Assignment 4

Thrisha Rajkumar

2024-12-28

Q1.

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

```
# Adding tidyverse to run this function
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 become errors
```

```
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.

```
impute_by_median <- function(x) {
  median_of_x <- median(x, na.rm = TRUE)  # first compute the median of x
```

```r
  replace_function <- function(y) {
    if (is.na(y)) {  # correct reference to 'y' instead of 'z'
      return(median_of_x)
    } else {
      return(y)
    }
  }
  return(map_dbl(x, replace_function))  # Apply function to impute missing values
}

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

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

```r
impute_by_median(v)
```

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

Q2.

Question

Generate a data frame `df_xy` with the following specifications:

1. The variable `x` is a sequence defined as:

   - Starting value: 0
   - Ending value: 10
   - Increment: 0.1

2. The variable `y` is calculated as `y = 5x + 1` for each corresponding value of `x`.

3. Place these variables into a data frame called `df_xy`.

4. Display the first few rows of the data frame.

```r
x <- seq(from = 0, to=10, by=0.1)
y <- x*5 + 1
df_xy = data.frame(x,y)
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
```

Q3.

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
```

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 returns value otherwise.

```
sometimes_missing <- function(index, value){
  if(index%%5==0){
    return(NA)
  }else{
    return(value)
  }
}
sometimes_missing(14,25)
```

```
## [1] 25
```

we need to generate "df_xy_missing" with two variables x and y

"d_xy_missing" "row_number(), map2_dbl(), mutate()

```
df_xy_missing <- df_xy %>%
  mutate(y=map2_dbl(row_number(),y,sometimes_missing))
df_xy_missing<-df_xy  %>%
  mutate(y=map2_dbl(.x=row_number(),.y=y,sometimes_missing))
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.

(Q5) Create a new data frame "df_xy_imputed" with two variables and . For the first variable we have a sequence (, ,), which is precisely the same as with "df_xy". For the second variable we have a sequence (, ,) which is formed from (, ,) by imputing any 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()".

```
df_xy_impute<-df_xy_missing %>%
mutate(y=impute_by_median(y))
head(df_xy_impute)
```

```
##     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
```

Tidying data with pivot functions

""HockeyLeague.csv" " downloaded

```
library(readxl) # load the readxl library
```

```
## Warning: package 'readxl' was built under R version 4.4.2
```

```
folder_path <- "C:/Users/pc/Desktop/R-Programming - SCEM/Assignment 4"
file_name <- "HockeyLeague.xlsx"
file_path <- paste0(folder_path, "/", file_name)

wins_data_frame <- read_excel(file_path, sheet="Wins") # read a sheet from an xl file
```

```
## New names:
## * `` -> `...1`
```

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

```
## # A tibble: 3 x 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
```

(Q1)

```r
w_l_narrow<-function(w_or_l){
return(
read_excel(file_path,sheet=w_or_l)%>%
rename(Team=...1)%>%
pivot_longer(!Team,names_to="Year",values_to="val")%>%
mutate(Year=as.integer(Year))%>%
separate(col=val,into=c(w_or_l,"Total"),sep=" of ",convert=TRUE) )
}
wins_tidy<-w_l_narrow(w_or_l="Wins")
```

```
## New names:
## * `` -> `...1`
```

```r
wins_tidy %>% dim()
```

```
## [1] 248   4
```

```r
wins_tidy%>%head(5)
```

```
## # A tibble: 5 x 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
```

```r
losses_tidy<-w_l_narrow(w_or_l="Losses")
```

(Q2)

```
## New names:
## * `` -> `...1`
```

```r
losses_tidy %>% head(5)
```

```
## # A tibble: 5 x 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
```

(Q3)

```
hockey_df<-inner_join(wins_tidy,losses_tidy)%>%
  mutate(Draws=Total-Wins-Losses)%>%
  mutate(across(starts_with(c("Wins","Losses","Draws")),~.x/Total, .names="{.col}_rt"))
```

```
## Joining with `by = join_by(Team, Year, Total)`
```

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

```
## # A tibble: 5 x 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)

```
hockey_df %>%
  select(-Wins,-Draws,-Losses) %>%
  group_by(Team) %>%
  summarise(across(starts_with(c("Wins","Losses","Draws")),
                   list(md=median,mn=mean),
                   .names="{substring(.col,1,1)}_{.fn}")) %>%
  arrange(desc(W_md))
```

```
## # A tibble: 8 x 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
```

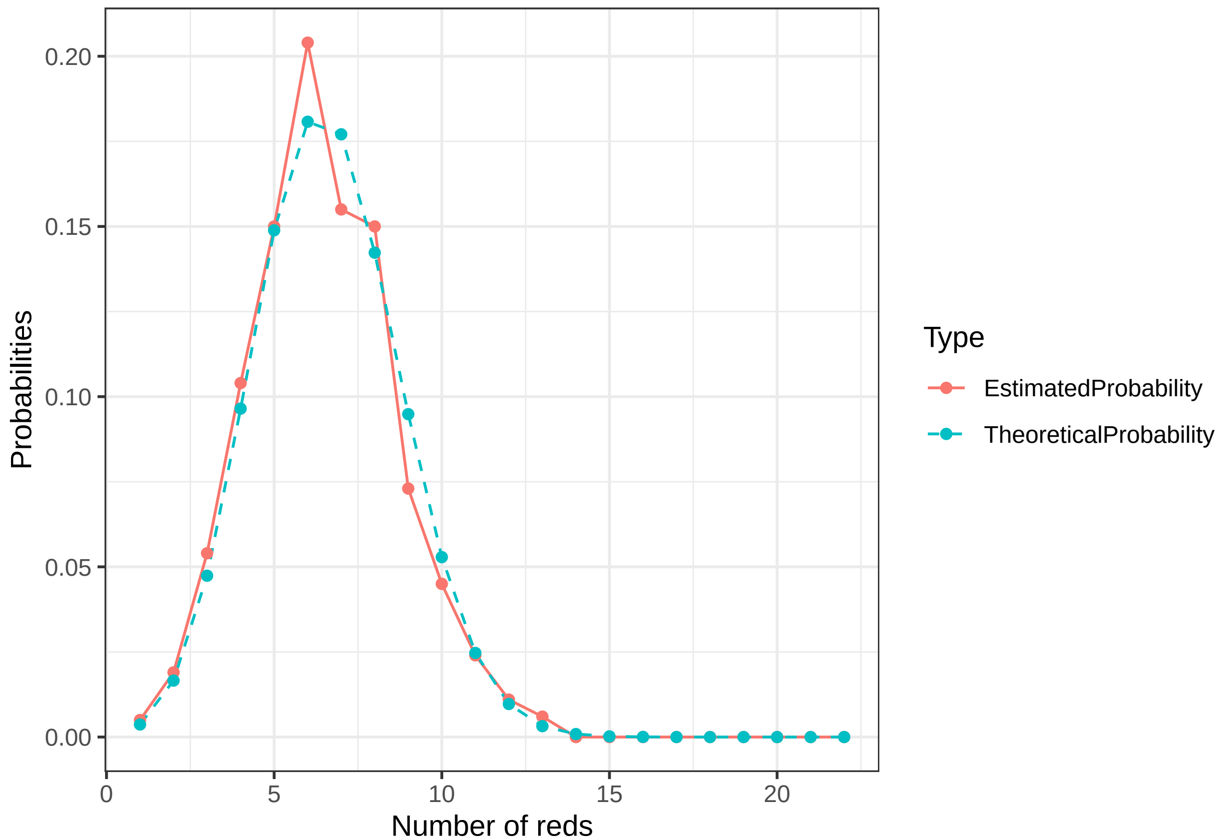
1.3 Simulation experiments of probabilities

```
num_red_balls<-3
num_blue_balls<-7
total_draws<-22
prob_red_spheres<-function(z){
```

```r
  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)
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))
}

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 = map(.x=trial, ~sample(10,22, replace = TRUE))) %>%
  mutate(num_reds = map_dbl( .x=sample_balls, ~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=map_dbl(.x=num_reds, ~sum(num_reds_in_simulation==.x))/num_trials)
```

Q2

```
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")
```



2. Conditional probability, Bayes rule and independence

Bayes' Theorem

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

the law of total probability:

$$\mathbb{P}(B) = \sum_i \mathbb{P}(A_i \cap B) = \sum_i \mathbb{P}(B \mid A_i) \cdot \mathbb{P}(A_i)$$

Independent and Dependent Events

Let $(\Omega, \mathcal{E}, \mathbb{P})$ be a probability space. We define the following:

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)

```
p_A<-0.9
p_B_given_A<-0.8
p_not_B_given_not_A<-0.75
p_B<-p_B_given_A*p_A+(1-p_not_B_given_not_A)*(1-p_A)
p_A_given_B<-p_B_given_A*p_A/p_B
p_A_given_B
```

```
## [1] 0.966443
```

2.2 Conditional probabilities

(Q1)

Conditional Probabilities

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

(a) Expression for $\mathbb{P}(A \mid B)$ when $A \subseteq B$ and $\mathbb{P}(B) \neq 0$ We are given $A \subseteq B$, which implies that $A \cap B = A$. By the definition of conditional probability, we have:

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

Thus, the conditional probability of A given B is:

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

(b) If additionally, $\mathbb{P}(B \setminus A) = 0$, what is $\mathbb{P}(A \mid B)$? If $\mathbb{P}(B \setminus A) = 0$, then $B \subseteq A$, which means that $A = B$. Therefore, we have:

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(A)}{\mathbb{P}(B)} = 1$$

Thus, $\mathbb{P}(A \mid B) = 1$ when $B \subseteq A$ and $\mathbb{P}(B \setminus A) = 0$.

(c) Suppose $A \cap B = \emptyset$. What is $\mathbb{P}(A \mid B)$? If $A \cap B = \emptyset$, then $\mathbb{P}(A \cap B) = 0$. The conditional probability $\mathbb{P}(A \mid B)$ is given by:

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)} = \frac{0}{\mathbb{P}(B)} = 0$$

Thus, $\mathbb{P}(A \mid B) = 0$ when $A \cap B = \emptyset$.

(d) Does the result still hold for $\mathbb{P}(A \cap B) = 0$? If $\mathbb{P}(A \cap B) = 0$, then the result still holds:

$$\mathbb{P}(A \mid B) = 0$$

This is because if $A \cap B = \emptyset$, then $\mathbb{P}(A \mid B) = 0$, and if $\mathbb{P}(A \cap B) = 0$, the numerator of the conditional probability is zero, leading to the same conclusion.

(e) Suppose $B \subseteq A$. What is $\mathbb{P}(A \mid B)$? If $B \subseteq A$, then $A \cap B = B$, and the conditional probability is:

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)} = \frac{\mathbb{P}(B)}{\mathbb{P}(B)} = 1$$

Thus, $\mathbb{P}(A \mid B) = 1$ when $B \subseteq A$.

(f) Is $\mathbb{P}(A \mid \Omega)$ equal to $\mathbb{P}(A)$? Why? Yes, $\mathbb{P}(A \mid \Omega)$ is equal to $\mathbb{P}(A)$. This is because the probability of A conditioned on the entire sample space Ω is simply the probability of A:

$$\mathbb{P}(A \mid \Omega) = \frac{\mathbb{P}(A \cap \Omega)}{\mathbb{P}(\Omega)} = \frac{\mathbb{P}(A)}{1} = \mathbb{P}(A)$$

Thus, $\mathbb{P}(A \mid \Omega) = \mathbb{P}(A)$.

(g) Show that $\mathbb{P}(A \cap B \cap C) = \mathbb{P}(A \mid B \cap C) \cdot \mathbb{P}(B \mid C) \cdot \mathbb{P}(C)$ We want to express $\mathbb{P}(A \cap B \cap C)$ in terms of conditional probabilities. First, let $D = B \cap C$. Then, by the chain rule of probability:

$$\mathbb{P}(A \cap B \cap C) = \mathbb{P}(A \cap D) = \mathbb{P}(A \mid D) \cdot \mathbb{P}(D)$$

Now, $\mathbb{P}(D) = \mathbb{P}(B \cap C) = \mathbb{P}(B \mid C) \cdot \mathbb{P}(C)$. Substituting this into the equation:

$$\mathbb{P}(A \cap B \cap C) = \mathbb{P}(A \mid B \cap C) \cdot \mathbb{P}(B \mid C) \cdot \mathbb{P}(C)$$

(h) Show that $\mathbb{P}(A \cap B \cap C) = \mathbb{P}(B \mid A \cap C) \cdot \mathbb{P}(A \mid C) \cdot \mathbb{P}(C)$ We proceed similarly. Again, let $D = A \cap C$. Then:

$$\mathbb{P}(A \cap B \cap C) = \mathbb{P}(B \cap D) = \mathbb{P}(B \mid D) \cdot \mathbb{P}(D)$$

Now, $\mathbb{P}(D) = \mathbb{P}(A \cap C) = \mathbb{P}(A \mid C) \cdot \mathbb{P}(C)$. Substituting this:

$$\mathbb{P}(A \cap B \cap C) = \mathbb{P}(B \mid A \cap C) \cdot \mathbb{P}(A \mid C) \cdot \mathbb{P}(C)$$

(i) Show that if $\mathbb{P}(B \cap C) \neq 0$, we have:

$$\mathbb{P}(A \mid B \cap C) = \frac{\mathbb{P}(A \cap B \cap C)}{\mathbb{P}(B \cap C)} = \mathbb{P}(A \mid B \cap C)$$

This result is consistent with the previous findings, as it shows the relationship between conditional probabilities when the intersection of B and C is nonzero.

(Q2)

Conditional Probability Example

Let A be the event that the flight is **not cancelled**, and B be the event that it is **windy**.

We are given the following conditional probabilities:

$$\mathbb{P}(A \mid B) = 1 - 0.3 = 0.7$$

and

$$\mathbb{P}(A \mid B^c) = 1 - 0.1 = 0.9$$

where B^c is the complement of B (i.e., the event that it is **not windy**).

By the **law of total probability**, we can express $\mathbb{P}(A)$ as:

$$\mathbb{P}(A) = \mathbb{P}(A \mid B)\mathbb{P}(B) + \mathbb{P}(A \mid B^c)\mathbb{P}(B^c)$$

Substituting the given values:

$$\mathbb{P}(A) = (0.7 \cdot 0.8) + (0.9 \cdot 0.2) = 0.56 + 0.18 = 0.86$$

Thus, the probability that the flight is not cancelled is $\mathbb{P}(A) = 0.86$.

2.3 Mutual independence and pair-wise independent

Q1

Independence of Events Example

We are given the following probabilities:

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

From this, we can deduce:

$$\mathbb{P}(A) = \mathbb{P}(B) = \mathbb{P}(C) = \frac{1}{2}$$

Pairwise Independence

Since each of the intersections $A \cap B$, $A \cap C$, and $B \cap C$ has only one element, we have:

$$\mathbb{P}(A \cap B) = \frac{1}{4} = \left(\frac{1}{2}\right) \cdot \left(\frac{1}{2}\right) = \mathbb{P}(A) \cdot \mathbb{P}(B)$$

$$\mathbb{P}(A \cap C) = \frac{1}{4} = \left(\frac{1}{2}\right) \cdot \left(\frac{1}{2}\right) = \mathbb{P}(A) \cdot \mathbb{P}(C)$$

$$\mathbb{P}(C \cap B) = \frac{1}{4} = \left(\frac{1}{2}\right) \cdot \left(\frac{1}{2}\right) = \mathbb{P}(C) \cdot \mathbb{P}(B)$$

Thus, A, B, and C are **pairwise independent**.

Mutual Independence

On the other hand, we have:

$$A \cap B \cap C = \emptyset \quad \text{and} \quad \mathbb{P}(A \cap B \cap C) = 0$$

Now, we check if the events are mutually independent. The product of the individual probabilities is:

$$\mathbb{P}(A) \cdot \mathbb{P}(B) \cdot \mathbb{P}(C) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{8}$$

Since:

$$\mathbb{P}(A \cap B \cap C) = 0 \quad \text{and} \quad \mathbb{P}(A) \cdot \mathbb{P}(B) \cdot \mathbb{P}(C) = \frac{1}{8}$$

we conclude that A, B, and C are **not mutually independent**.