, 3,4),
  verbose = TRUE
)

player1 <- Player$new(verbose = TRUE, pos = 25)
monopoly <- SpaceTracker$new(verbose = TRUE)

for (i in 1:3) {
  cat("--- Turn", i, "---\n")
  take_turn(player1, monopoly)
  cat("\n")
}
```

```
## --- Turn 1 ---
## Dice Rolled: 3 3
## Doubles count is now 1.
## Player starts at 25: Illinois Avenue.
## Player moves forward 6.
## Player is now at 31: Go to jail.
## Player goes to jail.
## Added tally to 11: Jail.
##
## --- Turn 2 ---
## Dice Rolled: 1 2
## Player stays in jail.
## Added tally to 11: Jail.
##
## --- Turn 3 ---
## Dice Rolled: 3 3
## In jail but rolled doubles.
## Player exits jail.
```

```
## Player starts at 11: Jail.
## Player moves forward 6.
## Player is now at 17: St. James Place.
## Added tally to 17: St. James Place.
```

```
print(setNames(monopoly$counts, 1:40))
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
##  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  1  0  0  0  0  0  0  0
## 27 28 29 30 31 32 33 34 35 36 37 38 39 40
##  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

## Test Case 6: 20 Predetermined Turns

```
# Using preset dices
dice <- PresetDice$new(
  rolls = c(6,4, 5,3, 3,5, 4,4, 4,4, 2,2, 4,3, 4,4, 1,4,
            3,4, 1,2, 3,6, 5,4, 5,5, 1,2, 5,4, 3,3, 6,1,
            1,1, 2,3, 5,5, 5,4, 4,1, 2,2, 2,4),
  verbose = TRUE
)
set.seed(2)
chance <- CardDeck$new(chancedeck, verbose = TRUE)
community <- CardDeck$new(communitydeck, verbose = TRUE)
```

```
player1 <- Player$new(verbose = TRUE)
monopoly <- SpaceTracker$new(verbose = TRUE)

for (i in 1:20) {
  cat("--- Turn", i, "---\n")
  take_turn(player1, monopoly)
  cat("\n")
}
```

```
## --- Turn 1 ---
## Dice Rolled: 6 4
## Player starts at 1: Go.
## Player moves forward 10.
## Player is now at 11: Jail.
## Added tally to 11: Jail.
##
## --- Turn 2 ---
## Dice Rolled: 5 3
## Player starts at 11: Jail.
## Player moves forward 8.
## Player is now at 19: Tennessee Avenue.
## Added tally to 19: Tennessee Avenue.
##
## --- Turn 3 ---
## Dice Rolled: 3 5
## Player starts at 19: Tennessee Avenue.
```



```

## Player moves forward 8.
## Player is now at 27: Atlantic Avenue.
## Added tally to 27: Atlantic Avenue.
##
## --- Turn 4 ---
## Dice Rolled: 4 4
## Doubles count is now 1.
## Player starts at 27: Atlantic Avenue.
## Player moves forward 8.
## Player is now at 35: Pennsylvania Avenue.
## Added tally to 35: Pennsylvania Avenue.
##
## Player rolled doubles, so they take another turn.
## Dice Rolled: 4 4
## Doubles count is now 2.
## Player starts at 35: Pennsylvania Avenue.
## Player moves forward 8.
## Player is now at 3: Community Chest.
## Added tally to 3: Community Chest.
## Draw a Community Chest card.
## Card: You have won second prize in a beauty contest
##
## Player rolled doubles, so they take another turn.
## Dice Rolled: 2 2
## Doubles count is now 3.
## Player goes to jail.
## Added tally to 11: Jail.
##
## --- Turn 5 ---
## Dice Rolled: 4 3
## Player stays in jail.
## Added tally to 11: Jail.
##
## --- Turn 6 ---
## Dice Rolled: 4 4
## In jail but rolled doubles.
## Player exits jail.
## Player starts at 11: Jail.
## Player moves forward 8.
## Player is now at 19: Tennessee Avenue.
## Added tally to 19: Tennessee Avenue.
##
## --- Turn 7 ---
## Dice Rolled: 1 4
## Player starts at 19: Tennessee Avenue.
## Player moves forward 5.
## Player is now at 24: Indiana Avenue.
## Added tally to 24: Indiana Avenue.
##
## --- Turn 8 ---
## Dice Rolled: 3 4
## Player starts at 24: Indiana Avenue.
## Player moves forward 7.
## Player is now at 31: Go to jail.

```

```

## Player goes to jail.
## Added tally to 11: Jail.
##
## --- Turn 9 ---
## Dice Rolled: 1 2
## Player stays in jail.
## Added tally to 11: Jail.
##
## --- Turn 10 ---
## Dice Rolled: 3 6
## Player stays in jail.
## Added tally to 11: Jail.
##
## --- Turn 11 ---
## Dice Rolled: 5 4
## Player's third turn in jail. Player must exit jail.
## Player exits jail.
## Player starts at 11: Jail.
## Player moves forward 9.
## Player is now at 20: New York Avenue.
## Added tally to 20: New York Avenue.
##
## --- Turn 12 ---
## Dice Rolled: 5 5
## Doubles count is now 1.
## Player starts at 20: New York Avenue.
## Player moves forward 10.
## Player is now at 30: Marvin Gardens.
## Added tally to 30: Marvin Gardens.
##
## Player rolled doubles, so they take another turn.
## Dice Rolled: 1 2
## Player starts at 30: Marvin Gardens.
## Player moves forward 3.
## Player is now at 33: North Carolina Avenue.
## Added tally to 33: North Carolina Avenue.
##
## --- Turn 13 ---
## Dice Rolled: 5 4
## Player starts at 33: North Carolina Avenue.
## Player moves forward 9.
## Player is now at 2: Mediterranean Avenue.
## Added tally to 2: Mediterranean Avenue.
##
## --- Turn 14 ---
## Dice Rolled: 3 3
## Doubles count is now 1.
## Player starts at 2: Mediterranean Avenue.
## Player moves forward 6.
## Player is now at 8: Chance.
## Added tally to 8: Chance.
## Draw a Chance card.
## Card: Advance token to the nearest Railroad
## Player moves to 16: Pennsylvania Railroad.

```

```

## Added tally to 16: Pennsylvania Railroad.
##
## Player rolled doubles, so they take another turn.
## Dice Rolled: 6 1
## Player starts at 16: Pennsylvania Railroad.
## Player moves forward 7.
## Player is now at 23: Chance.
## Added tally to 23: Chance.
## Draw a Chance card.
## Card: Take a ride on the Reading Railroad
## Player moves to 6: Reading Railroad.
## Added tally to 6: Reading Railroad.
##
## --- Turn 15 ---
## Dice Rolled: 1 1
## Doubles count is now 1.
## Player starts at 6: Reading Railroad.
## Player moves forward 2.
## Player is now at 8: Chance.
## Added tally to 8: Chance.
## Draw a Chance card.
## Card: You have been elected Chairman of the Board
##
## Player rolled doubles, so they take another turn.
## Dice Rolled: 2 3
## Player starts at 8: Chance.
## Player moves forward 5.
## Player is now at 13: Electric Company.
## Added tally to 13: Electric Company.
##
## --- Turn 16 ---
## Dice Rolled: 5 5
## Doubles count is now 1.
## Player starts at 13: Electric Company.
## Player moves forward 10.
## Player is now at 23: Chance.
## Added tally to 23: Chance.
## Draw a Chance card.
## Card: Go to Jail
## Player moves to 11: Jail.
## Added tally to 11: Jail.
##
## --- Turn 17 ---
## Dice Rolled: 5 4
## Player stays in jail.
## Added tally to 11: Jail.
##
## --- Turn 18 ---
## Dice Rolled: 4 1
## Player stays in jail.
## Added tally to 11: Jail.
##
## --- Turn 19 ---
## Dice Rolled: 2 2

```

```
## Player's third turn in jail. Player must exit jail.
## Player exits jail.
## Player starts at 11: Jail.
## Player moves forward 4.
## Player is now at 15: Virginia Avenue.
## Added tally to 15: Virginia Avenue.
##
## --- Turn 20 ---
## Dice Rolled: 2 4
## Player starts at 15: Virginia Avenue.
## Player moves forward 6.
## Player is now at 21: Free Parking.
## Added tally to 21: Free Parking.
```

```
monopoly$counts
```

```
## [1] 0 1 1 0 0 1 0 2 0 0 9 0 1 0 1 1 0 0 2 1 1 0 2 1 0 0 1 0 0 1 0 0 1 0 1 0 0 0
## [39] 0 0
```

```
cbind(gameboard, counts = monopoly$counts)
```

##	space	title	counts
## 1	1	Go	0
## 2	2	Mediterranean Avenue	1
## 3	3	Community Chest	1
## 4	4	Baltic Avenue	0
## 5	5	Income Tax	0
## 6	6	Reading Railroad	1
## 7	7	Oriental Avenue	0
## 8	8	Chance	2
## 9	9	Vermont Avenue	0
## 10	10	Connecticut Avenue	0
## 11	11	Jail	9
## 12	12	St. Charles Place	0
## 13	13	Electric Company	1
## 14	14	States Avenue	0
## 15	15	Virginia Avenue	1
## 16	16	Pennsylvania Railroad	1
## 17	17	St. James Place	0
## 18	18	Community Chest	0
## 19	19	Tennessee Avenue	2
## 20	20	New York Avenue	1
## 21	21	Free Parking	1
## 22	22	Kentucky Avenue	0
## 23	23	Chance	2
## 24	24	Indiana Avenue	1
## 25	25	Illinois Avenue	0
## 26	26	B & O Railroad	0
## 27	27	Atlantic Avenue	1
## 28	28	Ventnor Avenue	0
## 29	29	Water Works	0
## 30	30	Marvin Gardens	1
## 31	31	Go to jail	0

## 32	32	Pacific Avenue	0
## 33	33	North Carolina Avenue	1
## 34	34	Community Chest	0
## 35	35	Pennsylvania Avenue	1
## 36	36	Short Line Railroad	0
## 37	37	Chance	0
## 38	38	Park Place	0
## 39	39	Luxury Tax	0
## 40	40	Boardwalk	0

## Part 2: 1000 simulated games

The next part is to run 1,000 simulations of a two-player game that lasts 150 turns. This is a total of over 3 hundred thousand tosses of the dice - 1000 games x 150 turns x 2 players + additional rolls if the player gets doubles. The task is to keep track of where the players land. We ultimately want to build a distribution showing which spaces are most likely to be landed upon. Advance the tokens around the board according to the rules. The special situations involve the cards, jail, and rolling doubles. After 150 turns, reset the game and start over. Simulate 1000 games. The final output will be two tables of the spaces on the board and their frequencies.

```
library(dplyr)
```

```
## Use non-verbose random dice for Part 2
set.seed(2)
chance <- CardDeck$new(chancedeck, verbose = FALSE)
community <- CardDeck$new(communitydeck, verbose = FALSE)
dice <- RandomDice$new()

player1 <- Player$new(verbose = FALSE)
player2 <- Player$new(verbose = FALSE)
monopoly <- SpaceTracker$new(verbose = FALSE)

for (g in 1:2000) {
  if (g % 100 == 0) {
    cat("#### SIMULATING GAME", g, "#### \n")
  }
  for (i in 1:150) {
    take_turn(player1, monopoly)
    take_turn(player2, monopoly)
  }
}
```

```
## #### SIMULATING GAME 100 ####
## #### SIMULATING GAME 200 ####
## #### SIMULATING GAME 300 ####
## #### SIMULATING GAME 400 ####
## #### SIMULATING GAME 500 ####
## #### SIMULATING GAME 600 ####
## #### SIMULATING GAME 700 ####
## #### SIMULATING GAME 800 ####
## #### SIMULATING GAME 900 ####
## #### SIMULATING GAME 1000 ####
```

```
## #### SIMULATING GAME 1100 #####
## #### SIMULATING GAME 1200 #####
## #### SIMULATING GAME 1300 #####
## #### SIMULATING GAME 1400 #####
## #### SIMULATING GAME 1500 #####
## #### SIMULATING GAME 1600 #####
## #### SIMULATING GAME 1700 #####
## #### SIMULATING GAME 1800 #####
## #### SIMULATING GAME 1900 #####
## #### SIMULATING GAME 2000 #####
```

```
print(setNames(monopoly$counts, 1:40))
```

```
##      1      2      3      4      5      6      7      8      9     10     11     12     13
## 20821 14188 14204 14404 15548 19103 15299 15259 15612 15490 81166 18251 18651
##      14     15     16     17     18     19     20     21     22     23     24     25     26
## 15252 17243 17864 19008 18505 19825 19943 19870 18334 19486 18279 21020 19134
##      27     28     29     30     31     32     33     34     35     36     37     38     39
## 17763 17636 18941 17171      0 17657 17099 18159 16519 16051 15044 14462 14339
##      40
## 17547
```

```
results <- cbind(gameboard, tally = monopoly$counts)
results <- cbind(results, rel = monopoly$counts/sum(monopoly$counts))
print(results)
```

```
##      space      title tally      rel
## 1      1      Go 20821 0.02813090
## 2      2 Mediterranean Avenue 14188 0.01916917
## 3      3      Community Chest 14204 0.01919078
## 4      4      Baltic Avenue 14404 0.01946100
## 5      5      Income Tax 15548 0.02100664
## 6      6      Reading Railroad 19103 0.02580974
## 7      7      Oriental Avenue 15299 0.02067022
## 8      8      Chance 15259 0.02061617
## 9      9      Vermont Avenue 15612 0.02109311
## 10     10 Connecticut Avenue 15490 0.02092828
## 11     11      Jail 81166 0.10966200
## 12     12 St. Charles Place 18251 0.02465862
## 13     13 Electric Company 18651 0.02519905
## 14     14      States Avenue 15252 0.02060672
## 15     15      Virginia Avenue 17243 0.02329672
## 16     16 Pennsylvania Railroad 17864 0.02413575
## 17     17 St. James Place 19008 0.02568138
## 18     18      Community Chest 18505 0.02500179
## 19     19 Tennessee Avenue 19825 0.02678522
## 20     20 New York Avenue 19943 0.02694465
## 21     21      Free Parking 19870 0.02684602
## 22     22 Kentucky Avenue 18334 0.02477075
## 23     23      Chance 19486 0.02632720
## 24     24      Indiana Avenue 18279 0.02469645
## 25     25 Illinois Avenue 21020 0.02839976
## 26     26 B & O Railroad 19134 0.02585162
```

```
## 27 27 Atlantic Avenue 17763 0.02399929
## 28 28 Ventnor Avenue 17636 0.02382770
## 29 29 Water Works 18941 0.02559086
## 30 30 Marvin Gardens 17171 0.02319945
## 31 31 Go to jail 0 0.00000000
## 32 32 Pacific Avenue 17657 0.02385607
## 33 33 North Carolina Avenue 17099 0.02310217
## 34 34 Community Chest 18159 0.02453432
## 35 35 Pennsylvania Avenue 16519 0.02231854
## 36 36 Short Line Railroad 16051 0.02168623
## 37 37 Chance 15044 0.02032569
## 38 38 Park Place 14462 0.01953936
## 39 39 Luxury Tax 14339 0.01937318
## 40 40 Boardwalk 17547 0.02370745
```

```
arrange(results, desc(tally))
```

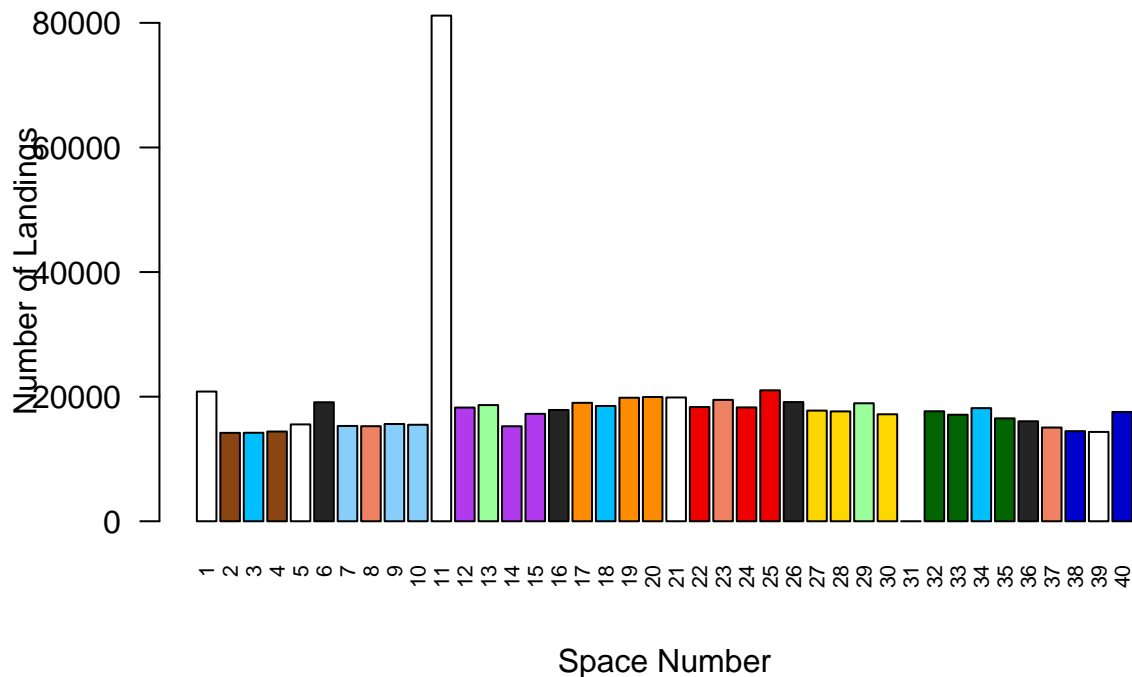
```
## space title tally rel
## 1 11 Jail 81166 0.10966200
## 2 25 Illinois Avenue 21020 0.02839976
## 3 1 Go 20821 0.02813090
## 4 20 New York Avenue 19943 0.02694465
## 5 21 Free Parking 19870 0.02684602
## 6 19 Tennessee Avenue 19825 0.02678522
## 7 23 Chance 19486 0.02632720
## 8 26 B & O Railroad 19134 0.02585162
## 9 6 Reading Railroad 19103 0.02580974
## 10 17 St. James Place 19008 0.02568138
## 11 29 Water Works 18941 0.02559086
## 12 13 Electric Company 18651 0.02519905
## 13 18 Community Chest 18505 0.02500179
## 14 22 Kentucky Avenue 18334 0.02477075
## 15 24 Indiana Avenue 18279 0.02469645
## 16 12 St. Charles Place 18251 0.02465862
## 17 34 Community Chest 18159 0.02453432
## 18 16 Pennsylvania Railroad 17864 0.02413575
## 19 27 Atlantic Avenue 17763 0.02399929
## 20 32 Pacific Avenue 17657 0.02385607
## 21 28 Ventnor Avenue 17636 0.02382770
## 22 40 Boardwalk 17547 0.02370745
## 23 15 Virginia Avenue 17243 0.02329672
## 24 30 Marvin Gardens 17171 0.02319945
## 25 33 North Carolina Avenue 17099 0.02310217
## 26 35 Pennsylvania Avenue 16519 0.02231854
## 27 36 Short Line Railroad 16051 0.02168623
## 28 9 Vermont Avenue 15612 0.02109311
## 29 5 Income Tax 15548 0.02100664
## 30 10 Connecticut Avenue 15490 0.02092828
## 31 7 Oriental Avenue 15299 0.02067022
## 32 8 Chance 15259 0.02061617
## 33 14 States Avenue 15252 0.02060672
## 34 37 Chance 15044 0.02032569
## 35 38 Park Place 14462 0.01953936
## 36 4 Baltic Avenue 14404 0.01946100
```

```
## 37      39      Luxury Tax 14339 0.01937318
## 38       3      Community Chest 14204 0.01919078
## 39       2 Mediterranean Avenue 14188 0.01916917
## 40      31          Go to jail      0 0.00000000
```

```
# set colors for the bar plot
color_vec <- rep(NA, 40)
color_vec[c(2,4)] <- "chocolate4" # mediterranean, baltic
color_vec[c(7,9,10)] <- "lightskyblue" # oriental, vermont, connecticut
color_vec[c(12,14,15)] <- "darkorchid2" # st charles, states, virgina
color_vec[c(17,19,20)] <- "darkorange" # st james, tennessee, new york
color_vec[c(22,24,25)] <- "red2" # kentucky, indiana, illinois
color_vec[c(27,28,30)] <- "gold1" # atlantic, ventnor, marvin
color_vec[c(32,33,35)] <- "darkgreen" # pacific, n. carolina, pennsylvania
color_vec[c(38,40)] <- "blue3" # park place, boardwalk
color_vec[c(6,16,26,36)] <- "gray14" # railroads
color_vec[c(13,29)] <- "palegreen1" # utilities
color_vec[c(8,23,37)] <- "salmon2" # chance
color_vec[c(3,18,34)] <- "deepskyblue" # community chest

barplot(monopoly$counts,
  main = "Barplot of Frequency of Landing for Each Space",
  xlab = "Space Number", ylab = "Number of Landings",
  las = 2, col = color_vec, names.arg = 1:40, cex.names = 0.65)
```

## Barplot of Frequency of Landing for Each Space





```
# this one sets the y-limits so the max count of jail doesn't shrink everything
barplot(monopoly$counts,
  main = "Barplot of Frequency of Landing for Each Space",
  xlab = "Space Number", ylab = "Number of Landings",
  las = 2, col = color_vec, names.arg = 1:40, cex.names = 0.65,
  ylim = c(0, sort(monopoly$counts, decreasing = TRUE)[2]))
```

