Stats 102A - Homework 2 - Output File

Yuetong Li

Homework questions and prompts copyright Miles Chen, Do not post, share, or distribute without permission.

To receive full credit the functions you write must pass all tests. We may conduct further tests that are not included on this page as well.

Academic Integrity Statement

By including this statement, I, Yuetong Li, 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.

```r
source("102a_hw_02_script_Yuetong_Li.R")  # edit with your file name
```

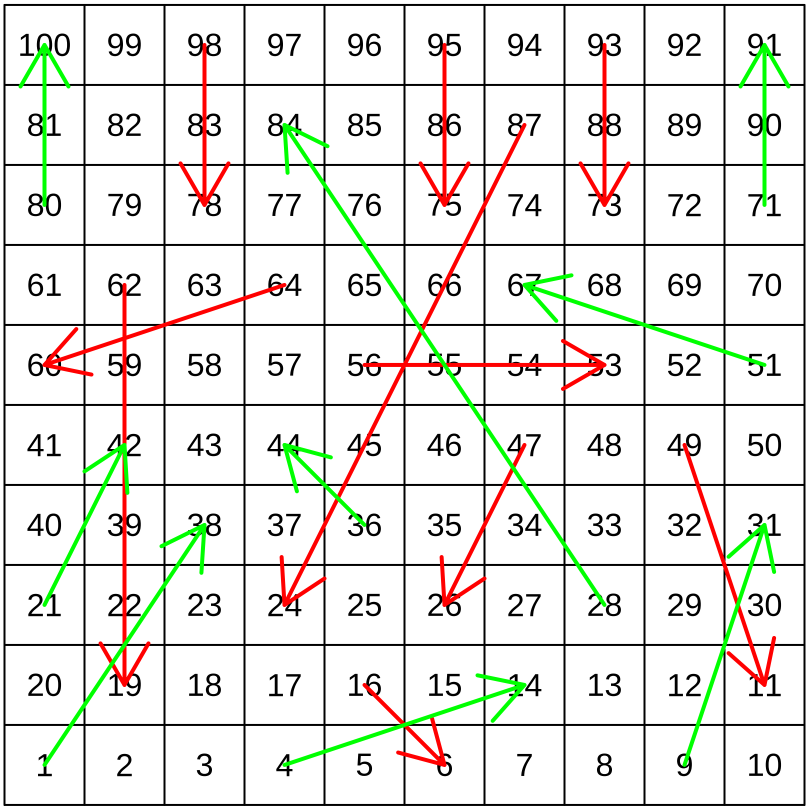
Part 1: Board representation

Create a single list object called `board` where you store the features of the game board in R.

```r
board <- list(dim = c(10, 10),
              chutes = rbind(c(16,6), c(47, 26), c(49, 11), c(56, 53), c(62, 19),
                             c(64, 60), c(87, 24), c(93, 73), c(95, 75), c(98, 78)),
              ladders = rbind(c(1, 38), c(4, 14), c(9, 31), c(21, 42), c(28, 84),
c(36, 44), c(51, 67), c(71, 91), c(80, 100)))
```

Part 2: Plot of Game board

```
# par() should help the plot be more visible. you can adjust this as necessary
par(mar = c(0, 0, 0, 0))
show_board(board)
```

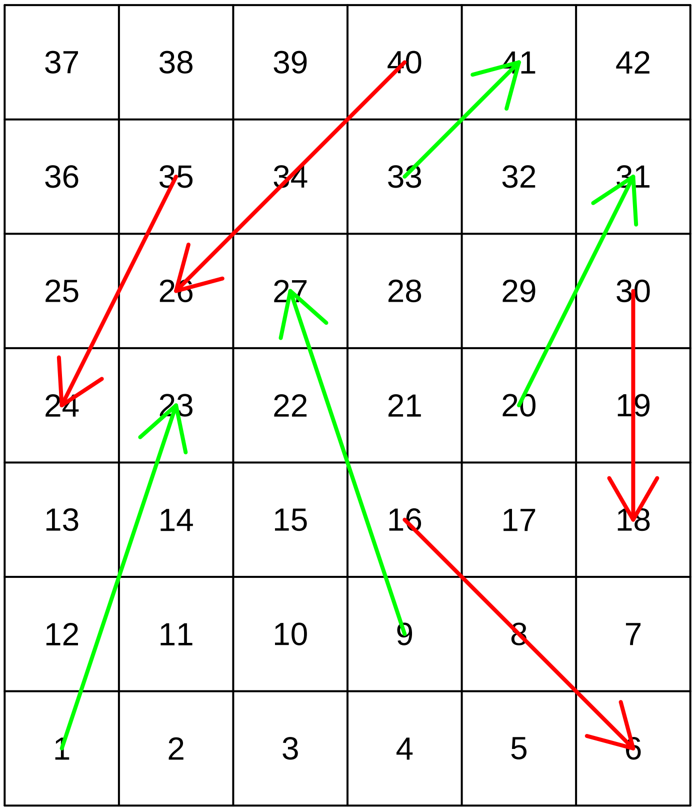


100	99	98	97	96	95	94	93	92	91
81	82	83	84	85	86	87	88	89	90
80	79	78	77	76	75	74	73	72	71
61	62	63	64	65	66	67	68	69	70
60	59	58	57	56	55	54	53	52	51
41	42	43	44	45	46	47	48	49	50
40	39	38	37	36	35	34	33	32	31
21	22	23	24	25	26	27	28	29	30
20	19	18	17	16	15	14	13	12	11
1	2	3	4	5	6	7	8	9	10

Part 3: Miniboards

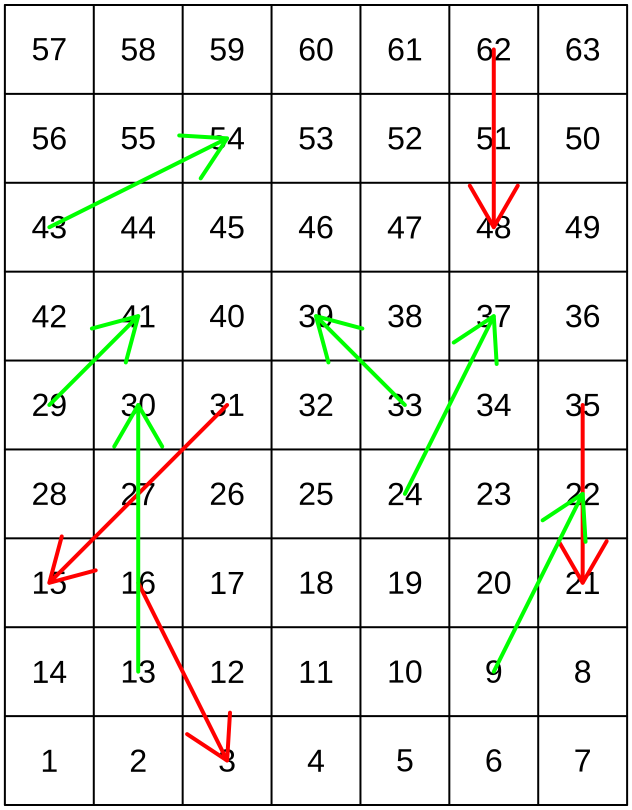
Create the `miniboard` objects and plots.

```
miniboard1 <- list(dim = c(7, 6),
                   chutes = rbind(c(16, 6), c(30, 18), c(35, 24), c(40, 26)),
                   ladders = rbind(c(1, 23), c(9, 27), c(20, 31), c(33, 41)))
par(mar = c(0, 0, 0, 0))
show_board(miniboard1)
```



37	38	39	40	41	42
36	35	34	33	32	31
25	26	27	28	29	30
24	23	22	21	20	19
13	14	15	16	17	18
12	11	10	9	8	7
1	2	3	4	5	6

```
miniboard2 <- list(dim = c(9, 7),
                   chutes = rbind(c(16, 3), c(31, 15), c(35, 21), c(62, 48)),
                   ladders = rbind(c(9, 22), c(13, 30), c(24, 37), c(29, 41), c(33, 39), c(43, 54)))
par(mar = c(0, 0, 0, 0))
show_board(miniboard2)
```



```
miniboard3 <- list(dim = c(9, 8),
                   chutes = NULL,
                   ladders = NULL)
par(mar = c(0, 0, 0, 0))
show_board(miniboard3)
```

65	66	67	68	69	70	71	72
64	63	62	61	60	59	58	57
49	50	51	52	53	54	55	56
48	47	46	45	44	43	42	41
33	34	35	36	37	38	39	40
32	31	30	29	28	27	26	25
17	18	19	20	21	22	23	24
16	15	14	13	12	11	10	9
1	2	3	4	5	6	7	8

Part 4: Verbose output of one single player game

```r
set.seed(5)
play_solo(board, verbose = TRUE)
```

```
## Turn 1
## Start at 0
## Spinner: 2
## Turn ends at 2
##
## Turn 2
## Start at 2
## Spinner: 3
## Turn ends at 5
##
## Turn 3
## Start at 5
## Spinner: 1
## Turn ends at 6
##
## Turn 4
## Start at 6
## Spinner: 3
## Landed on: 9
## Ladder!
## Turn ends at 31
##
## Turn 5
## Start at 31
## Spinner: 1
## Turn ends at 32
##
## Turn 6
## Start at 32
## Spinner: 1
## Turn ends at 33
##
## Turn 7
## Start at 33
## Spinner: 5
## Turn ends at 38
##
## Turn 8
## Start at 38
## Spinner: 6
## Turn ends at 44
##
## Turn 9
## Start at 44
## Spinner: 3
## Landed on: 47
## Chute!
## Turn ends at 26
##
## Turn 10
## Start at 26
## Spinner: 3
## Turn ends at 29
##
## Turn 11
## Start at 29
## Spinner: 6
## Turn ends at 35
##
## Turn 12
## Start at 35
## Spinner: 2
## Turn ends at 37
##
## Turn 13
## Start at 37
## Spinner: 5
## Turn ends at 42
##
## Turn 14
## Start at 42
## Spinner: 4
## Turn ends at 46
##
## Turn 15
## Start at 46
## Spinner: 2
## Turn ends at 48
##
## Turn 16
## Start at 48
## Spinner: 5
## Turn ends at 53
##
## Turn 17
## Start at 53
## Spinner: 3
## Landed on: 56
## Chute!
## Turn ends at 53
##
## Turn 18
## Start at 53
## Spinner: 1
## Turn ends at 54
##
## Turn 19
## Start at 54
## Spinner: 6
## Turn ends at 60
##
```

```
## Turn 20
## Start at 60
## Spinner: 4
## Landed on: 64
## Chute!
## Turn ends at 60
##
## Turn 21
## Start at 60
## Spinner: 3
## Turn ends at 63
##
## Turn 22
## Start at 63
## Spinner: 2
## Turn ends at 65
##
## Turn 23
## Start at 65
## Spinner: 5
## Turn ends at 70
##
## Turn 24
## Start at 70
## Spinner: 2
## Turn ends at 72
##
## Turn 25
## Start at 72
## Spinner: 2
## Turn ends at 74
##
## Turn 26
## Start at 74
## Spinner: 3
## Turn ends at 77
##
## Turn 27
## Start at 77
## Spinner: 1
## Turn ends at 78
##
## Turn 28
## Start at 78
## Spinner: 2
## Landed on: 80
## Ladder!
## Turn ends at 100
```

```
## $turns
## [1] 28
##
## $chute_tally
##  [1] 0 1 0 1 0 1 0 0 0 0
```

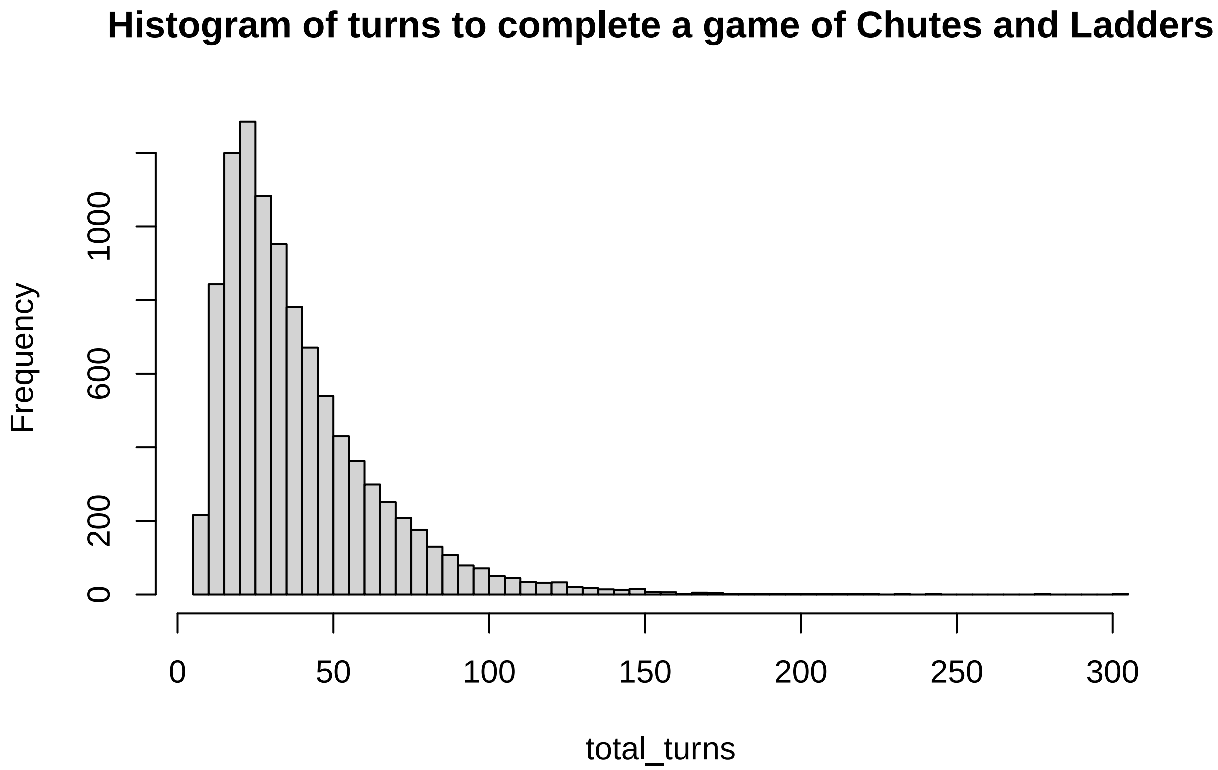
```
##
## $ladder_tally
## [1] 0 0 1 0 0 0 0 0 1
##
## $move_log
##  [1]   2   5   6  31  32  33  38  44  26  29  35  37  42  46  48  53  53  54  60
## [20]  60  63  65  70  72  74  77  78 100
```

Part 5: Monte Carlo Simulation Study

```
# run 10,000 games
total_turns <- replicate(n=10000, play_solo(board, verbose = FALSE)$turns)
```

- Create a histogram (breaks = 50) of the turns.

```
hist(total_turns, breaks = 50, main = "Histogram of turns to complete a game of Chutes and Ladders")
```



- Find the minimum number of turns. How many times out of 10,000 did a game finish with the minimum number of turns?

```
min(total_turns)
```

```
## [1] 7
```

```
sum(total_turns == min(total_turns))
```

```
## [1] 10
```

- Find the maximum number of turns.

```
max(total_turns)
```

```
## [1] 301
```

- What is the median number of turns?

```
median(total_turns)
```

```
## [1] 32
```

- What is the mean number of turns?

```
mean(total_turns)
```

```
## [1] 39.3484
```

- What proportion of games take 100 or more turns to complete?

```
sum(total_turns >= 100)/10000
```

```
## [1] 0.0329
```

- What proportion of games take 10 or fewer turns to complete?

```
sum(total_turns <= 10)/10000
```

```
## [1] 0.0216
```

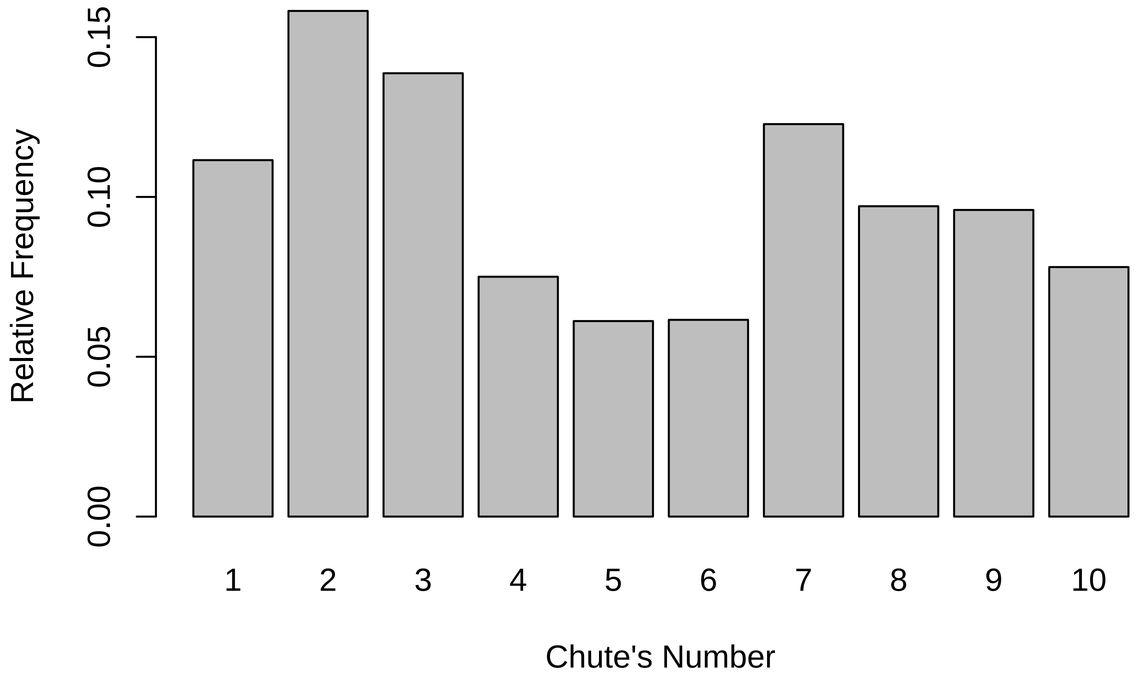
- What proportion of games utilize ladder 9 (the shortcut to win on space 80)?

```
sum(replicate(n=10000, play_solo(board, verbose = FALSE)$ladder_tally[9]) == 1)/10000
```

```
## [1] 0.484
```

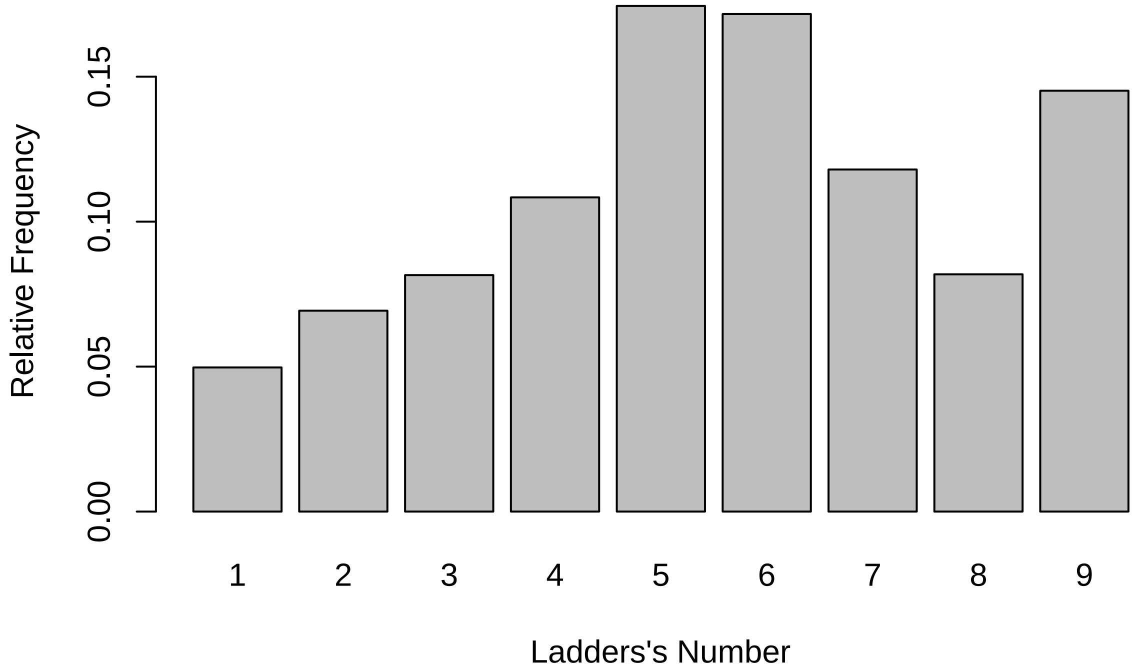
- Create a barplot of the relative frequency of how often each chute is utilized.

```
chute_data <- replicate(n=10000, play_solo(board, verbose = FALSE)$chute_tally)
num1 <- integer(length(chute_data[,1]))
for (i in seq_along(chute_data[,1])) {
  num1[i] <- sum(chute_data[i,])
}
num1 <- setNames(num1, 1:length(chute_data[,1]))
barplot(num1 / sum(num1), xlab = "Chute's Number", ylab = "Relative Frequency")
```



-Create a barplot of the relative frequency of how often each ladder is utilized.

```
ladder_data <- replicate(n=10000, play_solo(board, verbose = FALSE)$ladder_tally)
num2 <- integer(length(ladder_data[,1]))
for (i in seq_along(ladder_data[,1])) {
  num2[i] <- sum(ladder_data[i,])
}
num2 <- setNames(num2, 1:length(ladder_data[,1]))
barplot(num2 / sum(num2), xlab = "Ladders's Number", ylab = "Relative Frequency")
```


Relative Frequency
0.00
0.05
0.10
0.15
1
2
3
4
5
6
7
8
9
Ladders's Number