Assignment 1

Natali Tckvitishvili

2024-05-28

```
library(tidyr)
library(dplyr)
library(ggplot2)
library(stringi)
```

Task Set 1

For tasks 1.1-1.3, assume you throw 3 dice – normal dice with 6 sides each ;)

Task 1.1

Create a data frame with all possible combinations (outcomes) that can result from throwing all the dice. (Each row should represent one possible outcome.) Print the first and last 10 rows of the data frame and state how many possible outcomes there are.

Data frame with all possible combinations is created using expand.grid function which creates all the possible combinations of the outcomes.

```
dice_seq <- seq(1:6)
dice_throws <- data.frame(
  expand.grid(dice1 = dice_seq, dice2 = dice_seq, dice3 = dice_seq)
)
```

- top 10 rows

```
head(dice_throws, 10)
```

```
##    dice1 dice2 dice3
## 1      1     1     1
## 2      2     1     1
## 3      3     1     1
## 4      4     1     1
## 5      5     1     1
## 6      6     1     1
## 7      1     2     1
## 8      2     2     1
## 9      3     2     1
## 10     4     2     1
```

- last 10 rows

```
tail(dice_throws, 10)
```

```
##     dice1 dice2 dice3
## 207     3     5     6
## 208     4     5     6
## 209     5     5     6
## 210     6     5     6
## 211     1     6     6
## 212     2     6     6
## 213     3     6     6
## 214     4     6     6
## 215     5     6     6
## 216     6     6     6
```

- number of possible outcomes

```
nrow(dice_throws)
```

```
## [1] 216
```

Task 1.2

Create a data frame showing all possible sums that can result from throwing the three dice along with their probabilities. Report the results in a summary table (data frame) and a plot (visual graph).

- summary table

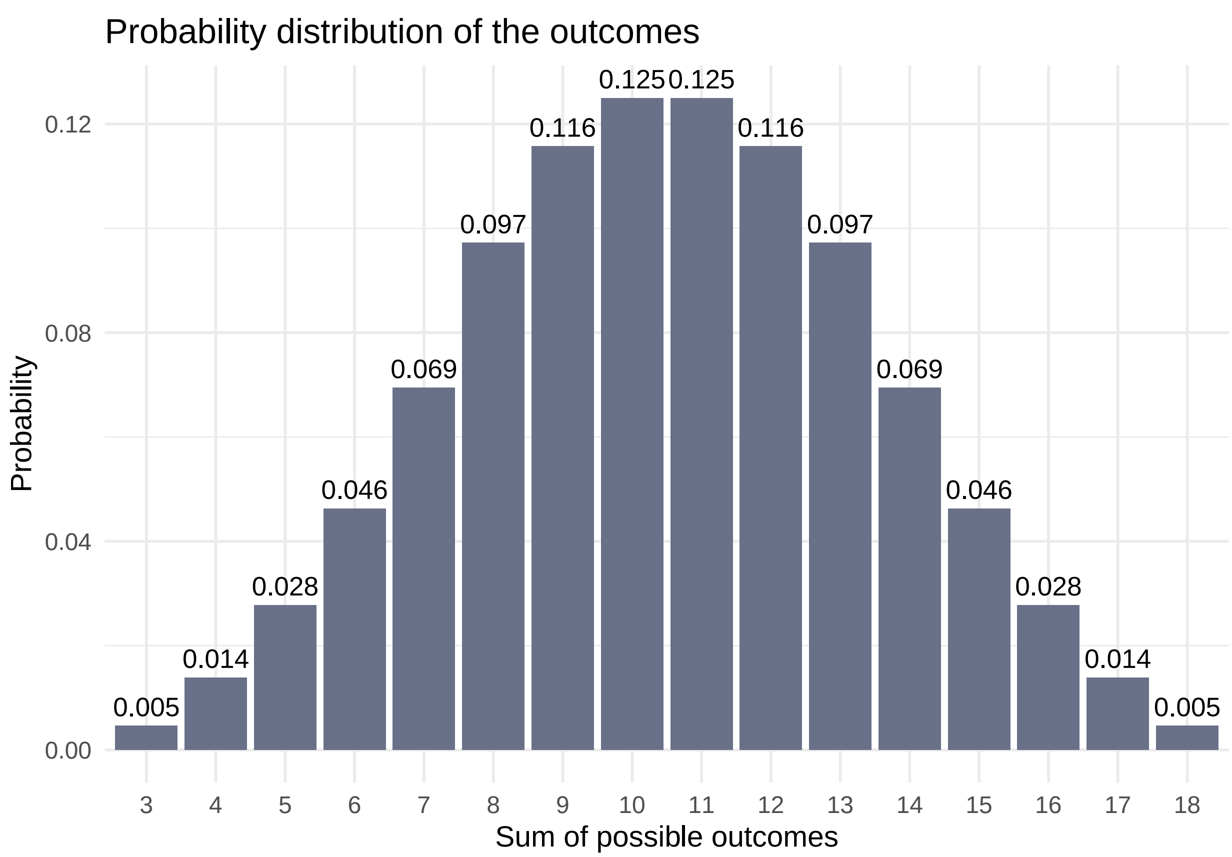
```
dice_throws <- dice_throws %>%
  mutate(possible_sum = dice1 + dice2 + dice3)

total_sum <- nrow(dice_throws) # total number of outcomes to calculate probability

dice_summary <- dice_throws %>%
  group_by(possible_sum) %>%
  summarise(frequency = n()) %>% # count of each sum
  mutate(probability = frequency / total_sum) # probability of each sum

data.frame(dice_summary)
```

```
##   possible_sum frequency probability
## 1            3         1  0.00462963
## 2            4         3  0.01388889
## 3            5         6  0.02777778
## 4            6        10  0.04629630
## 5            7        15  0.06944444
## 6            8        21  0.09722222
## 7            9        25  0.11574074
## 8           10        27  0.12500000
```

```
## 9            11        27  0.12500000
## 10           12        25  0.11574074
## 11           13        21  0.09722222
## 12           14        15  0.06944444
## 13           15        10  0.04629630
## 14           16         6  0.02777778
## 15           17         3  0.01388889
## 16           18         1  0.00462963
```

- visualization

```
ggplot(dice_summary, aes(x = as.factor(possible_sum), y = probability)) +
  geom_bar(stat = "identity", fill = "#697188") +
  geom_text(aes(label = round(probability, 3)), vjust = -0.5, size = 3.5) +
  theme_minimal() +
  labs(
    title = "Probability distribution of the outcomes",
    x = "Sum of possible outcomes",
    y = "Probability"
  )
```



Task 1.3

Compute the probability that the sum is $\geq$ 10, *given* that at least one of the dice shows a 3.

```r
# assigning variables to avoid hardcode
given_dice <- 3
condition_sum <- 10

given_data <- dice_throws %>%
  filter(dice1 == given_dice | dice2 == given_dice | dice3 == given_dice)

condition_data <- given_data %>%
  filter(possible_sum >= condition_sum)

nrow(condition_data) / nrow(given_data)
```

```
## [1] 0.5934066
```

Task Set 2

For Task 2.1-2.3, assume you toss a globe `10` times, leading to either land or water.

```r
n <- 10 # num of globe tosses
```

Task 2.1

Compute the probability of all possible numbers of occurrence of land, given the candidate proportion of `.5`. Report the results in a summary table and a plot and indicate whether the plot shows a probability distribution or a likelihood function.

```r
# create a data frame with all possible outcomes
globe <- c("L", "W")
toss_list <- replicate(n, globe, simplify = FALSE) # list of n elements with vector globe
globe_tosses <- data.frame(
  do.call(expand.grid, toss_list) # apply expand.grid to all the n elements
)

total_globe <- nrow(globe_tosses) # total number of outcomes to calculate probability

# calculate the probabilities of land occurrence
globe_tosses <- globe_tosses %>%
  mutate(count_land = rowSums(globe_tosses == "L"))

# calculate likelihood of each probability
globe_summary <- globe_tosses %>%
  group_by(count_land) %>%
  summarise(frequency = n()) %>% # count of each probability
  mutate(probability = frequency / total_globe) # likelihood of each probability

globe_summary
```

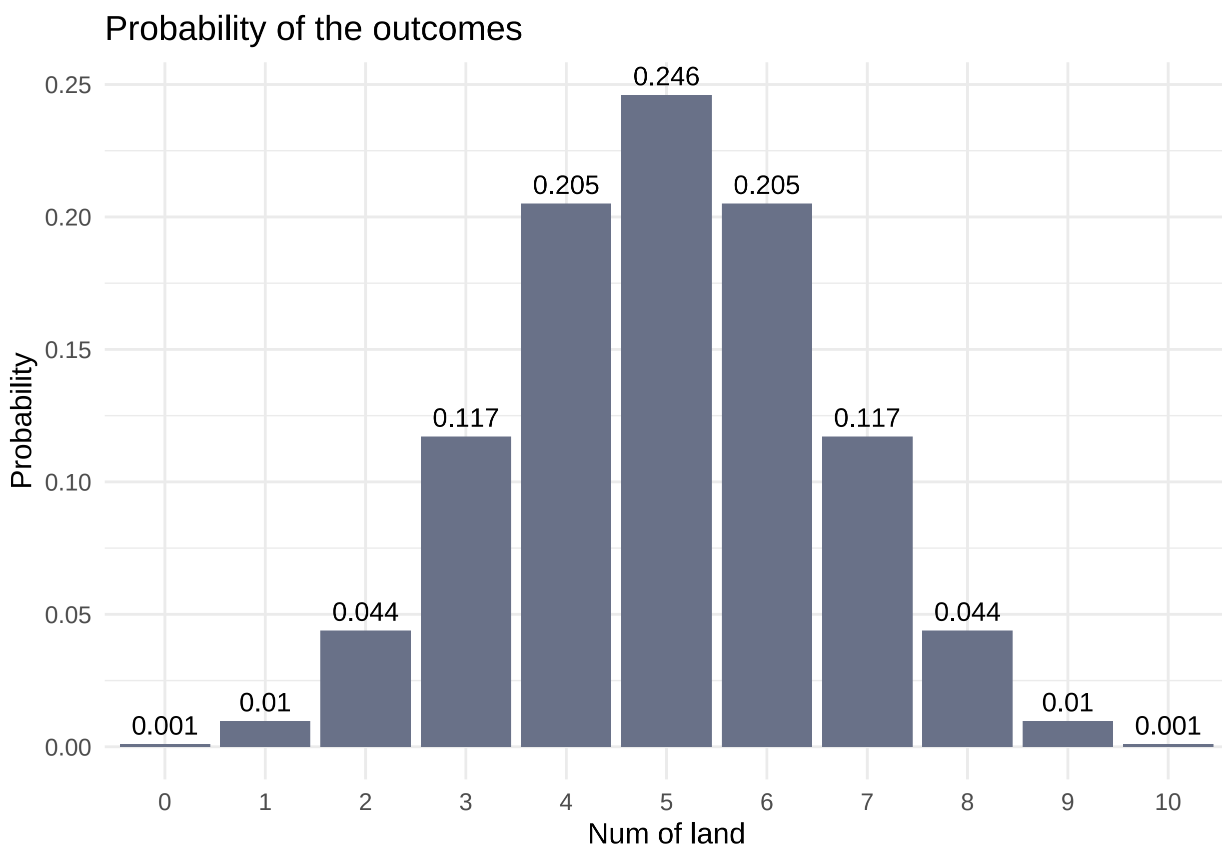
```
## # A tibble: 11 x 3
##    count_land frequency probability
##         <dbl>     <int>       <dbl>
##  1          0         1    0.000977
##  2          1        10    0.00977
##  3          2        45    0.0439
##  4          3       120    0.117
##  5          4       210    0.205
##  6          5       252    0.246
##  7          6       210    0.205
##  8          7       120    0.117
##  9          8        45    0.0439
## 10          9        10    0.00977
## 11         10         1    0.000977
```

Other way to do that is to use dbinom, but the solution above helped me to understand the underlying logic

```r
cp <- 0.5
probabilities <- dbinom(0:n, size = n, prob = cp)
globe_summary <- data.frame(count_land = 0:n, probability = probabilities)
data.frame(globe_summary)
```

```
##    count_land  probability
## 1           0 0.0009765625
## 2           1 0.0097656250
## 3           2 0.0439453125
## 4           3 0.1171875000
## 5           4 0.2050781250
## 6           5 0.2460937500
## 7           6 0.2050781250
## 8           7 0.1171875000
## 9           8 0.0439453125
## 10          9 0.0097656250
## 11         10 0.0009765625
```

```r
ggplot(globe_summary, aes(x = as.factor(count_land), y = probability)) +
  geom_bar(stat = "identity", fill = "#697188") +
  geom_text(aes(label = round(probability, 3)), vjust = -0.5, size = 3.5) +
  theme_minimal() +
  labs(
    title = "Probability of the outcomes",
    x = "Num of land",
    y = "Probability"
  )
```



The plot shows a probability distributions of observing each count of land.

Task 2.2

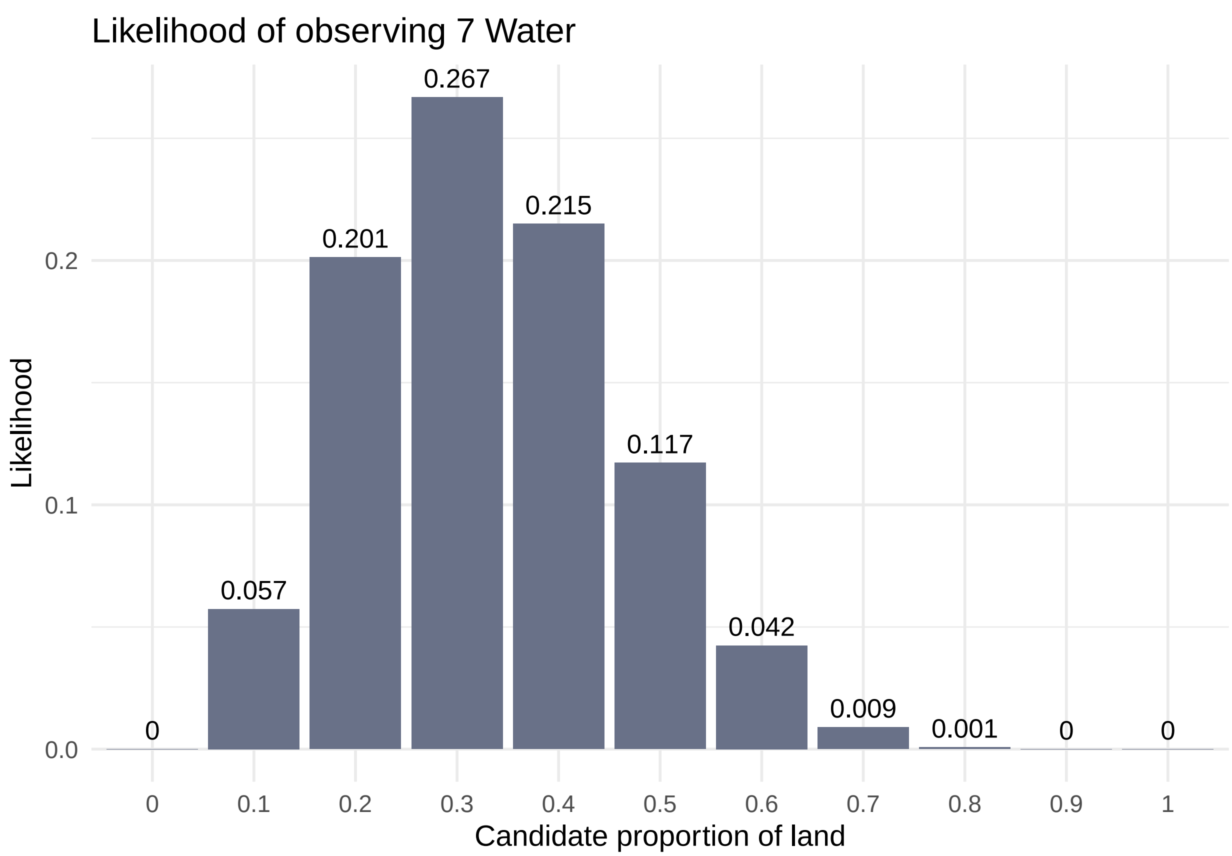
Assume you observe 7 water. Take the candidate proportions of land `cp =` 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. For each of these candidates, compute the probability of observing 7 water. Report the results in a summary table and a plot and indicate whether the plot shows a probability distribution or a likelihood function.

```
cp_land <- seq(0, 1, .1)
cp_water <- 1 - cp_land
water <- 7 # number of water observed

# probability of observing 7 water for each candidate
prob_water <- dbinom(water, size = n, prob = cp_water)
water_summary <- data.frame(cp_land = cp_land, prob_water = prob_water) %>%
  arrange(cp_land)

water_summary
```

```
##    cp_land  prob_water
## 1      0.0 0.000000000
## 2      0.1 0.057395628
## 3      0.2 0.201326592
## 4      0.3 0.266827932
```

```
## 5       0.4 0.214990848
## 6       0.5 0.117187500
## 7       0.6 0.042467328
## 8       0.7 0.009001692
## 9       0.8 0.000786432
## 10      0.9 0.000008748
## 11      1.0 0.000000000
```

```r
ggplot(water_summary, aes(x = as.factor(cp_land), y = prob_water)) +
  geom_bar(stat = "identity", fill = "#697188") +
  geom_text(aes(label = round(prob_water, 3)), vjust = -0.5, size = 3.5) +
  theme_minimal() +
  labs(
    title = "Likelihood of observing 7 Water",
    x = "Candidate proportion of land",
    y = "Likelihood"
  )
```



This plot shows a likelihood function of observing 7 water for each candidate proportion of land.

Task 2.3

For each candidate proportion of land, compute the probability of all possible number of occurrences of land. Report the results in a summary table, showing the probability distributions as columns and the likelihood functions as rows.

```r
cp_land <- seq(0, 1, .1) # repeat not to search above

probabilities_all <- data.frame(matrix(ncol = length(cp_land), nrow = 0)) # create empty df
for (i in 1:n) {
  probabilities_all <- rbind(probabilities_all, dbinom(0:n, size = n, prob = cp_land[i]))
}
colnames(probabilities_all) <- c(cp_land)
probabilities_all
```

```
##               0         0.1          0.2         0.3         0.4         0.5
## 1  1.0000000000 0.000000000 0.0000000000 0.000000000 0.000000000 0.000000000
## 2  0.3486784401 0.387420489 0.1937102445 0.057395628 0.011160261 0.001488035
## 3  0.1073741824 0.268435456 0.3019898880 0.201326592 0.088080384 0.026424115
## 4  0.0282475249 0.121060821 0.2334744405 0.266827932 0.200120949 0.102919345
## 5  0.0060466176 0.040310784 0.1209323520 0.214990848 0.250822656 0.200658125
## 6  0.0009765625 0.009765625 0.0439453125 0.117187500 0.205078125 0.246093750
## 7  0.0001048576 0.001572864 0.0106168320 0.042467328 0.111476736 0.200658125
## 8  0.0000059049 0.000137781 0.0014467005 0.009001692 0.036756909 0.102919345
## 9  0.0000001024 0.000004096 0.0000737280 0.000786432 0.005505024 0.026424115
## 10 0.0000000001 0.000000009 0.0000003645 0.000008748 0.000137781 0.001488035
##            0.6         0.7          0.8         0.9            1
## 1  0.000000000 0.000000000 0.0000000000 0.000000000 0.0000000000
## 2  0.000137781 0.000008748 0.0000003645 0.000000009 0.0000000001
## 3  0.005505024 0.000786432 0.0000737280 0.000004096 0.0000001024
## 4  0.036756909 0.009001692 0.0014467005 0.000137781 0.0000059049
## 5  0.111476736 0.042467328 0.0106168320 0.001572864 0.0001048576
## 6  0.205078125 0.117187500 0.0439453125 0.009765625 0.0009765625
## 7  0.250822656 0.214990848 0.1209323520 0.040310784 0.0060466176
## 8  0.200120949 0.266827932 0.2334744405 0.121060821 0.0282475249
## 9  0.088080384 0.201326592 0.3019898880 0.268435456 0.1073741824
## 10 0.011160261 0.057395628 0.1937102445 0.387420489 0.3486784401
```