Assignment 1

Kerem Karagöz Immanuel Klein

General Information

- **Points**: Assignment 1 comprises of 6 tasks, 2 points each (12 in total). 2 points are obtained for complete and correct answers. 1 point is obtained for a proper approach or if only part of the task is solved.
- **Submission**: Hand in the assignment as a `Markdown` report (RMarkdown or Quarto) rendered as PDF. The PDF report should show the result(s), the code that produced the result(s), and possibly additional text or comment. Also indicate your name. The report should be uploaded on Moodle until Wednesday, May 29, 9:45 am.
- **Code**: Improve the readibility of your code by `#commenting` it.
- **Working in teams**: Everyone needs to hand in a report on Moodle. However, the report can be handed in as a team work (max. 2 people). When working in teams, state at the beginning of the document, who you worked with. It Ideally, teams use GitHub and add a link to the GitHub repository to which both contributed.
- **Document cannot be created**
  - No `LaTeX` installed: Install `TinyTex` (`tinytex::install_tinytex()`) to be able to render the document as PDF.
  - Packages are not loaded: In the first code chunk of the `Markdown` document, load all the packages (`library()`) that you use throughout the document, otherwise the report will not be rendered.
  - Errors: If code produces an error, the `Markdown` report will not be rendered, unless you tell `Markdown` that the respective code chunk should be ignored (not evaluated). In `RMarkdown`, do so via the chunk settings in the top-right corner of a code chunk. In Quarto, do so by including `#| echo: true` and `#| eval: false` in the code chunk (first two lines).
  - In case of further questions, consult the two links above, search the internet, or ask other students. As the very last resort: Hand in a simple `R` script.

```
library(dplyr)
library(ggplot2)
library(tinytex)
```

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.

```
# Define the number of sides of the dice
dice_sides <- 1:6

# Generate all possible sequences of three dice throws
all_combinations <- expand.grid(dice_sides, dice_sides, dice_sides)

# Name the columns
colnames(all_combinations) <- c("Die1", "Die2", "Die3")

# Get the first 10 rows
first_10 <- head(all_combinations, 10)

# Get the last 10 rows
last_10 <- tail(all_combinations, 10)

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

print(first_10)
print(last_10)
print(num_outcomes)
```

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).

```
# Generate all possible outcomes when throwing 3 dice
outcomes <- expand.grid(Die1 = 1:6, Die2 = 1:6, Die3 = 1:6)

# Calculate the sum of each outcome
outcomes$Sum <- outcomes$Die1 + outcomes$Die2 + outcomes$Die3

# Create a summary table for the sums and their probabilities
sum_probabilities <- outcomes %>%
  group_by(Sum) %>%
  summarise(Probability = n() / nrow(outcomes))

print(sum_probabilities)

# Plot the bar chart of the possible sums
ggplot(sum_probabilities, aes(x = Sum, y = Probability)) +
  geom_bar(stat = "identity", fill = "blue", alpha = 0.7) +
  labs(title = "Probability Distribution of Sums for 3 Dice Rolls",
       x = "Sum", y = "Probability") +
  scale_x_continuous(breaks = sum_probabilities$Sum)
```

Task 1.3

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

```
# Filter outcomes where at least one of the dies shows a 3
filtered_outcomes <- outcomes %>%
  filter(Die1 == 3 | Die2 == 3 | Die3 == 3)

# Calculate the probability that the sum of the dies is  10
total_filtered_outcomes <- nrow(filtered_outcomes)
desired_outcomes <- filtered_outcomes %>%
  filter(Sum >= 10)

probability <- nrow(desired_outcomes) / total_filtered_outcomes
```

```
# Probability of the sum is  10, given that at least one of the dice shows a 3
print(paste("Probability:", probability))
```

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.

```
# 10 tosses
n <- 10
# Candidate proportion of land
cp <- 0.5

# Probabilities of all possible numbers of occurence of land
# These are probabilities because we condition on the parameters (cp)
# and vary the outcome (0:n)
probabilities <- dbinom(0:n, size = n, prob = cp)

# Summary table
summary.table <- data.frame(land = 0:n, probability = probabilities)
summary.table

# Plot
ggplot(summary.table, aes(x = land, y = probability)) +
  geom_line(color="blue") +
  geom_point(color="red") +
  labs(title = "Probability Distribution of Land Occurrences",
       x = "Number of Land Occurrences",
       y = "Probability")
```

Task 2.2

Assume you observe `7` water. Take the candidate proportions of land `cp = r seq(0,1,.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.

```
# 10 tosses
n <- 10
# Candidate proportions of land
cp <- seq(0, 1, 0.1)
# Number of land occurrences
land <- 3

# Likelihood for each candidate proportion
# These are likelihoods because we condition on the observed data (land)
# and vary the parameters (cp)
likelihoods <- dbinom(land, size = n, prob = cp)

# Summary table
summary.table <- data.frame(cp = cp, likelihood = likelihoods)
summary.table

# Plot
ggplot(summary.table, aes(x = cp, y = likelihood)) +
  geom_line(color="blue") +
  geom_point(color="red") +
  labs(title = "Likelihood Function for Observing 7 Water (3 Land)",
       x = "Candidate Proportion of Land",
       y = "Likelihood")
```

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.

```
# 10 tosses
n <- 10
# Candidate proportions of land
cp <- seq(0, 1, 0.1)
# Number of land occurrences
land <- seq(0, n, 1)

# Summary table with the first column consisting of occurences of land
summary.table <- data.frame(land = land)

# Go through every value in the candidate proportions
```

```
for(val in cp) {
  # Create a new column with the the proportion as the name.
  # Fill the column with the probabilities,
  # conditioning on the parameter (val) and varying the outcome (land).
  summary.table[[paste("cp = ", val)]] <- dbinom(land, size = n, prob = val)
}

summary.table
```