Stats 102A - Homework 4 - Output File

Charles Liu (304804942)

Academic Integrity Statement

By including this statement, I, Charles Liu, declare that all of the work in this assignment is my own original work. At no time did I look at the code of other students nor did I search for code solutions online. I understand that plagiarism on any single part of this assignment will result in a 0 for the entire assignment and that I will be referred to the dean of students.

I did discuss ideas related to the homework with Leah Skelton, but I received little help and some ideas of if(...) {...} else(...) {...} statements from her. I did give advice on whether to use Tibble(...) or cbind(...) to Richard (I don't know his last name) and how to correctly use barplot().

```
getwd()
```

```
## [1] "C:/Users/cliuk/Documents/UCLA Works/UCLA Winter 2020/Stats 102A/Homeworks/HW 4"
```

```
setwd("C:/Users/cliuk/Documents/UCLA Works/UCLA Winter 2020/Stats 102A/Homeworks/HW 4")
source(file = "102a_hw_04_output_Charles_Liu.R")  # edit with your file name
```

```
## Warning in file(filename, "r", encoding = encoding): cannot open file
## '102a_hw_04_output_Charles_Liu.R': No such file or directory
```

```
## Error in file(filename, "r", encoding = encoding): cannot open the connection
```

Part 1: 20 Predetermined Turns

```
# You will not simulate the entire game. You will simulate only the movement of pieces,
# and will keep track of which squares the pieces land on.
library(R6)

# gameboard and decks ------------------------------------------------------

gameboard <- data.frame(
  space = 1:40,
  title = c(
    "Go", "Mediterranean Avenue", "Community Chest", "Baltic Avenue",
    "Income Tax", "Reading Railroad", "Oriental Avenue", "Chance",
    "Vermont Avenue", "Connecticut Avenue", "Jail", "St. Charles Place",
    "Electric Company", "States Avenue", "Virginia Avenue",
    "Pennsylvania Railroad", "St. James Place", "Community Chest",
    "Tennessee Avenue", "New York Avenue", "Free Parking",
    "Kentucky Avenue", "Chance", "Indiana Avenue", "Illinois Avenue",
    "B & O Railroad", "Atlantic Avenue", "Ventnor Avenue", "Water Works",
    "Marvin Gardens", "Go to jail", "Pacific Avenue",
    "North Carolina Avenue", "Community Chest", "Pennsylvania Avenue",
    "Short Line Railroad", "Chance", "Park Place", "Luxury Tax",
    "Boardwalk"), stringsAsFactors = FALSE)

chancedeck <- data.frame(
  index = 1:15,
  card = c(
    "Advance to Go", "Advance to Illinois Ave.",
    "Advance to St. Charles Place", "Advance token to nearest Utility",
    "Advance token to the nearest Railroad",
    "Take a ride on the Reading Railroad",
    "Take a walk on the Boardwalk", "Go to Jail", "Go Back 3 Spaces",
    "Bank pays you dividend of $50", "Get out of Jail Free",
    "Make general repairs on all your property", "Pay poor tax of $15",
    "You have been elected Chairman of the Board",
    "Your building loan matures"), stringsAsFactors = FALSE)

communitydeck <- data.frame(
  index = 1:16,
  card = c(
    "Advance to Go", "Go to Jail",
    "Bank error in your favor. Collect $200", "Doctor's fees Pay $50",
    "From sale of stock you get $45", "Get Out of Jail Free",
    "Grand Opera Night Opening", "Xmas Fund matures", "Income tax refund",
    "Life insurance matures. Collect $100", "Pay hospital fees of $100",
    "Pay school tax of $150", "Receive for services $25",
    "You are assessed for street repairs",
    "You have won second prize in a beauty contest",
    "You inherit $100"), stringsAsFactors = FALSE)

# RandomDice class ---------------------------------------------------------

RandomDice <- R6Class(
  classname = "RandomDice",
```

```
  public = list(
    verbose = NA,
    initialize = function(verbose = FALSE){
      stopifnot(is.logical(verbose))
      self$verbose = verbose
    },
    roll = function() {
      outcome <- sample(1:6, size = 2, replace = TRUE)
      if(self$verbose) {
        cat("Dice Rolled:", outcome[1], collapse = ", ", outcome[2], "\n")
      }
      outcome
    }
  )
)

# Preset dice ------------------------------------------------------------

PresetDice <- R6Class(
  classname = "PresetDice",
  public = list(
    verbose = NA,
    preset_rolls = double(0),
    position = 1,
    initialize = function(rolls, verbose = FALSE){
      stopifnot(is.logical(verbose))
      stopifnot(is.numeric(rolls))
      self$preset_rolls = rolls
      self$verbose = verbose
    },
    roll = function(){
      if(self$position > length(self$preset_rolls)){
        stop("You have run out of predetermined dice outcomes.")
      }
      outcome <- c(self$preset_rolls[self$position],
                   self$preset_rolls[self$position + 1])
      self$position <- self$position + 2
      if(self$verbose){
        cat("Dice Rolled:", outcome[1], collapse = ",", outcome[2], "\n")
      }
      outcome
    }
  )
)

# Chance and Community Decks ---------------------------------------------

CardDeck <- R6Class(
  classname = "CardDeck",
  public = list(
    verbose = NA,
    deck_order = double(0),
    deck = data.frame(),
```

```
    position = 1,
    initialize = function(deck, verbose = FALSE){
      stopifnot(is.data.frame(deck),
                is.numeric(deck[[1]]),
                is.character(deck[[2]]))
      self$deck_order <- sample(length(deck[[1]]))
      self$verbose <- verbose
      self$deck <- deck
    },
    draw = function(){
      if(self$position > length(self$deck_order)){
        # if we run out of cards, shuffle deck
        # reset the position to 1
        if(self$verbose){
          cat("Shuffling deck.\n")
        }
        self$deck_order <- sample(length(self$deck[[1]]))
        self$position <- 1
      }
      outcome <- c(self$deck_order[self$position])
      self$position <- self$position + 1
      if(self$verbose){
        cat("Card:", self$deck[outcome, 2], "\n")
      }
      outcome
    }
  )
)


# R6 Class Player -------------------------------------------------------

Player <- R6Class(
  classname = "Player",
  public = list(
    pos = 1, # starting position
    jail_times = 0, # count the number of jail times
    double_times = 0, # count the number of getting double rolls
    verbose = TRUE,
    initialize = function(pos, jail_times, double_times, verbose = FALSE) {
      stopifnot(is.numeric(pos))
      stopifnot(is.numeric(jail_times))
      stopifnot(is.numeric(double_times))
      stopifnot(is.logical(verbose))
      self$pos = pos
      self$jail_times = jail_times
      self$double_times = double_times
      self$verbose = verbose
    },
    # takes the player's movement
    move_fwd = function(n) {
      if(self$verbose) {
      self$pos <- self$pos + n
```

```
      if(self$pos > 40) {
        self$pos <- self$pos - 40
      }
      # tells where the player moved to and by how many
      if(self$verbose) {
        cat("Player Moved This Many:", n, "\n")
        self$pos <- self$pos + n
        cat("Player is Now at:", self$pos, "\n")
      }
      # Space 31 is where Jail space is located
      if(self$pos == 31) {
        cat("You Will Go to Jail, Do Not Pass Go", "\n")
        self$pos <- 11 # Space of jail located
        self$jail_times <- self$jail_times + 1
      }
      }
    },
# Interesting note is that you cannot use "<-" within R6 but you have to use "=" for functions
    # If spaces make you jump a space
    Jump_to_Space = function(n) {
      if(self$verbose) {
        cat("Player at: ", self$pos, "-->", gameboard$title[self$pos], "\n")
        self$pos <- n
        }
      if(self$verbose) {
        cat("Player Now at: ", self$pos, "\n")
        }
    },
    # Tallying doubles to send you to jail if done 3 times in a row or straight jail
    Counts_of_Doubles_Rolled = function(n) {
      if(self$verbose) {
        self$double_times <- self$double_times + 1
        cat("Double count is now: ", self$double_times, "\n")
      }
      if(self$double_times == 3){
        cat("You Will Go to Jail, Do Not Pass Go", "\n")
        self$pos <- 11 # Space of jail located
        self$jail_times <- self$jail_times + 1
      }
    },
    # What happens if you land on the Chance deck card card (will take a while to code)
    Chancedeck_Draw = function(x) {
      if(self$verbose) {
        cat("Draw a Card from the Community Chest", x, "\n")
        x <- sample(chancedeck$index, 1)
      }
      if(self$verbose) {
        cat("Chance Card Picked: ", self$pos, chancedeck$card[x], "\n")
    # I've tried running position to position and was too hard for me. So, I found out
        # the easiest method is to add or subtract the position (+/-)
      }
      # "There are nine cards in the Chance deck that move the player's token"
      if(x == 1) {
```

```
        self$pos <- self$pos + 1
    }
      else
    if(x == 2) {
    if(self$pos == 8) {
        self$pos <- self$pos + 17
    }
      else
    if(self$pos == 23) {
        self$pos <- self$pos + 2
    }
      else
    if(self$pos == 37) {
        self$pos <- self$pos - 12
    }
    }
      else
    if(x == 3) {
    if(self$pos == 8) {
        self$pos <- self$pos + 4
    }
      else
    if(self$pos == 23) {
        self$pos <- self$pos - 11
    }
      else
    if(self$pos == 37) {
        self$pos <- self$pos - 25
    }
    }
      else
    if(x == 4) {
    if(self$pos == 8) {
        self$pos <- self$pos + 5
    }
      else
    if(pos == 23) {
        self$pos <- self$pos + 6
    }
      else
    if(self$pos == 37) {
        self$pos <- self$pos + 2
    }
    }
      else
    if(x == 5) {
    if(self$pos == 8) {
        self$pos <- self$pos + 8
    }
      else
    if(self$pos == 23) {
        self$pos <- self$pos + 3
    }
```

```
    else
  if(self$pos == 37) {
      self$pos <- self$pos -31
  }
  }
    else
  if(x == 6) {
  if(self$pos == 8) {
      self$pos <- self$pos - 2
  }
    else
  if(self$pos == 23) {
      self$pos <- self$pos - 17
  }
    else
  if(self$pos == 37) {
      self$pos <- self$pos - 31
  }
  }
    else
  if(x == 7) {
  if(self$pos == 8) {
      self$pos <- self$pos +32
  }
    else
  if(self$pos == 23) {
      self$pos <- self$pos + 17
  }
    else
  if(self$pos == 37) {
      self$pos <- self$pos + 3
  }
  }
    else
  if(x == 8) {
    self$pos <- 11
  }
    else
  if(x == 9) {
    self$pos <- self$pos - 3
  }
    else {
    self$pos <- self$pos
  }
},
# What happens if you draw a Community Chest deck card
Communitydeck_Draw = function(x) {
  if(self$verbose) {
    cat("Drew community card")
    x <- sample(communitydeck$index,1)
  }
  if(self$verbose) {
    cat("Community Card Draw ", self$pos, communitydeck$card[x], "\n")
```

```
      }
      if(x == 1) {
        self$pos <- 1
      }
      if(x == 2) {
        self$pos <- 11
        self$jail_times <- self$jail_times + 1
      }
      else {
        self$pos <- self$pos
    }
    }
  )
)

# R6 Class SpaceTracker ------------------------------------------------

SpaceTracker <- R6Class(
  classname = "SpaceTracker",
  public = list(
    counts = rep(0, 40),
    verbose = TRUE,
    initialize = function(counts, verbose = FALSE) {
      stopifnot(is.numeric(counts))
      stopifnot(is.logical(verbose))
      self$counts = counts
      self$verbose = verbose
    },
    tally = function(x, verbose) {
      self$counts[x] <- self$counts[x] + 1
      if(self$verbose) {
        cat("Added tally to:", x, ": ", gameboard$title[x], "\n")
      }
    }
  )
)

# VERY BASIC turn taking example ---------------------------------------

take_turn <- function(player, spacetracker){
  dice_rolls <- dice$roll()
  player$move_fwd(sum(dice_rolls))
  spacetracker$tally(player$pos)
}

##############################################################################################
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
)
```

```r
player1 <- Player$new(pos = 1, jail_times = 0, double_times = 0, verbose = TRUE)
monopoly <- SpaceTracker$new(counts = rep(0,40), verbose = TRUE)

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

```
## --- Turn 1 ---
## Dice Rolled: 6 , 4
## Player Moved This Many: 10
## Player is Now at: 21
## Added tally to: 21 :  Free Parking
##
## --- Turn 2 ---
## Dice Rolled: 5 , 3
## Player Moved This Many: 8
## Player is Now at: 37
## Added tally to: 37 :  Chance
##
## --- Turn 3 ---
## Dice Rolled: 3 , 5
## Player Moved This Many: 8
## Player is Now at: 13
## Added tally to: 13 :  Electric Company
##
## --- Turn 4 ---
## Dice Rolled: 4 , 4
## Player Moved This Many: 8
## Player is Now at: 29
## Added tally to: 29 :  Water Works
##
## --- Turn 5 ---
## Dice Rolled: 4 , 4
## Player Moved This Many: 8
## Player is Now at: 45
## Added tally to: 45 :  NA
##
## --- Turn 6 ---
## Dice Rolled: 2 , 2
## Player Moved This Many: 4
## Player is Now at: 13
## Added tally to: 13 :  Electric Company
##
## --- Turn 7 ---
## Dice Rolled: 4 , 3
## Player Moved This Many: 7
## Player is Now at: 27
## Added tally to: 27 :  Atlantic Avenue
##
## --- Turn 8 ---
## Dice Rolled: 4 , 4
## Player Moved This Many: 8
```

```
## Player is Now at: 43
## Added tally to: 43 :  NA
##
## --- Turn 9 ---
## Dice Rolled: 1 , 4
## Player Moved This Many: 5
## Player is Now at: 13
## Added tally to: 13 :  Electric Company
##
## --- Turn 10 ---
## Dice Rolled: 3 , 4
## Player Moved This Many: 7
## Player is Now at: 27
## Added tally to: 27 :  Atlantic Avenue
##
## --- Turn 11 ---
## Dice Rolled: 1 , 2
## Player Moved This Many: 3
## Player is Now at: 33
## Added tally to: 33 :  North Carolina Avenue
##
## --- Turn 12 ---
## Dice Rolled: 3 , 6
## Player Moved This Many: 9
## Player is Now at: 11
## Added tally to: 11 :  Jail
##
## --- Turn 13 ---
## Dice Rolled: 5 , 4
## Player Moved This Many: 9
## Player is Now at: 29
## Added tally to: 29 :  Water Works
##
## --- Turn 14 ---
## Dice Rolled: 5 , 5
## Player Moved This Many: 10
## Player is Now at: 49
## Added tally to: 49 :  NA
##
## --- Turn 15 ---
## Dice Rolled: 1 , 2
## Player Moved This Many: 3
## Player is Now at: 15
## Added tally to: 15 :  Virginia Avenue
##
## --- Turn 16 ---
## Dice Rolled: 5 , 4
## Player Moved This Many: 9
## Player is Now at: 33
## Added tally to: 33 :  North Carolina Avenue
##
## --- Turn 17 ---
## Dice Rolled: 3 , 3
## Player Moved This Many: 6
```

```
## Player is Now at: 45
## Added tally to: 45 :  NA
##
## --- Turn 18 ---
## Dice Rolled: 6 , 1
## Player Moved This Many: 7
## Player is Now at: 19
## Added tally to: 19 :  Tennessee Avenue
##
## --- Turn 19 ---
## Dice Rolled: 1 , 1
## Player Moved This Many: 2
## Player is Now at: 23
## Added tally to: 23 :  Chance
##
## --- Turn 20 ---
## Dice Rolled: 2 , 3
## Player Moved This Many: 5
## Player is Now at: 33
## Added tally to: 33 :  North Carolina Avenue
```

```
a <- na.omit(monopoly$counts)
a
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0 1 0 3 0 1 0 0 0 1 0 1 0 1 0 0 0 2 0 2 0 0 0 3 0 0 0 1 0
## [39] 0 0
## attr(,"na.action")
## [1] 41 42 43 44 45 46 47 48 49
## attr(,"class")
## [1] "omit"
```

```
cbind(gameboard, counts = a)
```

```
##    space                 title counts
## 1      1                    Go      0
## 2      2  Mediterranean Avenue      0
## 3      3       Community Chest      0
## 4      4         Baltic Avenue      0
## 5      5            Income Tax      0
## 6      6      Reading Railroad      0
## 7      7       Oriental Avenue      0
## 8      8               Chance      0
## 9      9        Vermont Avenue      0
## 10    10    Connecticut Avenue      0
## 11    11                  Jail      1
## 12    12     St. Charles Place      0
## 13    13      Electric Company      3
## 14    14         States Avenue      0
## 15    15       Virginia Avenue      1
## 16    16 Pennsylvania Railroad      0
## 17    17       St. James Place      0
## 18    18       Community Chest      0
## 19    19      Tennessee Avenue      1
## 20    20       New York Avenue      0
## 21    21          Free Parking      1
## 22    22       Kentucky Avenue      0
```

```
## 23    23               Chance      1
## 24    24       Indiana Avenue      0
## 25    25      Illinois Avenue      0
## 26    26       B & O Railroad      0
## 27    27      Atlantic Avenue      2
## 28    28       Ventnor Avenue      0
## 29    29          Water Works      2
## 30    30       Marvin Gardens      0
## 31    31           Go to jail      0
## 32    32       Pacific Avenue      0
## 33    33 North Carolina Avenue      3
## 34    34      Community Chest      0
## 35    35  Pennsylvania Avenue      0
## 36    36  Short Line Railroad      0
## 37    37               Chance      1
## 38    38           Park Place      0
## 39    39           Luxury Tax      0
## 40    40            Boardwalk      0
```

```
## You must use these dice for Part 1
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)
sample(15)
```

```
##  [1]  5  6 14  8  1 11  9  2  3 10  7 12  4 13 15
```

```
chance <- CardDeck$new(chancedeck, TRUE)
community <- CardDeck$new(communitydeck, TRUE)
# if your chance cards different from mine,
# check to make sure sample(15) returns the following
# > set.seed(2)
# > sample(15)
#  [1]  5  6 14  8  1 11  9  2  3 10  7 12  4 13 15
```

Part 2: 1000 simulated games

```
## Use silent 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$new(pos = 1, jail_times = 0, double_times = 0, verbose = FALSE)
player2 <- Player$new(pos = 1, jail_times = 0, double_times = 0, verbose = FALSE)
space_tracking <- SpaceTracker$new(counts = rep(0, 40), verbose = FALSE)

for(g in 1:1000) {
  if(g %% 100 == 0) {
    cat("#### SIMULATING GAME", g, "##### \n")
```

```
  }
  for(i in 1:150){
    take_turn(player1, space_tracking)
    take_turn(player2, space_tracking)
  }
}
```

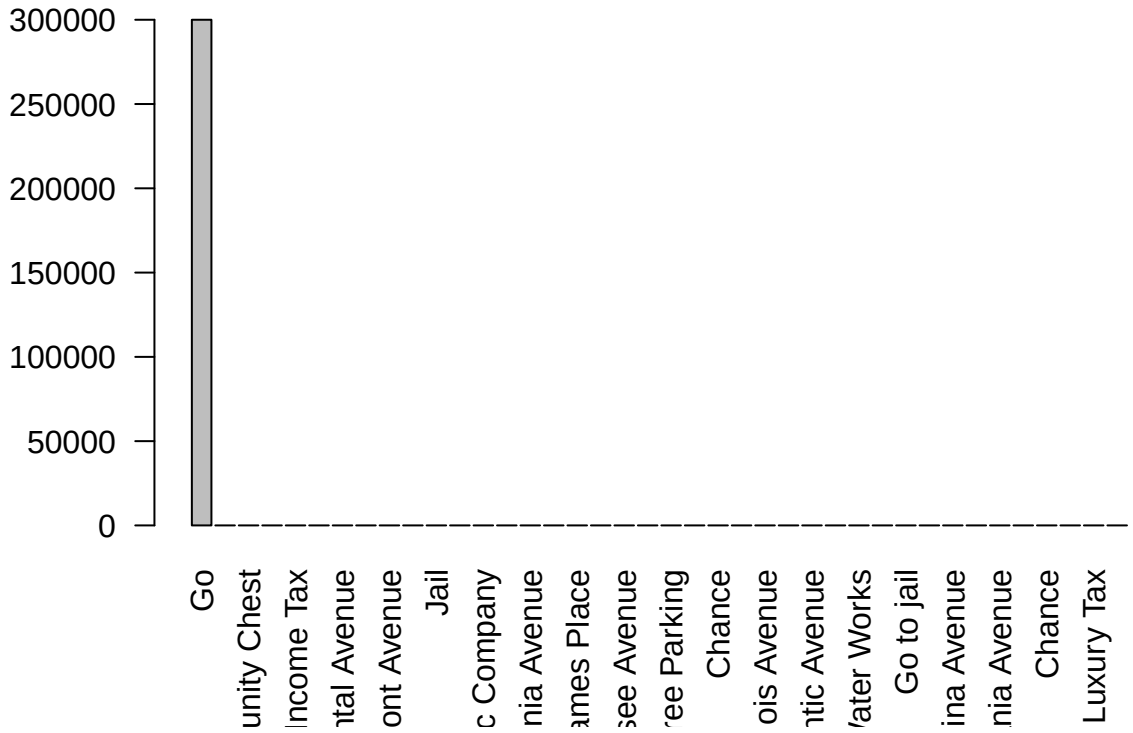
```
## #### SIMULATING GAME 100 #####
## #### SIMULATING GAME 200 #####
## #### SIMULATING GAME 300 #####
## #### SIMULATING GAME 400 #####
## #### SIMULATING GAME 500 #####
## #### SIMULATING GAME 600 #####
## #### SIMULATING GAME 700 #####
## #### SIMULATING GAME 800 #####
## #### SIMULATING GAME 900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

```
results <- cbind(gameboard, tally = space_tracking$counts)
results <- cbind(results, rel = results$tally/sum(results$tally))
results_arranged <- results[order(results$tally, decreasing = TRUE), ]
print(results_arranged)
```

```
##    space                 title tally rel
## 1      1                    Go 3e+05   1
## 2      2  Mediterranean Avenue 0e+00   0
## 3      3       Community Chest 0e+00   0
## 4      4         Baltic Avenue 0e+00   0
## 5      5            Income Tax 0e+00   0
## 6      6      Reading Railroad 0e+00   0
## 7      7       Oriental Avenue 0e+00   0
## 8      8               Chance 0e+00   0
## 9      9        Vermont Avenue 0e+00   0
## 10    10    Connecticut Avenue 0e+00   0
## 11    11                  Jail 0e+00   0
## 12    12     St. Charles Place 0e+00   0
## 13    13      Electric Company 0e+00   0
## 14    14         States Avenue 0e+00   0
## 15    15       Virginia Avenue 0e+00   0
## 16    16 Pennsylvania Railroad 0e+00   0
## 17    17       St. James Place 0e+00   0
## 18    18       Community Chest 0e+00   0
## 19    19      Tennessee Avenue 0e+00   0
## 20    20       New York Avenue 0e+00   0
## 21    21          Free Parking 0e+00   0
## 22    22       Kentucky Avenue 0e+00   0
## 23    23               Chance 0e+00   0
## 24    24        Indiana Avenue 0e+00   0
## 25    25       Illinois Avenue 0e+00   0
## 26    26        B & O Railroad 0e+00   0
## 27    27       Atlantic Avenue 0e+00   0
## 28    28        Ventnor Avenue 0e+00   0
## 29    29           Water Works 0e+00   0
## 30    30        Marvin Gardens 0e+00   0
```

```
## 31    31            Go to jail 0e+00   0
## 32    32        Pacific Avenue 0e+00   0
## 33    33 North Carolina Avenue 0e+00   0
## 34    34       Community Chest 0e+00   0
## 35    35   Pennsylvania Avenue 0e+00   0
## 36    36   Short Line Railroad 0e+00   0
## 37    37               Chance 0e+00   0
## 38    38           Park Place 0e+00   0
## 39    39           Luxury Tax 0e+00   0
## 40    40            Boardwalk 0e+00   0
```

```
barplot(results$tally, names.arg = results$title, horiz = FALSE, las = 2)
```



I was unable to fully simulate 1000 games with 150 turns. My simulation with 2 players would not work for some reason. I was able to complete one game with 20 turns and verbose = TURE.