Assignment-1 Q&A Bilge Salman (03796071), Salim Kaplan (0378856) GitHub link: https://github.com/salimkaplan/BayesIntro24-Assignments/tree/main

```r
# load packages here
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(tidyr)
library(stats)
library(tinytex)

theme_set(theme_light())
```

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.

```r
# df for all possible combinations of outcomes from throwing three dice
all_combinations <- expand.grid(die1 = 1:6, die2 = 1:6, die3 = 1:6)

# the first and last 10 rows of the df
print(head(all_combinations, 10))
```

```
   die1 die2 die3
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
```

```r
print(tail(all_combinations, 10))
```

```
    die1 die2 die3
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
```

```r
# Number of possible outcomes
num_outcomes <- nrow(all_combinations)
```

```
cat("Number of possible outcomes:", num_outcomes)
```

```
Number of possible outcomes: 216
```

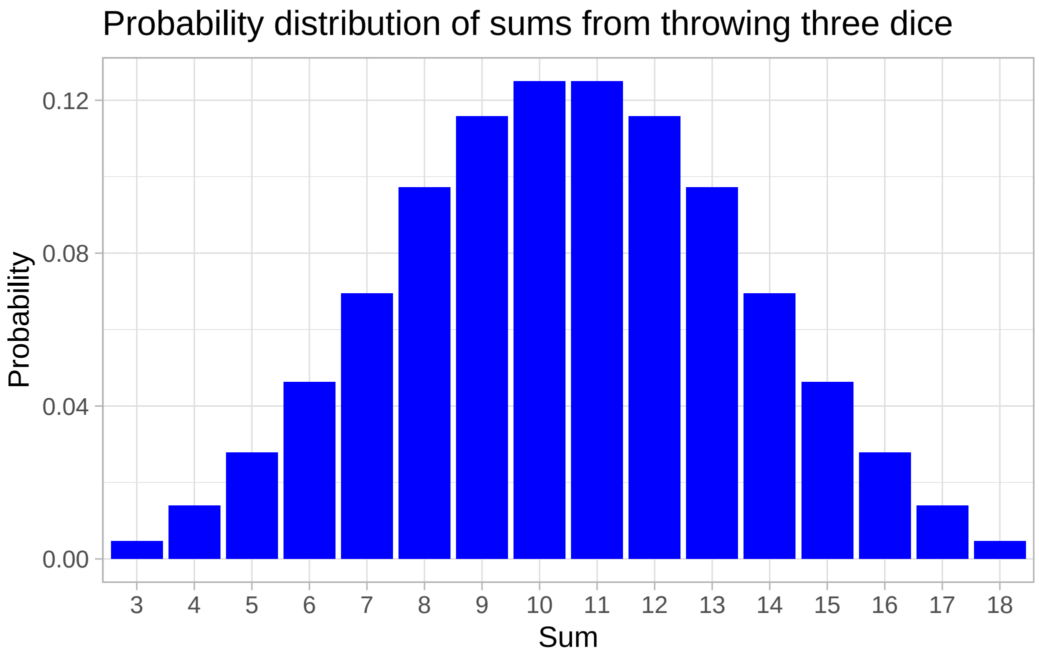
Task 1.2

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).

```
# Calculate the sum of each row
all_combinations$sum <- rowSums(all_combinations)

# Calculate the probabilities of each sum
sum_probabilities <- as.data.frame(table(all_combinations$sum))
colnames(sum_probabilities) <- c("Sum", "Frequency")
sum_probabilities$Probability <- sum_probabilities$Frequency / num_outcomes

# the summary table
sum_probabilities
```

```
   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
```

```r
# plot
ggplot(sum_probabilities, aes(x = Sum, y = Probability)) +
  geom_bar(stat = "identity", fill = 'blue') +
  labs(title = "Probability distribution of sums from throwing three dice",
       x = "Sum",
       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
# filtering the outcomes where at least one die shows a 3
outcomes_with_3 <- subset(all_combinations, die1 == 3 | die2 == 3 | die3 == 3)

# calculating the total number of the outcomes
total_outcomes_with_3 <- nrow(outcomes_with_3)

# outcomes where the sum is greater than 9
outcomes_with_sum_ge_10 <- subset(outcomes_with_3, sum >= 10)
num_outcomes_with_sum_ge_10 <- nrow(outcomes_with_sum_ge_10)
```

```r
# probability calculation
probability_sum_ge_10_given_3 <- num_outcomes_with_sum_ge_10 / total_outcomes_with_3
cat("Probability that the sum is >= 10 given that at least one die shows a 3 is:", round(p
```

```
Probability that the sum is >= 10 given that at least one die shows a 3 is: 0.59
```

Task Set 2

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

Task 2.1

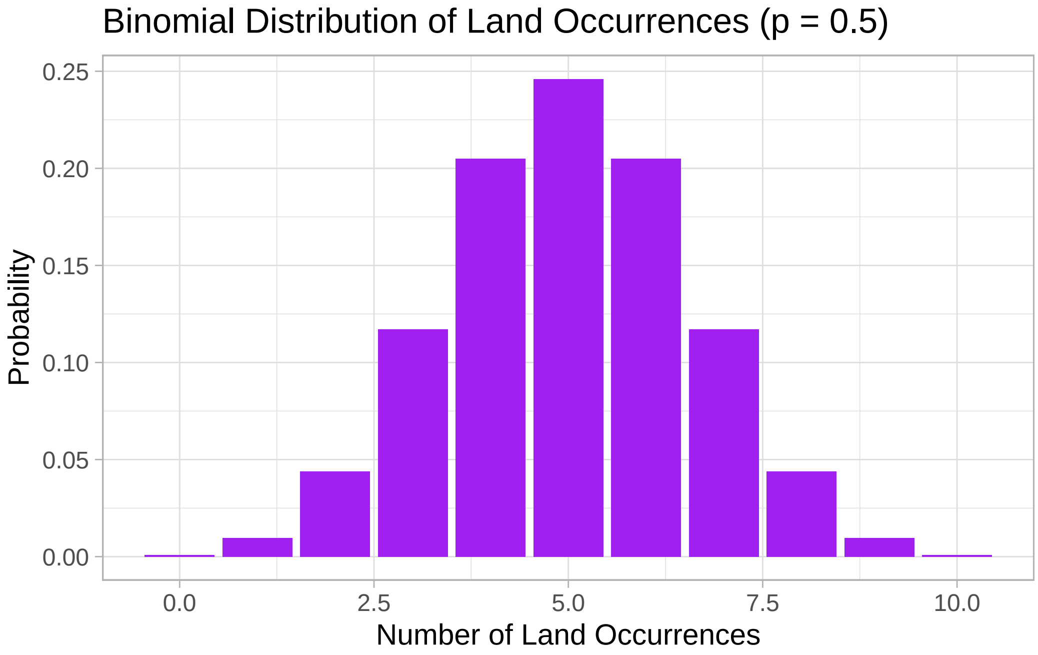
Compute the probability of all possible numbers of occurrence of land, given the candidate proportion of `.5`. Report the results in a summary table and a plot and indicate whether the plot shows a probability distribution or a likelihood function.

```r
# Simulating global tosses with function
simulate_global_tosses <- function(n,p) {

  data <- sample(c("L","W"), size = n, replace= TRUE, prob = c(p,1-p))
  return(data)
}


# Computing the probabilities for all possible numbers of occurrences of land
counter <- function(data,cp){

  n <- length(data)
  L <- sum(data == "L")

  probabilities <- sapply(0:n, function(k) {
    dbinom(k, n, cp)})

  results <- data.frame("Land" = 0:n, "Probability" = probabilities)
  return(results)

}

# we are assuming toss a globe 10 times with 0.5 probability
data <- simulate_global_tosses(10, .5)

# Computing probabilities
summary_table <- counter(data, cp = 0.5)
summary_table
```

```
   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
# Plot the probability distribution
ggplot(summary_table, aes(x = Land , y = Probability)) +
  geom_bar(stat = "identity", fill = "purple") +
  labs(x = "Number of Land Occurrences",
       y = "Probability",
       title = "Binomial Distribution of Land Occurrences (p = 0.5)")
```



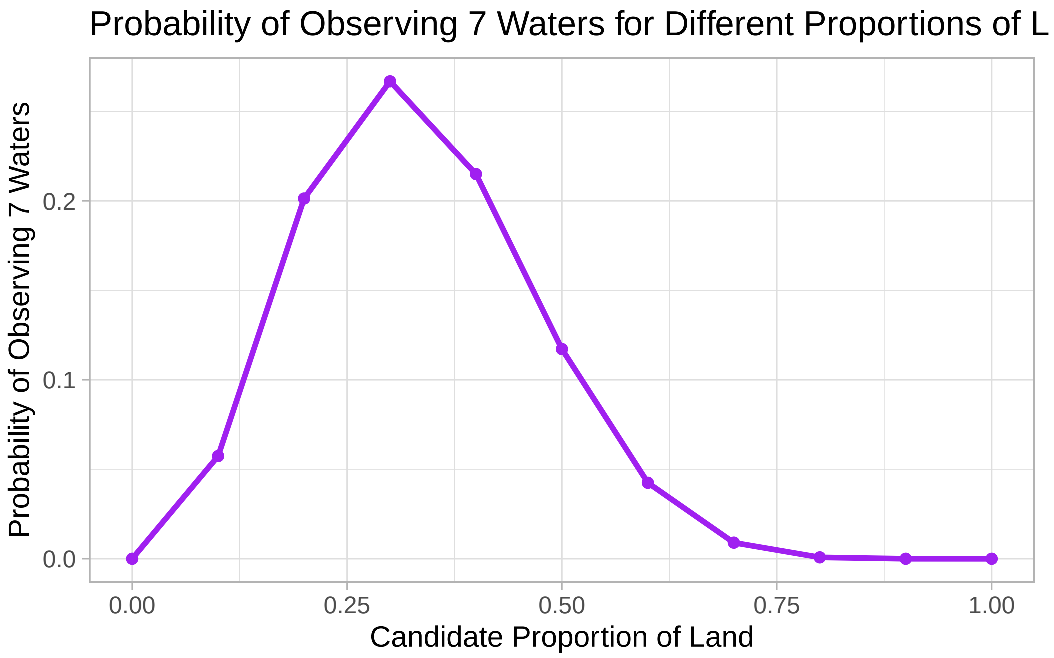
Task 2.2

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.

```
# Computing the probability of observing 7 waters for different candidate proportions of l
counter_2 <- function(n, W) {
  data_2 <-  c("W","W","W","W","W","W","W") # we are assuming we observed 7 water
  W <- sum(data_2 == "W")

  probabilities <- sapply(cp, function(p) {
    dbinom(W, n, 1 - p)
  })
  results <- data.frame("Candidate Proportion" = cp, "Probability" = probabilities, check.
  return(results)
}

# Setting parameters
n <- 10
cp = seq(0,1,0.1)

# Computing probabilities
summary_table_2 <- counter_2(n, W)
summary_table_2
```

```
   Candidate Proportion Probability
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
```

```
# Plot the probability distribution
ggplot(summary_table_2, aes(x = `Candidate Proportion`, y = Probability)) +
  geom_line(linewidth = 1, color = "purple") +
  geom_point(color = "purple") +
  labs(x = "Candidate Proportion of Land",
       y = "Probability of Observing 7 Waters",
       title = "Probability of Observing 7 Waters for Different Proportions 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.

```
counter_3 <- function(n) {

  results_matrix <- matrix(NA, nrow = n + 1, ncol = length(cp))

  for (i in 1:length(cp)) {
    probabilities <- dbinom(0:n, n, cp[i])
    results_matrix[ ,i ] <- probabilities
  }

  results <- as.data.frame(results_matrix)
  rownames(results) <- paste("Land ", 0:n, sep = " ")
  colnames(results) <- cp
  return(results)
}


# Setting parameters
n <- 10
cp = seq(0,1,0.1)

# Compute results
summary_table_3 <- counter_3(n)
summary_table_3
```

```
         0          0.1          0.2          0.3          0.4          0.5
Land  0  1 0.3486784401 0.1073741824 0.0282475249 0.0060466176 0.0009765625
Land  1  0 0.3874204890 0.2684354560 0.1210608210 0.0403107840 0.0097656250
Land  2  0 0.1937102445 0.3019898880 0.2334744405 0.1209323520 0.0439453125
Land  3  0 0.0573956280 0.2013265920 0.2668279320 0.2149908480 0.1171875000
Land  4  0 0.0111602610 0.0880803840 0.2001209490 0.2508226560 0.2050781250
Land  5  0 0.0014880348 0.0264241152 0.1029193452 0.2006581248 0.2460937500
Land  6  0 0.0001377810 0.0055050240 0.0367569090 0.1114767360 0.2050781250
Land  7  0 0.0000087480 0.0007864320 0.0090016920 0.0424673280 0.1171875000
Land  8  0 0.0000003645 0.0000737280 0.0014467005 0.0106168320 0.0439453125
Land  9  0 0.0000000090 0.0000040960 0.0001377810 0.0015728640 0.0097656250
Land  10 0 0.0000000001 0.0000001024 0.0000059049 0.0001048576 0.0009765625
                  0.6          0.7          0.8          0.9 1
Land  0  0.0001048576 0.0000059049 0.0000001024 0.0000000001 0
Land  1  0.0015728640 0.0001377810 0.0000040960 0.0000000090 0
Land  2  0.0106168320 0.0014467005 0.0000737280 0.0000003645 0
Land  3  0.0424673280 0.0090016920 0.0007864320 0.0000087480 0
Land  4  0.1114767360 0.0367569090 0.0055050240 0.0001377810 0
Land  5  0.2006581248 0.1029193452 0.0264241152 0.0014880348 0
Land  6  0.2508226560 0.2001209490 0.0880803840 0.0111602610 0
Land  7  0.2149908480 0.2668279320 0.2013265920 0.0573956280 0
Land  8  0.1209323520 0.2334744405 0.3019898880 0.1937102445 0
Land  9  0.0403107840 0.1210608210 0.2684354560 0.3874204890 0
Land  10 0.0060466176 0.0282475249 0.1073741824 0.3486784401 1
```