Assignment 4

EMATM0061: Statistical Computing and Empirical Methods, TB1, 2024

Dr. Rihuan Ke

Introduction

This assignment is mainly based on Lectures 9 and 10. It is recommended that you watch the video lectures before starting.

Create an R Markdown for the assignment

It is a good practice to use R Markdown to organise your code and results. For example, you can start with the template called Assignment02_TEMPLATE.Rmd which can be downloaded via Blackboard. If you try to write mathematical expressions in R Markdown, examples can be found in the document "Assignment_R MarkdownMathformulasandSymbolsExamples.rmd" (under the resource list tab on blackboard course webpage).

You can optionally submit this assignment by 13:00 Monday 14th October, which will help us understand your work but will not count towards your final grade. The submission point can be found under the assignment tab at Blackboards (click the title "Assignment 04").

Load packages

Load the tidyverse package:

```
library(tidyverse)
```

Additionally, download the file called "HockeyLeague.xlsx" from Blackboards which will be needed by some of the questions in this assignment.

1. Tidy data and iteration

Tidy data and iteration has been introduced in Lecture 9.

1.1. Missing data and iteration

In this task we investigate the effect of missing data and writing iterations in R.

(Q1) The following function performs imputation by mean. What library do we need to load to run this function?

```
impute_by_mean<-function(x){
  mu<-mean(x,na.rm=TRUE) # first compute the mean of x
```

```
  impute_f<-function(z){ # coordinate-wise imputation
    if(is.na(z)){
      return(mu) # if z is na replace with mean
      }else{
        return(z) # otherwise leave in place
      }
    }
return(map_dbl(x,impute_f)) # apply the map function to impute across
vector
}
```

(Q2) Create a function called "impute_by_median" which imputes missing values based on the median of the sample, rather than the mean.

You can test your function on the following sample vector:

```
v<-c(1,2,NA,4)
impute_by_median(v)

## [1] 1 2 2 4
```

(Q3) Next generate a data frame with two variables x and y. For our first variable x we have a sequence $(x_1, x_2, \cdots, x_n)$ where $x_1 = 0, x_n = 10$ and for each $i = 1, \cdots, n-1, x_{i+1} = x_i + 0.1$. For our second variable y we set $y_i = 5x_i + 1$ for $i = 1, \cdots, n$. Generate data of this form and place within a data frame called "df_xy".

```
df_xy %>% head(5)

##     x   y
## 1 0.0 1.0
## 2 0.1 1.5
## 3 0.2 2.0
## 4 0.3 2.5
## 5 0.4 3.0
```

(Q4)

map2: The "map2()" function is similar to the "map()" function but iterates over two variables in parallel rather than one. You can learn more here https://purrr.tidyverse.org/reference/map2.html. The following simple example shows you how "map2_dbl()" can be combined with the "mutate()" function.

```
df_xy %>%
  mutate(z=map2_dbl(x,y,~.x+.y)) %>%
  head(5)

##     x   y   z
## 1 0.0 1.0 1.0
## 2 0.1 1.5 1.6
## 3 0.2 2.0 2.2
## 4 0.3 2.5 2.8
## 5 0.4 3.0 3.4
```

We will now use **map2_dbl()** to generate a new data frame with missing data.

First create a function "sometimes_missing" with two arguments: "index" and "value". The "function" should return "NA" if index is divisible by 5 and return "value" otherwise.

Your function should produce the following outputs:

```
sometimes_missing(14,25)

## [1] 25

sometimes_missing(15,25)

## [1] NA
```

Next generate a new data frame called "df_xy_missing" with two variables x and y, but some missing data. For the first variable x we have a sequence $(x_1, \cdots, x_n)$, which is precisely the same as with "df_xy". For the second variable y we have a sequence $(y_1, \cdots, y_n)$ where y_i = NA if i is divisible by 5 and otherwise $y_i = 5x_i + 1$. To generate the data frame "d_xy_missing" you may want to make use of the functions "row_number(), map2_dbl(), mutate()" as well as "sometimes_missing()".

Check that the first ten rows of your data frame are as follows:

```
df_xy_missing %>% head(10)

##      x   y
## 1  0.0 1.0
## 2  0.1 1.5
## 3  0.2 2.0
## 4  0.3 2.5
## 5  0.4  NA
## 6  0.5 3.5
## 7  0.6 4.0
## 8  0.7 4.5
## 9  0.8 5.0
## 10 0.9  NA
```

(Q5) Create a new data frame "df_xy_imputed" with two variables x and y. For the first variable x we have a sequence $(x_1, \cdots, x_n)$, which is precisely the same as with "df_xy". For the second variable y we have a sequence $(y'_1, \cdots, y'_n)$ which is formed from $(y_1, \cdots, y_n)$ by imputing all its missing values with the median. To generate "df_xy_imputed" from "df_xy_missing" by applying a combination of the functions "mutate()" and "impute_by_median()".

The first part of the data frame should look like this:

```
##     x    y
## 1 0.0  1.0
## 2 0.1  1.5
## 3 0.2  2.0
```

```
## 4 0.3  2.5
## 5 0.4 26.0
## 6 0.5  3.5
```

1.2 Tidying data with pivot functions

In this task you will read in data from a spreadsheet and apply some data wrangling tasks to tidy that data.

First download the excel spreadsheet entitled ""HockeyLeague.xlsx"". The excel file contains two spread-sheets - one with the wins for each team and one with the losses for each team. To read this spreadsheet into R we shall make use of the"readxl" library. You may need to install the library:

install.packages("readxl")

The following code shows how to read in a sheet within an excel file as a data frame. You will need to edit the "folder_path" variable to be the directory which contains your copy of the spreadsheet

```
library(readxl) # load the readxl library
folder_path <- "./"
#folder_path<-"C:/Users/" # set this to the name of the
# directory containing "HockeyLeague.xlsx"
file_name<-"HockeyLeague.xlsx" # set the file name
file_path<-paste(folder_path,file_name,sep="") # create the file_path
wins_data_frame<-read_excel(file_path,sheet="Wins") # read of a sheet
from an xl file
```

Inspect the first 3 rows of the first five columns:

```
wins_data_frame %>%
  select(1:5)%>%
  head(3)
```

```
## # A tibble: 3 × 5
##   ...1   `1990`   `1991`   `1992`   `1993`
##   <chr>  <chr>    <chr>    <chr>    <chr>
## 1 Ducks  30 of 50 11 of 50 30 of 50 12 of 50
## 2 Eagles 24 of 50 12 of 50 37 of 50 14 of 50
## 3 Hawks  20 of 50 22 of 50 33 of 50 11 of 50
```

A cell value of the form "a of b" means that games were won out of a total of b for that season. For example, the element for the "Ducks" row of the "1990" column is "30 of 50" meaning that 30 out of 50 games were won that season.

Is this tidy data?

(Q1) Now apply your data wrangling skills to transform the ""wins_data_frame"" data frame object into a data frame called ""wins_tidy"" which contains the same information but has just four columns entitled ""Team"",""Year"",""Wins"",""Total"".

The””Team”” column should contain the team name, the ””Year”” column should contain the year, the ””Wins”” column should contain the number of wins for that season and the ””Total”” column the total number of games for that season. The first column should be of character type and the remaining columns should be of integer type. You can do this by combining the following functions: “rename()”, “pivot_longer()”, “mutate()” and “separate()”.

You can check the shape of your data frame and the first five rows as follows:

```
wins_tidy %>% dim() # check the dimensions

## [1] 248   4

wins_tidy%>%head(5) # inspect the top 5 rows

## # A tibble: 5 × 4
##   Team   Year  Wins Total
##   <chr> <int> <int> <int>
## 1 Ducks  1990    30    50
## 2 Ducks  1991    11    50
## 3 Ducks  1992    30    50
## 4 Ducks  1993    12    50
## 5 Ducks  1994    24    50
```

(Q2) The ””HockeyLeague.xlsx”” also contains a sheet with the losses for each team by season. Apply a similar procedure to read the data from this sheet and transform that data into a data frame called ””losses_tidy”” with four columns: ””Team””,””Year””,””Losses””,””Total”” which are similar to those in the ””wins_tidy”” data frame except for the ””Losses”” column gives the number of losses for a given season and team, rather than the number of losses.

Your results should look like this:

```
losses_tidy %>% head(5)

## # A tibble: 5 × 4
##   Team   Year Losses Total
##   <chr> <int>  <int> <int>
## 1 Ducks  1990     20    50
## 2 Ducks  1991     37    50
## 3 Ducks  1992      1    50
## 4 Ducks  1993     30    50
## 5 Ducks  1994      7    50
```

You may notice that the number of wins plus the number of losses for a given team, in a given year does not add up to the total. This is because some of the games are neither wins nor losses but draws. That is, for a given year the number of draws is equal to the total number of games minus the sum of the wins and losses.

(Q3) Now combine your two data frames, ””wins_tidy”” and ””losses_tidy””, into a single data frame entitled””hockey_df”” which has 248 rows and 9 columns: A

““Team”” column which gives the name of the team as a character, the ““Year”” column which gives the season year, the ““Wins”” column which gives the number of wins for that team in the given year, the “Losses” column which gives the number of losses for that team in the given year and the ““Draws”” column which gives the number of draws for that team in the given year, the ““Wins_rt”” which gives the wins as a proportion of the total number of games (ie. “Wins/Total”) and similarly the ““Losses_rt”” and the ““Draws_rt”” which gives the losses and draws as a proportion of the total, respectively. To do this you can make use of the “mutate()” function. You may also want to utilise the “across()” function for a slightly neater solution.

The top five rows of your data frame should look as follows:

```
hockey_df %>% head(5)
```

```
## # A tibble: 5 × 9
##   Team   Year  Wins Total Losses Draws Wins_rt Losses_rt Draws_rt
##   <chr> <int> <int> <int>  <int> <int>   <dbl>     <dbl>    <dbl>
## 1 Ducks  1990    30    50     20     0    0.6       0.4      0
## 2 Ducks  1991    11    50     37     2    0.22      0.74     0.04
## 3 Ducks  1992    30    50      1    19    0.6       0.02     0.38
## 4 Ducks  1993    12    50     30     8    0.24      0.6      0.16
## 5 Ducks  1994    24    50      7    19    0.48      0.14     0.38
```

(Q4) To conclude this task generate a summary data frame which displays, for each team, the median win rate, the mean win rate, the median loss rate, the mean loss rate, the median draw rate and the mean draw rate. The number of rows in your summary should equal the number of teams. These should be sorted in descending order or median win rate. You may want to make use of the following functions: “select()”, “group_by()”, “across()”, “arrange()”.

```
## # A tibble: 8 × 7
##   Team         W_md  W_mn  L_md  L_mn  D_md  D_mn
##   <chr>       <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Eagles      0.45  0.437 0.25  0.279 0.317 0.284
## 2 Penguins    0.45  0.457 0.3   0.310 0.133 0.232
## 3 Hawks       0.417 0.388 0.233 0.246 0.32  0.366
## 4 Ducks       0.383 0.362 0.34  0.333 0.25  0.305
## 5 Owls        0.32  0.333 0.3   0.33  0.383 0.337
## 6 Ostriches   0.3   0.309 0.4   0.395 0.267 0.296
## 7 Storks      0.3   0.284 0.22  0.283 0.48  0.433
## 8 Kingfishers 0.233 0.245 0.34  0.360 0.4   0.395
```

1.3 Simulation experiments of probabilities

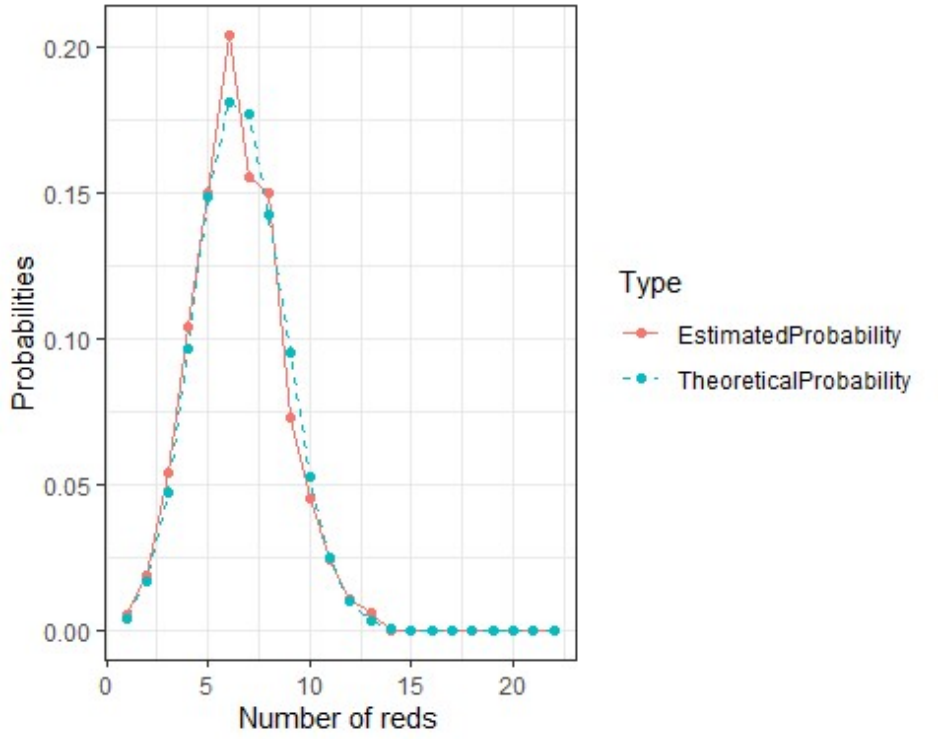
(Q1) The following code was used in last week’s assignment to compare a theoretical probability with an estimated probability in sampling with replacement. Now that we have learnt iterations with the map functions, can you rewrite the code using “map()” (and its variants)?

```
num_red_balls<-3
num_blue_balls<-7
total_draws<-22
prob_red_spheres<-function(z){
  total_balls<-num_red_balls+num_blue_balls
  log_prob<-log(choose(total_draws,z))+
    z*log(num_red_balls/total_balls)+(total_draws-
z)*log(num_blue_balls/total_balls)
  return(exp(log_prob))
}

itermap <- function(.x, .f) {
  result <- list()
  for (item in .x) { result <- c(result, list(.f(item))) }
  return(result)
}

itermap_dbl <- function(.x, .f) {
  result <- numeric(length(.x))
  for (i in 1:length(.x)) { result[i] <- .f(.x[[i]]) }
  return(result)
}

num_trials<-1000 # set the number of trials
set.seed(0) # set the random seed

num_reds_in_simulation <- data.frame(trial=1:num_trials) %>%
  mutate(sample_balls = itermap(.x=trial, function(x){sample(10,22,
replace = TRUE)})) %>%
  mutate(num_reds = itermap_dbl( .x=sample_balls, function(.x)
sum(.x<=3) ) ) %>%
  pull(num_reds)

prob_by_num_reds <- data.frame(num_reds=seq(22)) %>%
  mutate(TheoreticalProbability=prob_red_spheres(num_reds)) %>%
  mutate(EstimatedProbability=
           itermap_dbl(.x=num_reds, function(.x)
sum(num_reds_in_simulation==.x))/num_trials)
```

(Q2) Next, we can make use of "pivot_longer" and "ggplot" to create a plot to compare the theoretical probability with the estimated probability in the data frame "prob_by_num_reds". Your plot should look similar to the one below.

```
prob_by_num_reds %>%
  pivot_longer(cols=c("EstimatedProbability","TheoreticalProbability"),
               names_to="Type",values_to="count") %>%
  ggplot(aes(num_reds,count)) +
  geom_line(aes(linetype=Type, color=Type)) +
geom_point(aes(color=Type)) +
```

```
  scale_linetype_manual(values = c("solid", "dashed"))+
  theme_bw() + xlab("Number of reds") + ylab("Probabilities")
```



Try to make sense of each line in the above code.

2. Conditional probability, Bayes rule and independence

Recall that Bayes theorem helps to "invert" conditional probabilities, and the law of total probability allows us to write an (unconditional) probability in terms of a collection of conditional probabilities.

Bayes theorem

Suppose we have a probability space $(\Omega, \mathcal{E}, \mathbb{P})$. Given events $A, B \in \mathcal{E}$ with $\mathbb{P}(A) > 0$ and $\mathbb{P}(B) > 0$, we have

$$\mathbb{P}(B \mid A) = \frac{\mathbb{P}(B) \cdot \mathbb{P}(A \mid B)}{\mathbb{P}(A)}.$$

The law of total probability

Suppose we have a probability space $(\Omega, \mathcal{E}, \mathbb{P})$, and $A_1, A_2, \cdots \in \mathcal{E}$ forms a partition of Ω. For any event $B \in \mathcal{E}$, we have

$$\mathbb{P}(B) = \sum_i \mathbb{P}(A_i \cap B) = \sum_{\{i:\mathbb{P}(A_i)>0\}} \mathbb{P}(B \mid A_i) \cdot \mathbb{P}(A_i).$$

Independent and dependent events

Let $(\Omega, \mathcal{E}, \mathbb{P})$ be a probability space.

1. A pair of events $A, B \in \mathcal{E}$ are said to be independent if $\mathbb{P}(A \cap B) = \mathbb{P}(A) \cdot \mathbb{P}(B)$.

2. A pair of events $A, B \in \mathcal{E}$ are said to be dependent if $\mathbb{P}(A \cap B) \neq \mathbb{P}(A) \cdot \mathbb{P}(B)$.

2.1 Bayes theorem

(Q1) Let A be the event that it rains next week and B the event that the weather forecaster predicts that there will be rain next week.

Let's suppose that the probability of rain next week is $\mathbb{P}(A) = 0.9$.

Suppose also that the conditional probability that there is a forecast of rain, given that it really does rain, is $P(B|A) = 0.8$.

On the other hand, the conditional probability that there is a forecast of dry weather, given that there really isn't any rain is $P(B^c|A^c) = 0.75$.

Now suppose that there is a forecast of rain. What is the conditional probability of rain, given the forecast of rain $P(A|B)$?

2.2 Conditional probabilities

(Q1) Suppose we have a probability space $\Omega, \mathcal{E}, \mathbb{P}$.

1. Suppose that $A, B \in \mathcal{E}$ and $A \subseteq B$ and $\mathbb{P}(B) \neq 0$. Give an expression for $\mathbb{P}(A|B)$ in terms of $\mathbb{P}(A)$ and $\mathbb{P}(B)$. If additionally, assume $\mathbb{P}(B \setminus A) = 0$, where $B \setminus A$ is the complement of A in B, what is $\mathbb{P}(A|B)$?

2. Suppose that $A, B \in \mathcal{E}$ with $A \cap B = \emptyset$. Give an expression for $\mathbb{P}(A \mid B)$. Note that this is a special case of $\mathbb{P}(A \cap B) = 0$. Does you result still hold for $\mathbb{P}(A \cap B) = 0$?

3. Suppose that $A, B \in \mathcal{E}$ with $B \subseteq A$. Give an expression for $\mathbb{P}(A|B)$ (this question is different from the first one).

4. Suppose that $A \in \mathcal{E}$. Is $\mathbb{P}(A|\Omega)$ equal to $\mathbb{P}(A)$? Why?

5. Show that given three events $A, B, C \in \mathcal{E}$ we have $\mathbb{P}(A \cap B \cap C) = \mathbb{P}(A \mid B \cap C) \cdot \mathbb{P}(B \mid C) \cdot \mathbb{P}(C)$. (Tips: consider $B \cap C$ as a set, denoted by D, and then compute $\mathbb{P}(A \cap D)$). Similarly, can you show that $\mathbb{P}(A \cap B \cap C) = \mathbb{P}(B \mid A \cap C) \cdot \mathbb{P}(A \mid C) \cdot \mathbb{P}(C)$?

6. Show that given three events $A, B, C \in \mathcal{E}$ and $\mathbb{P}(B \cap C) \neq 0$, we have $\mathbb{P}(A|B \cap C) = \frac{\mathbb{P}(B|A \cap C) \cdot \mathbb{P}(A|C)}{\mathbb{P}(B|C)}$.

(Q2) Consider a flight from Bristol to Paris.

1. If it is windy, then the probability of the flight being cancelled is 0.3.
2. If it is not windy, then the probability of the flight being cancelled is 0.1.

The probability that it is windy is 0.2. Calculate the probability that the flight is not cancelled (this is an unconditional probability).

2.3 Mutual independence and pair-wise independent

(Q1) Consider a simple probability space $(\Omega, \mathcal{E}, \mathbb{P})$ with $\Omega = \{(0,0,0), (0,1,1), (1,0,1), (1,1,0)\}$. Since $(\Omega, \mathcal{E}, \mathbb{P})$ is a simple probability space containing four elements we have

$$\mathbb{P}(\{(0,0,0)\}) = \mathbb{P}(\{(0,1,1)\}) = \mathbb{P}(\{(1,0,1)\}) = \mathbb{P}(\{(1,1,0)\}) = 1/4.$$

Consider the events $A := \{(1,0,1), (1,1,0)\}, B := \{(0,1,1), (1,1,0)\}$ and $C := \{(0,1,1), (1,0,1)\}$.

Verify that $\mathbb{P}(A \cap B) = \mathbb{P}(A) \cdot \mathbb{P}(B)$, $\mathbb{P}(A \cap C) = \mathbb{P}(A) \cdot \mathbb{P}(C)$ and $\mathbb{P}(C \cap B) = \mathbb{P}(C) \cdot \mathbb{P}(B)$. Hence, we deduce that the events A, B, C are pair-wise independent.

What is $A \cap B \cap C$? What is $\mathbb{P}(A \cap B \cap C)$? Are the events A, B, C mutually independent?

The goal of this question is to understand the differences between the concepts of mutually independent and pair-wise independent.

~~2.4 The Monty hall problem(*)~~

This is an **optional** question. You might want to return to this after completing the others.

(Q1)

Consider the following game:

At a game show there are three seemingly identical doors. Behind one of the doors is a car, and behind the remaining two is a goat.

1. The contestant of the game first gets to choose one of the three doors. The host then opens one of the other two doors to reveal a goat.
2. The contestant now gets a chance to either (a) switch their choice to the other unopened door or (b) stick to their original choice.
3. The host then opens the door corresponding to the contestant's final choice. They get to keep whatever is behind their final choice of door.

Question: does the contestant improve their chances of winning the car if they switch their choice?

For clarity, we make the following assumptions:

1. The car is assigned to one of the doors at random with equal probability for each door.
2. The assignment of the car and the initial choice of the contestant are independent.
3. Once the contestant makes their initial choice, the host always opens a door which (a) has a goat behind it and (b) is not the contestant's initial choice. If there is more than one such door (i.e. when the contestant's initial choice corresponds to the door with a car behind it) the host chooses at random from the two possibilities with equal probability.

To formalise our problem we introduce the following events for $i = 1,2,3$:

- A_i denotes the event that car is placed behind the i-th door;
- B_i denotes the event that contestant initially chooses the i-th door;
- C_i denotes the event that the host opens the i-th door to reveal a goat.

Consider a situation in which the contestant initially selects the first door (B_1) and then the host opens the second door to reveal a goat (C_2). What is $\mathbb{P}(A_3 \mid B_1 \cap C_2)$?

What does this suggest about a good strategy? Should we switch choices?

(Q2)

Can you carry out a simulation study to simulate the probability of winning a car with the initial choice and the probability of winning a car by switching choice?