Name - Naman Dixit

Sap ID- 500125539

Batch - 7 Data science

**Fundamentals of Data Science**

**Experiment 1**

#1

die\_face <- as.integer(readline("Enter a die face (1-6): "))

if (die\_face >= 1 & die\_face <= 6) {

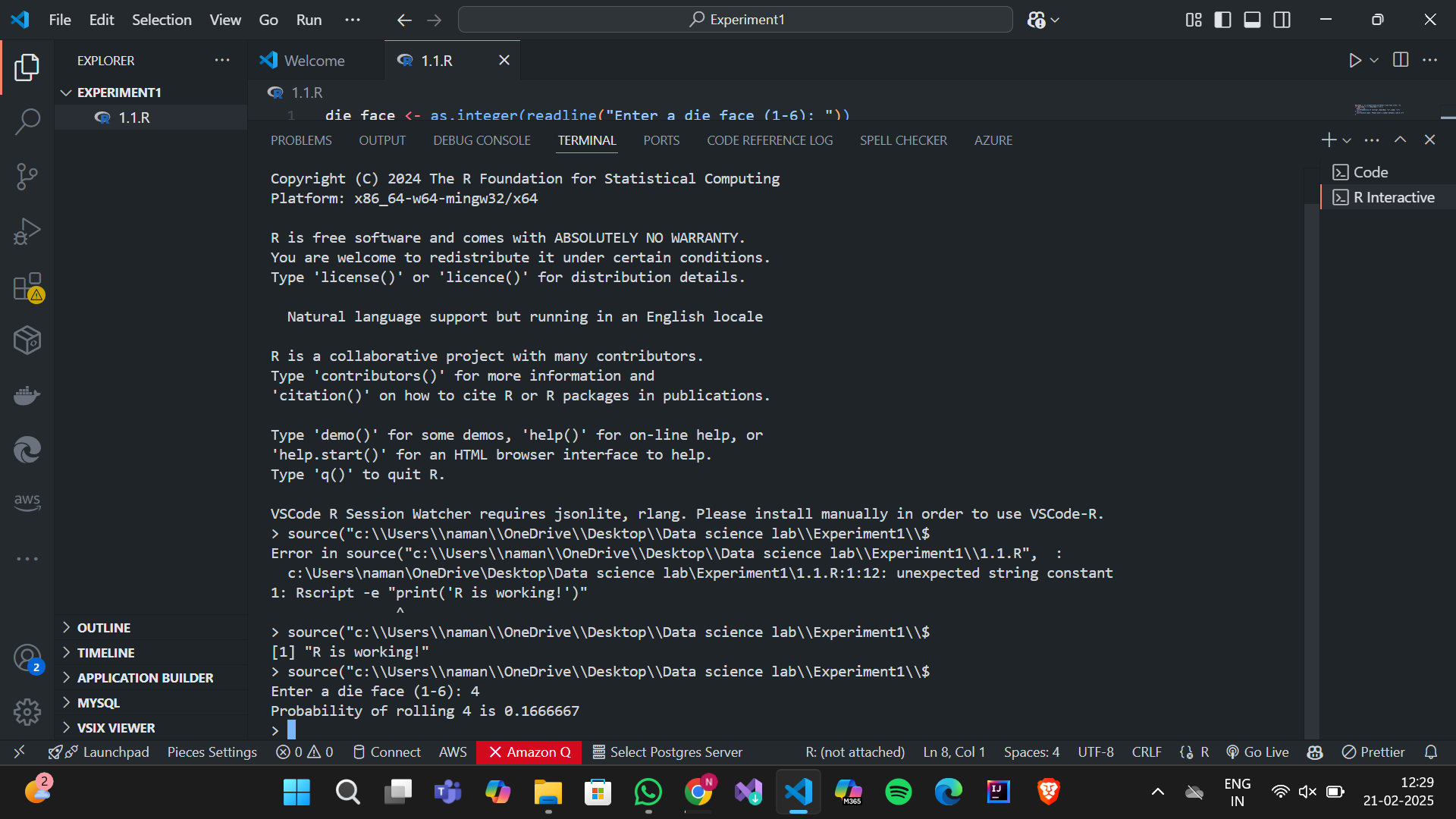
prob <- 1/6

cat("Probability of rolling", die\_face, "is", prob, "\n")

} else {

cat("Invalid input. Please enter a number between 1 and 6.\n")

}



#2

num1 <- as.integer(readline("Enter first number (1-6): "))

num2 <- as.integer(readline("Enter second number (1-6): "))

if (num1 >= 1 & num1 <= 6 & num2 >= 1 & num2 <= 6) {

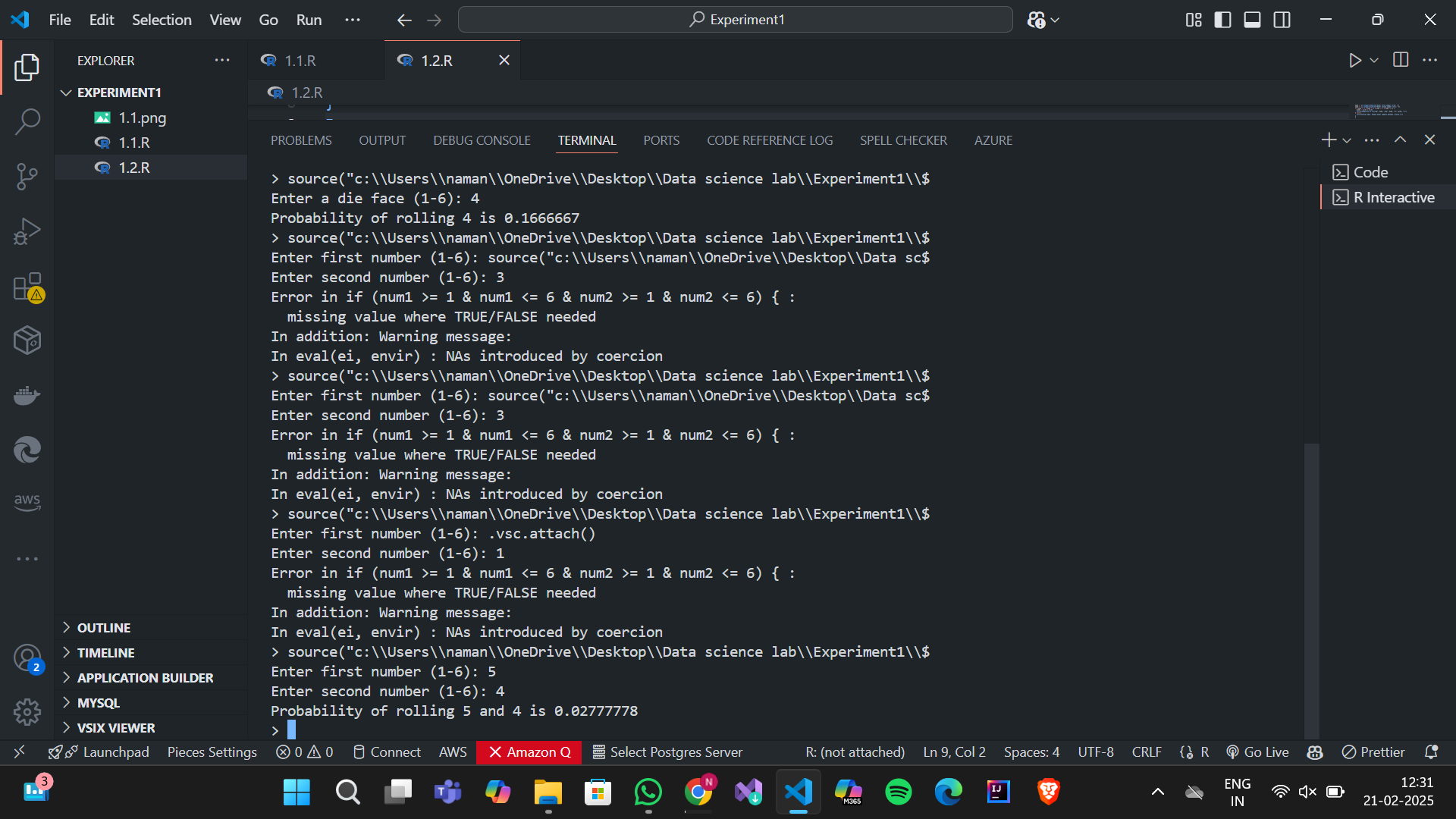
prob <- (1/6) \* (1/6)

cat("Probability of rolling", num1, "and", num2, "is", prob, "\n")

} else {

cat("Invalid input. Please enter numbers between 1 and 6.\n")

}



#3

set1 <- as.integer(strsplit(readline("Enter first set of numbers (comma-separated, 1-6): "), ",")[[1]])

set2 <- as.integer(strsplit(readline("Enter second set of numbers (comma-separated, 1-6): "), ",")[[1]])

if (all(set1 %in% 1:6) & all(set2 %in% 1:6)) {

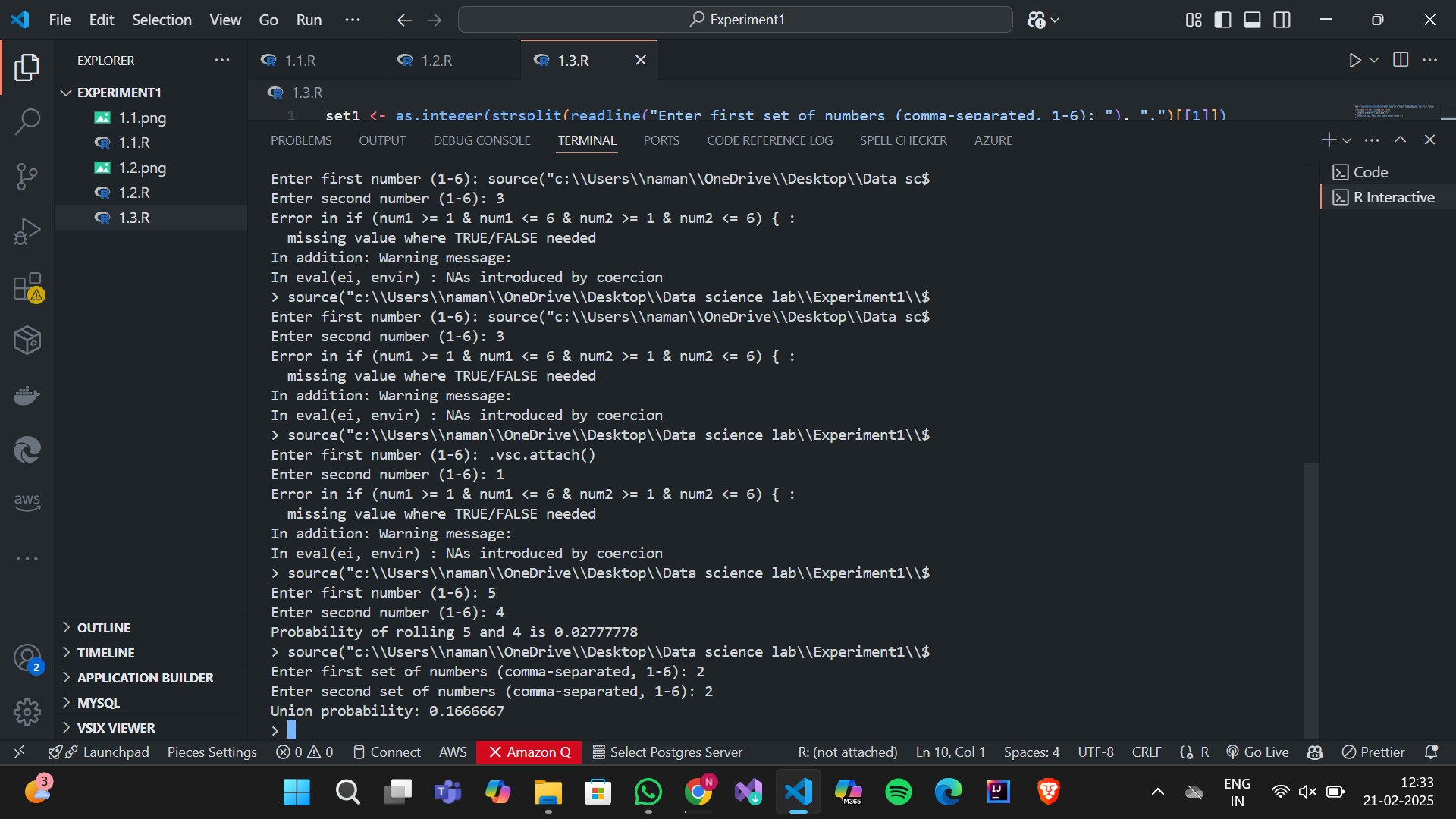
union\_prob <- length(unique(c(set1, set2))) / 6

cat("Union probability:", union\_prob, "\n")

} else {

cat("Invalid input. Please enter numbers between 1 and 6.\n")

}



#4

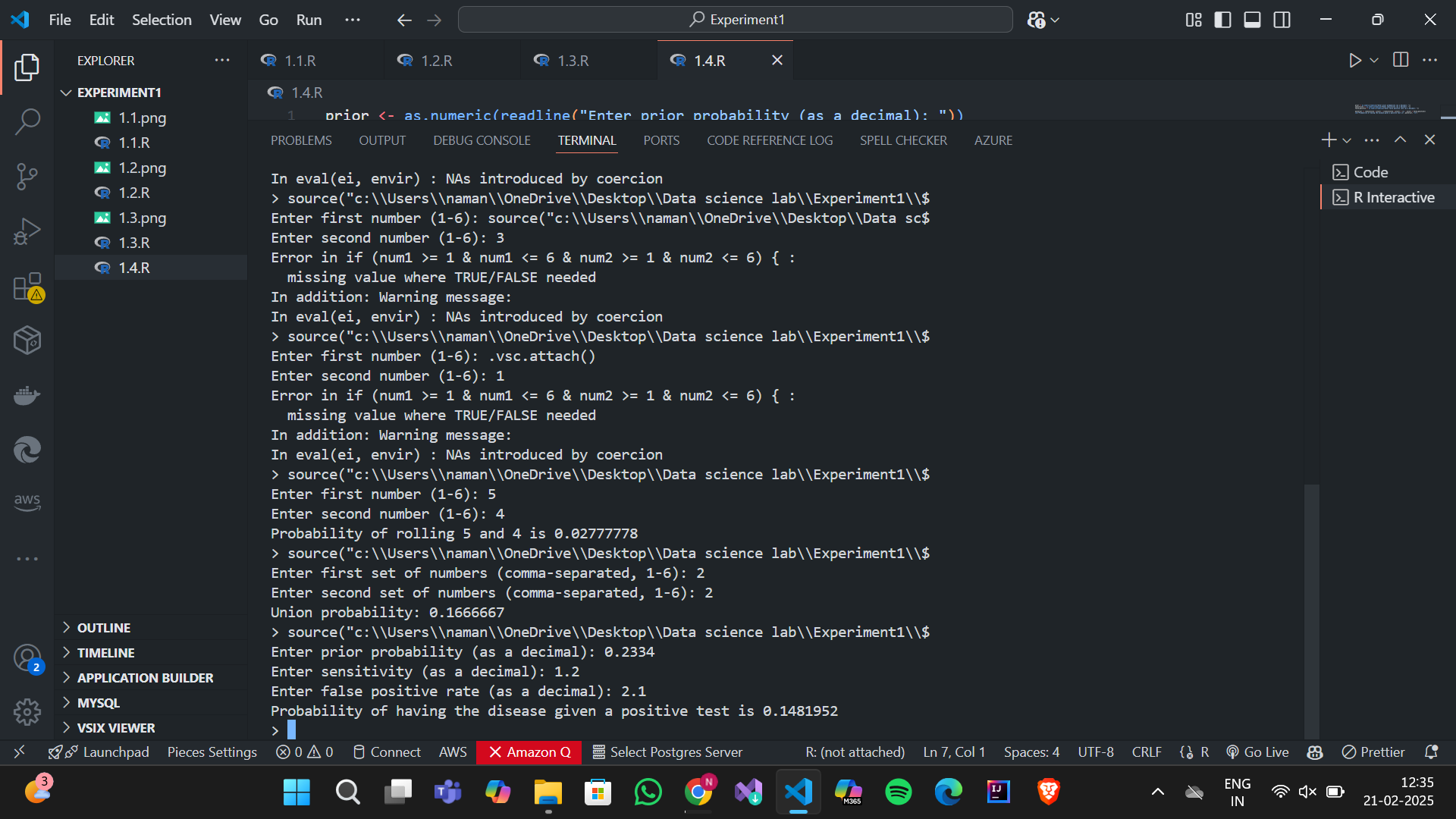
prior <- as.numeric(readline("Enter prior probability (as a decimal): "))

sensitivity <- as.numeric(readline("Enter sensitivity (as a decimal): "))

false\_positive <- as.numeric(readline("Enter false positive rate (as a decimal): "))

posterior <- (sensitivity \* prior) / ((sensitivity \* prior) + ((1 - prior) \* false\_positive))

cat("Probability of having the disease given a positive test is", posterior, "\n")



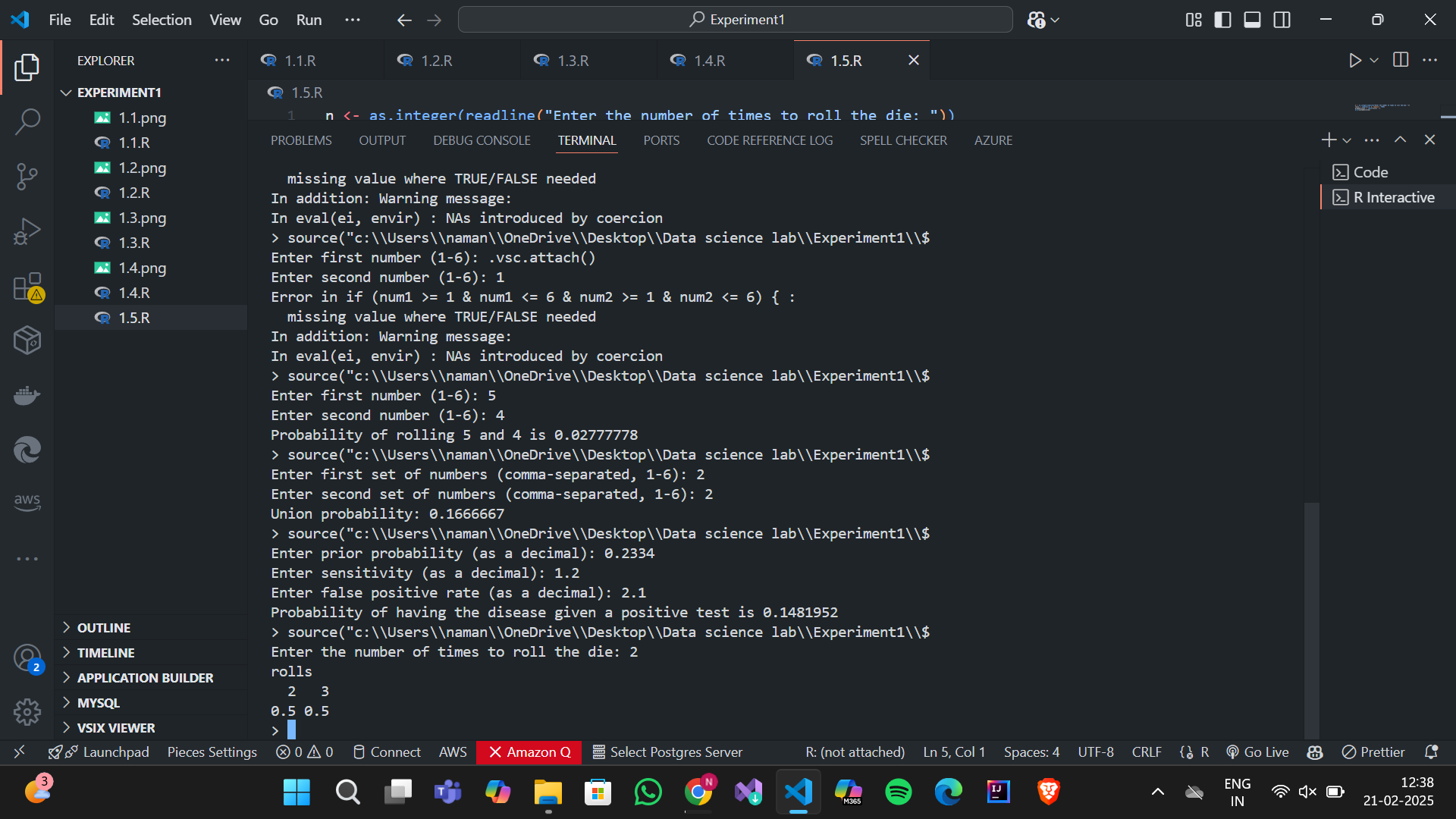
#5

n <- as.integer(readline("Enter the number of times to roll the die: "))

rolls <- sample(1:6, n, replace = TRUE)

probabilities <- table(rolls) / n

print(probabilities)



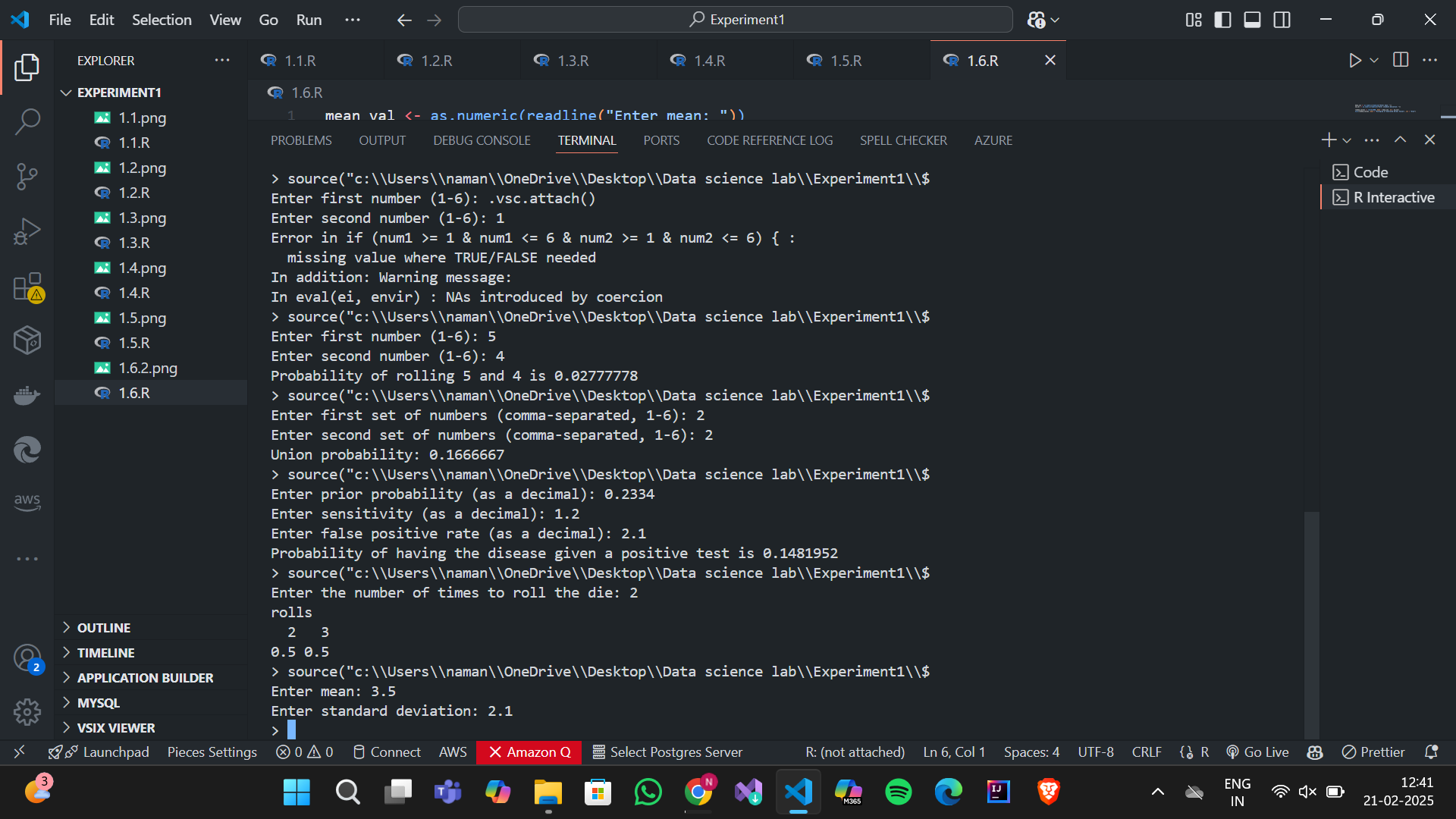
#6

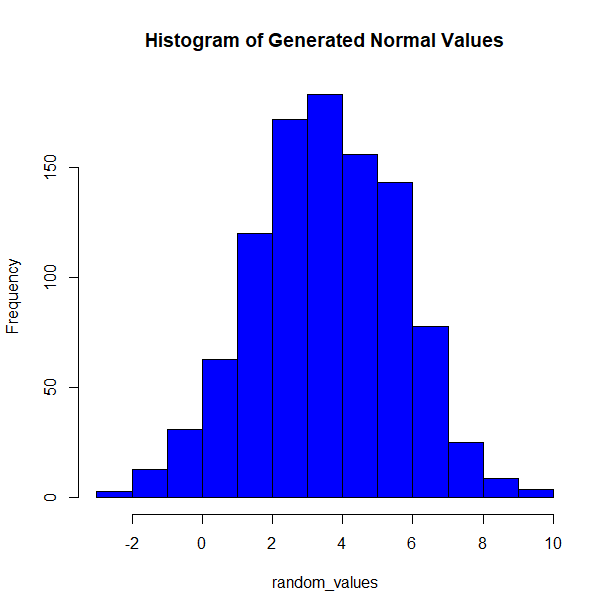
mean\_val <- as.numeric(readline("Enter mean: "))

sd\_val <- as.numeric(readline("Enter standard deviation: "))

random\_values <- rnorm(1000, mean = mean\_val, sd = sd\_val)

hist(random\_values, main = "Histogram of Generated Normal Values", col = "blue")





#7

cost <- as.numeric(readline("Enter cost of game: "))

outcomes <- as.numeric(strsplit(readline("Enter possible outcomes (comma-separated): "), ",")[[1]])

probabilities <- as.numeric(strsplit(readline("Enter their probabilities (comma-separated): "), ",")[[1]])

if (sum(probabilities) == 1) {

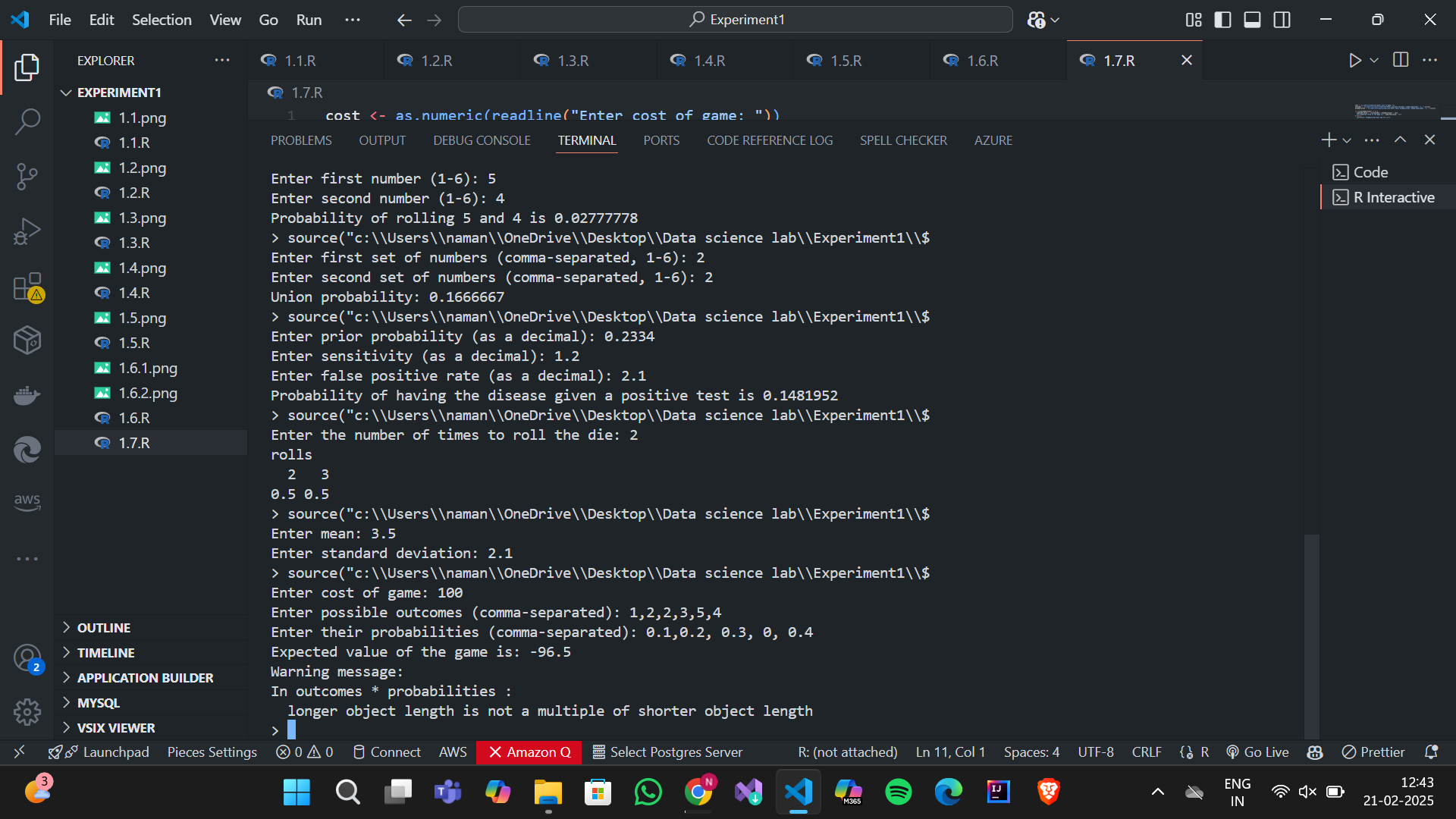
expected\_value <- sum(outcomes \* probabilities) - cost

cat("Expected value of the game is:", expected\_value, "\n")

} else {

cat("Error: Probabilities must sum to 1.\n")

}



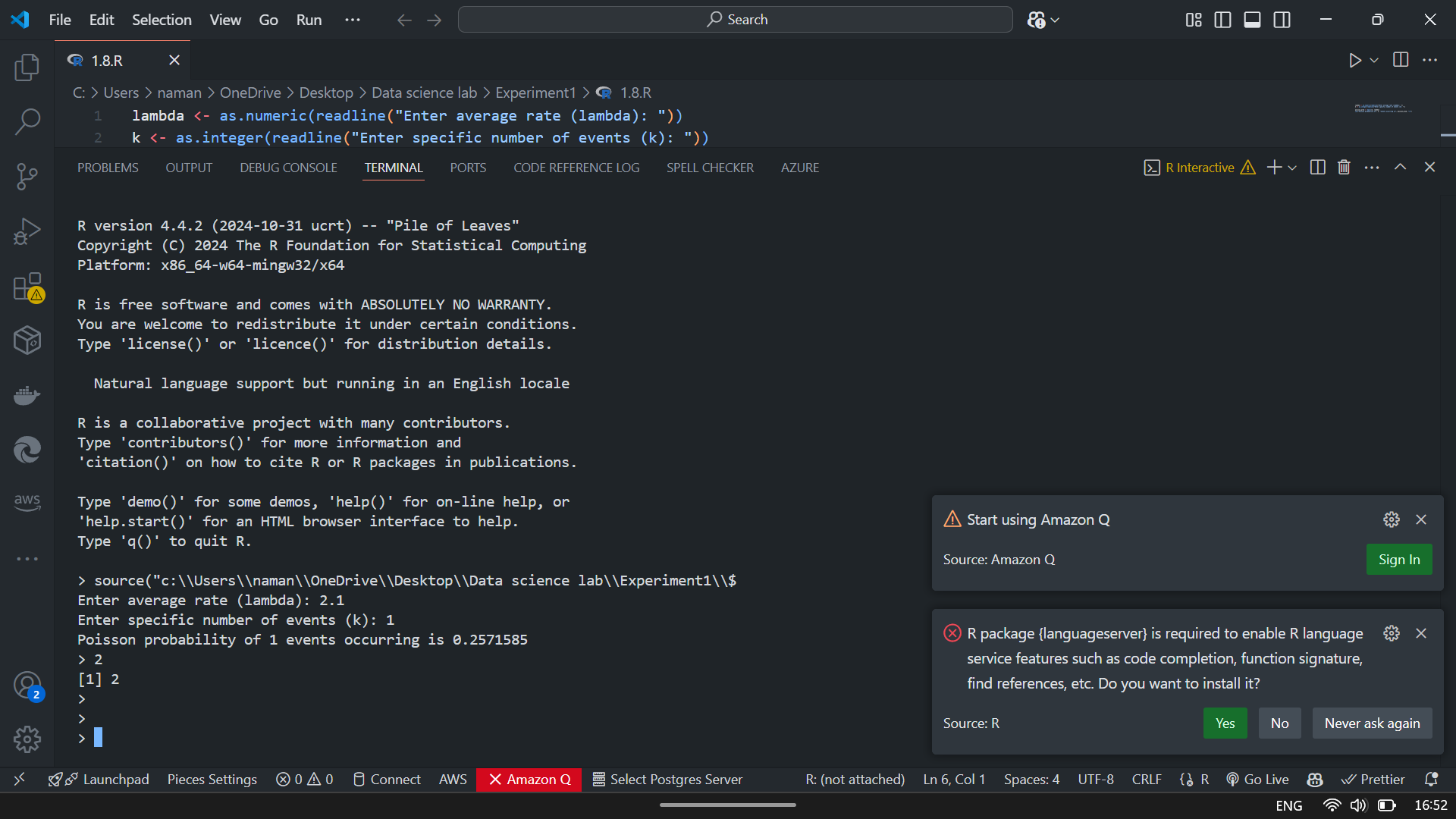
#8

lambda <- as.numeric(readline("Enter average rate (lambda): "))

k <- as.integer(readline("Enter specific number of events (k): "))

poisson\_prob <- dpois(k, lambda)

cat("Poisson probability of", k, "events occurring is", poisson\_prob, "\n")



**Experiment 2**

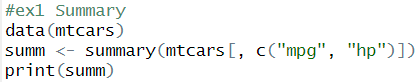
# Objective: To conduct basic data exploration by calculating summary statistics, creating histograms, and generating scatterplots.

1. Summary Statistics for a Dataset

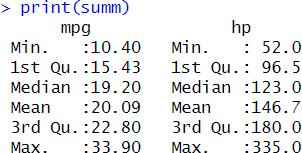
Dataset: Built-in mtcars dataset (Car Specifications)

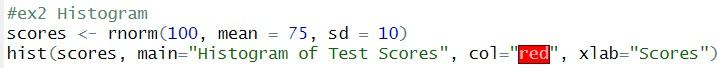
* + Compute summary statistics (mean, median, standard deviation, etc.).
  + Understand the distribution of miles per gallon (mpg) and horsepower (hp).

#INPUT

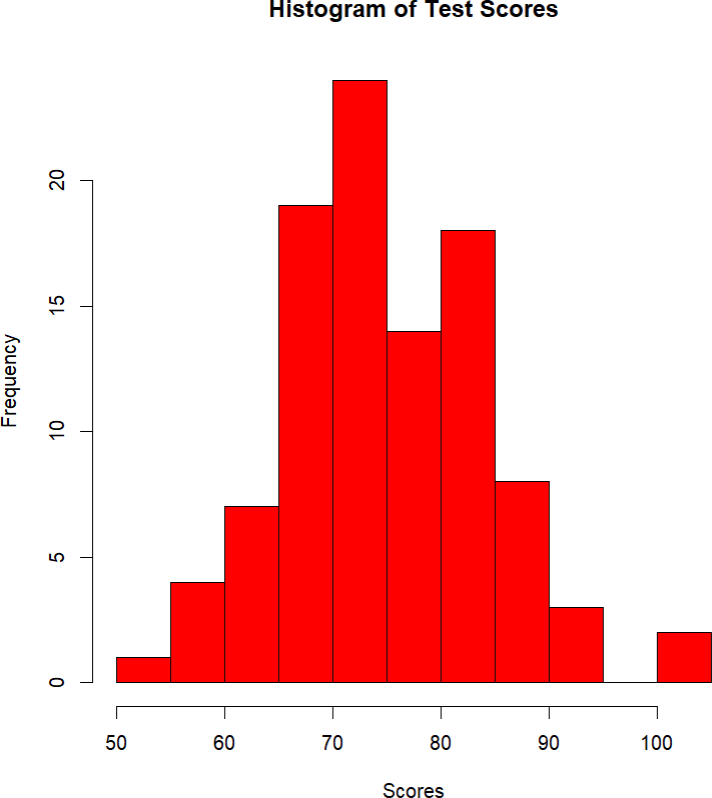


#OUTPUT



1. Create a Histogram
   * Generate a random dataset of students' test scores
   * Visualize data distribution using histograms.
   * Understand skewness and spread of data. #INPUT

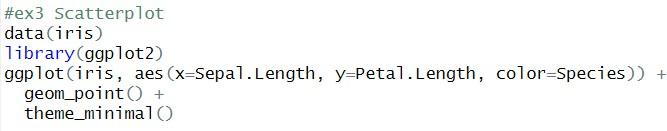
#OUTPUT



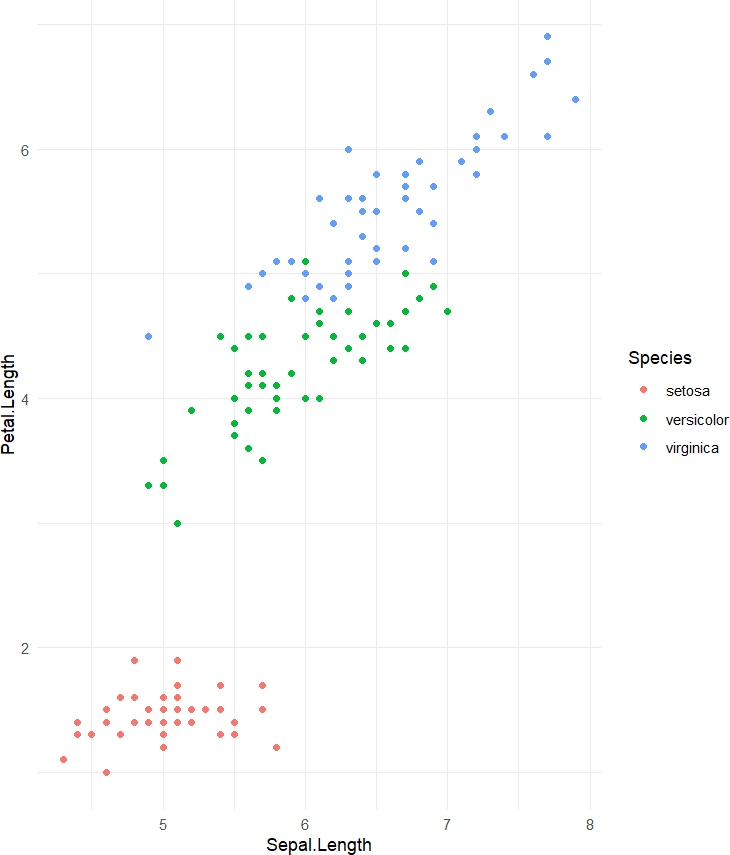
1. Scatterplot to Explore Relationships

Dataset: Built-in iris dataset (Flower Measurements)

The iris dataset contains sepal and petal lengths and widths of three flower species.

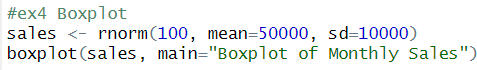
* + Create a scatterplot to explore relationships between variables.
  + Use colours to distinguish species. #INPUT

#OUTPUT

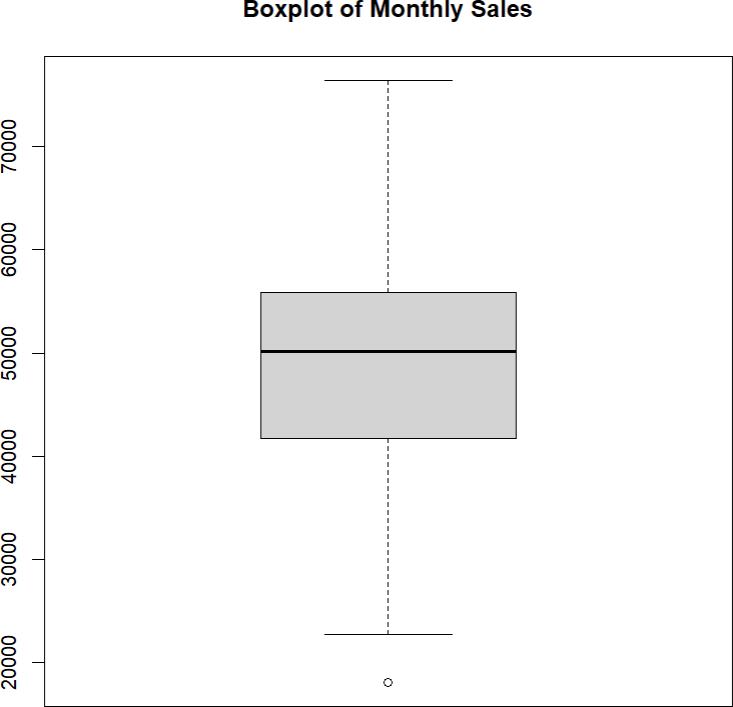


1. Boxplot for Detecting Outliers
   * Dataset: Simulated monthly sales data for a store.Create a boxplot to detect outliers..

#INPUT

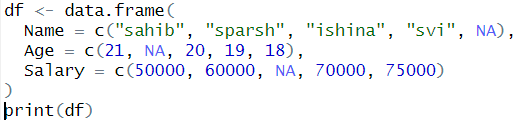


#OUTPUT

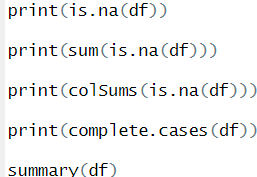


# **Experiment 3**

Objective: Learn data cleaning techniques, including handling missing data, outliers, and data imputation.

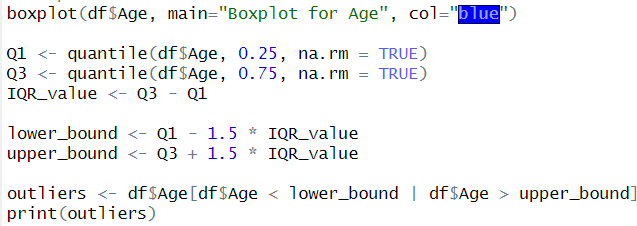


1. Exploring Inbuilt Functions for Data Cleaning
   * Check missing values in a dataset using is.na(), complete.cases(), and summary().

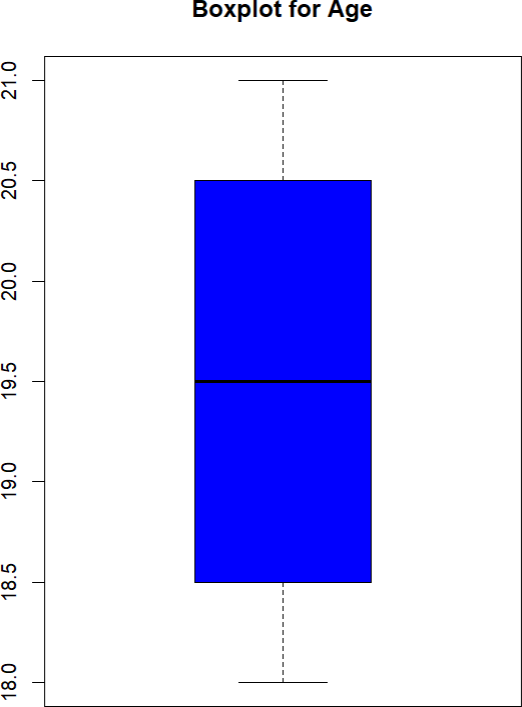
#INPUT

#OUTPUT

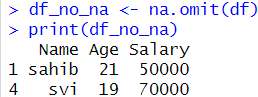


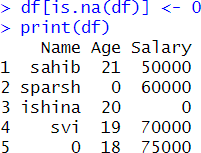
* + Identify outliers using boxplot(), quantile(), and IQR(). #INPUT

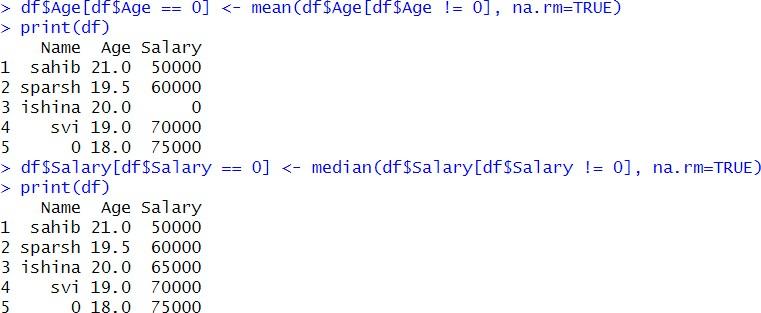
#OUTPUT

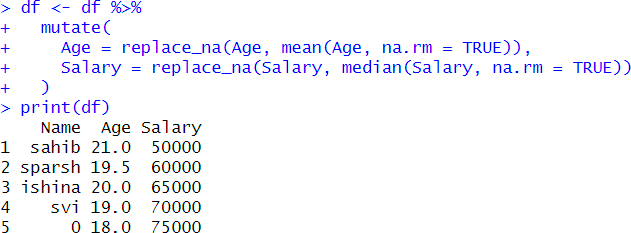


* + Explore imputation methods like mean, median, and mode replacement using na.omit(),impute(), and mice().



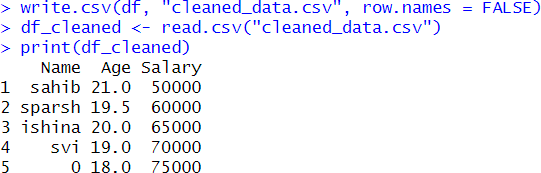




* + Learn about tidyverse functions (mutate(), filter(), replace\_na()).
  + Use summary(df) and str(df) to get dataset insights.



* + Read and write cleaned data using read.csv() and write.csv().



1. Handling Missing Data (NA, NaN, Inf, NULL)

Create a sample dataset (dataframe) with missing values for hands-on practice

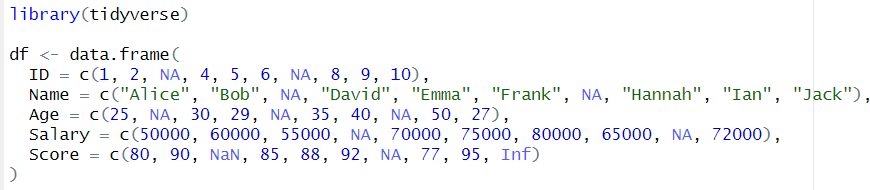
ID Name Age Salary Score

1 Alice 25 50000 80

2 Bob NA 60000 90

3 NA 30 55000 NaN

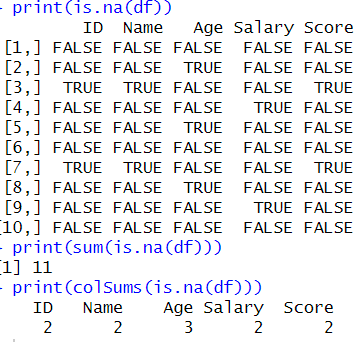
1. David 29 NA 85
2. Emma NA 70000 88
3. Frank 35 75000 92
4. NA 40 80000 NA
5. Hannah NA 65000 77
6. Ian 50 NA 95
7. Jack 27 72000 Inf



* 1. Identify missing data (is.na(df), sum(is.na(df))). #INPUT

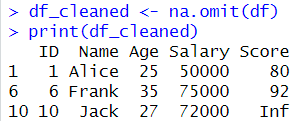


#OUTPUT



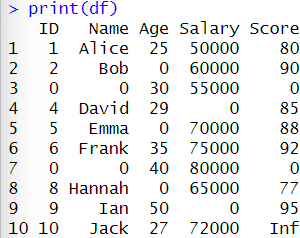
* 1. Remove missing rows (na.omit(df)). #INPUT



#OUTPUT

* 1. Replace NA with zero (df[is.na(df)] <- 0). #INPUT



#OUTPUT

* 1. Replace NA with column mean (df$Age[is.na(df$Age)] <- mean(df$Age, na.rm=TRUE)).

#INPUT

#OUTPUT

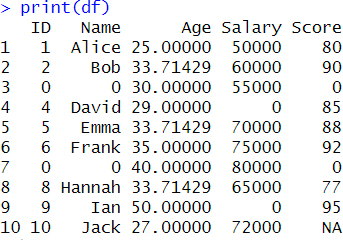


* 1. Remove Inf and NaN (df$Score[is.infinite(df$Score) | is.nan(df$Score)] <- NA).

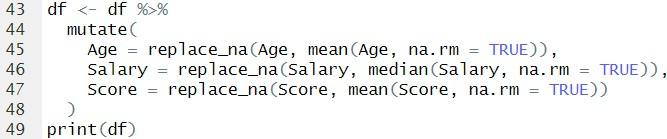
#INPUT

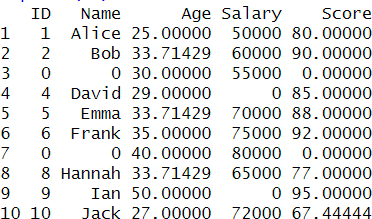


#OUTPUT



* 1. Use tidyverse’s replace\_na() for selective column handling. #INPUT

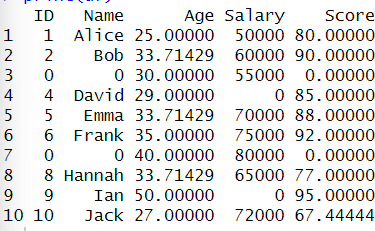


#OUTPUT

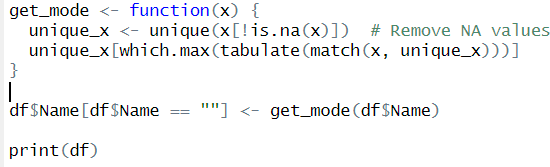
* 1. Drop columns with excessive missing data (df <- df[, colSums(is.na(df)) < nrow(df) \*0.5]).

#INPUT

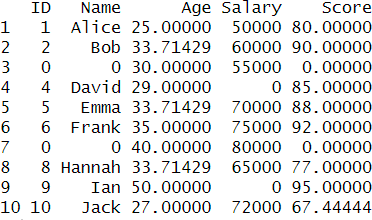


#OUTPUT

* 1. Fill missing categorical values with the mode. #INPUT



#OUTPUT

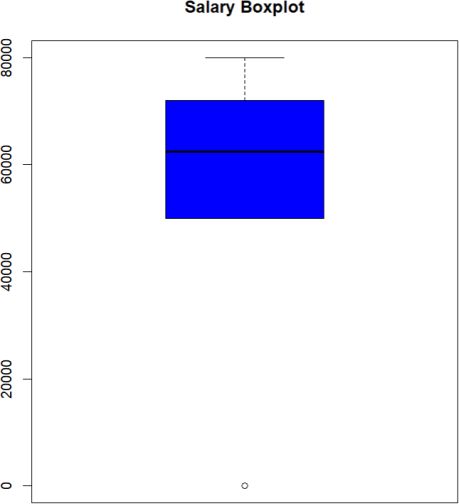


1. Outlier Detection & Handling

Detect and remove outliers from the dataset after handling missing data.

Tasks:

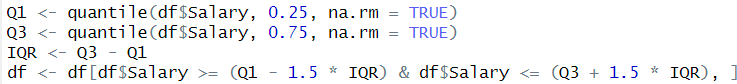
1. Boxplot Visualization to visualize salary data



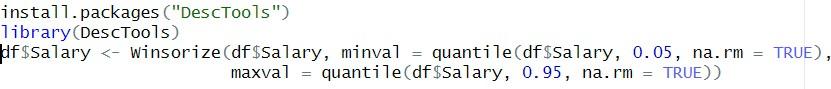
1. Z-Score Method (values outside ±3 standard deviations).



1. IQR Method: Remove values outside Q1 - 1.5\*IQR and Q3 + 1.5\*IQR.



1. Winsorization: Replace extreme values with percentiles (Winsorize()).



1. Detect & Remove Outliers Using tidyverse (filter()).



1. Detect Outliers in Multiple Columns (apply()).



1. Create a Clean Dataset After Removing Outliers.



1. Data Imputation

Explore data imputation techniques to fill missing values. Tasks:

1. Convert NaN and Inf values to NA before applying imputation.



1. Remove rows with missing values using na.omit(df).



1. Drop columns where more than 50% of data is missing.



1. Replace all NA values with 0 for numerical columns.
2. Replace missing values in Age with the mean



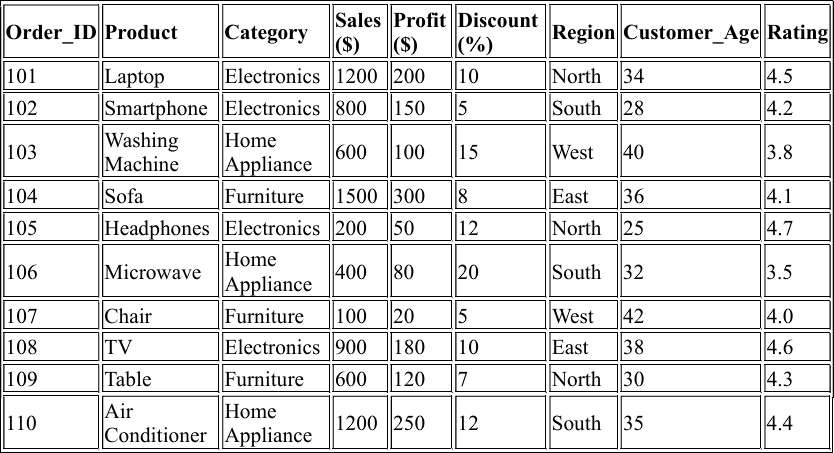
1. Replace missing values in Salary with the median.
2. Replace missing Name values with the most frequent name (Mode)



**Experiment - 4**

Analyze a dataset and decide which type of visualization is most appropriate for different insights.

Dataset: Sales Performance Data Below is a sample dataset containing information about product sales in different regions.



* 1. **Which product category generates the highest revenue?**

A bar plot or pie chart would be suitable to compare the total revenue generated by each category.

Bar plots are preferred for categorical comparisons. And, Pie charts give a visual sense of proportion.



* 1. **What is the relationship between discount percentage and profit?**

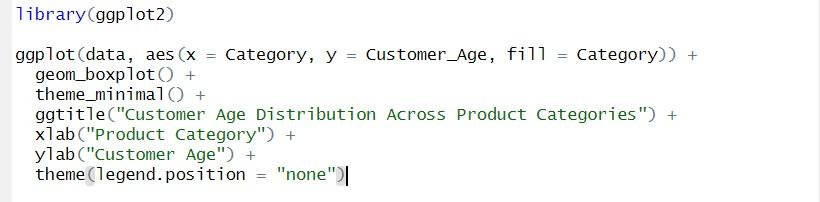
- **Choose an appropriate visualization to assess the impact of discounts on profitability.**

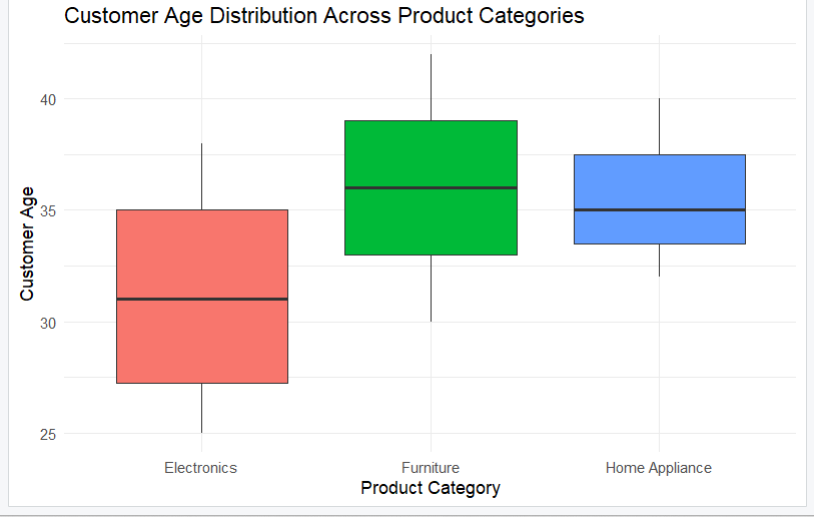
A Scatter Plot will be used here.



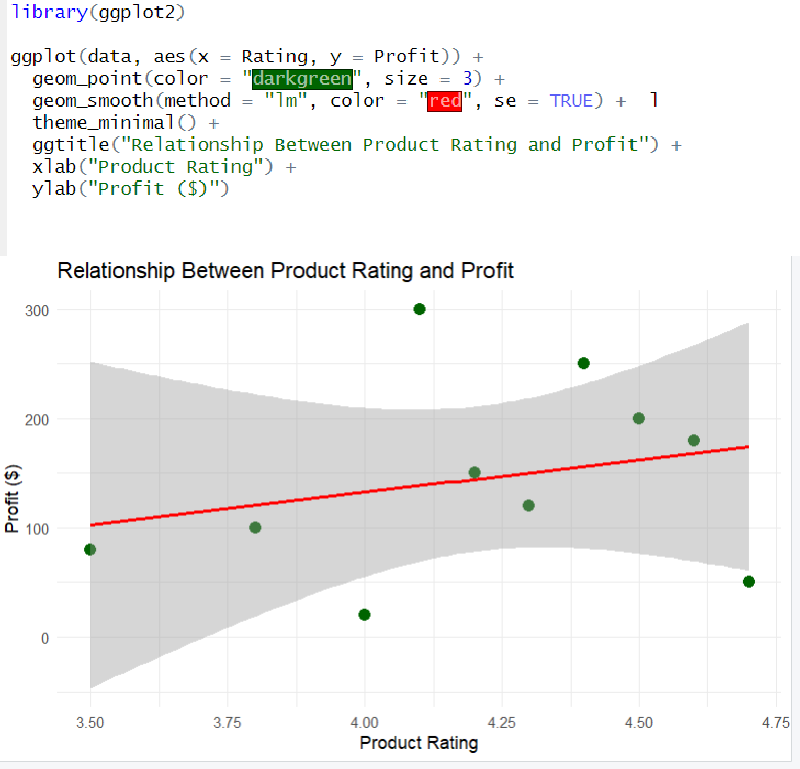
* 1. **How does customer age distribution vary across different product categories?**
     + **Select a plot that effectively compares age groups across different categories.**

A Box Plot can be used to compare Customer Age distribution across different Product Categories.





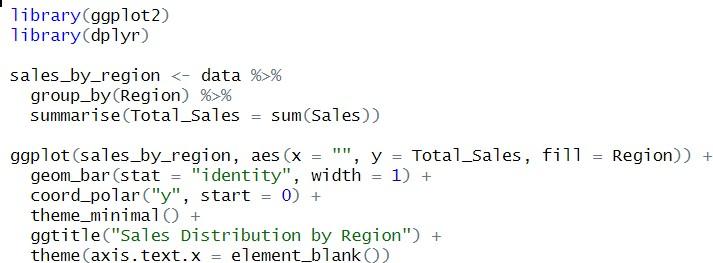
* 1. **Is there a relationship between product rating and profit?**
     + **Determine the best way to visualize this trend.**

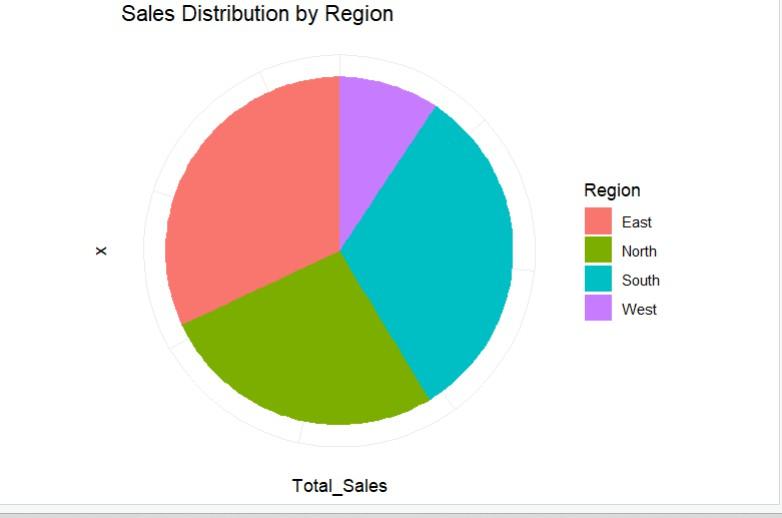
To analyze the relationship between Product Rating and Profit**,** a Scatter Plot with a trend line can be used.

**5. Which region contributes the most to total sales, and how are sales distributed among regions?**

**- Choose a visualization to highlight regional sales contributions.**

To compare sales contributions by region, a pie chart can be used.





**6. Identify outliers in the profit column. What method would you use to detect them?**

Outliers can be identified using the **Interquartile Range (IQR) Method**. This approach detects data points that are **below Q1 - 1.5**

* **IQR or above Q3 + 1.5 \* IQR**.

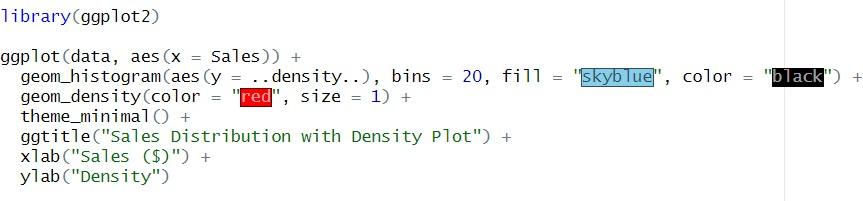
**IQR Calculation:**

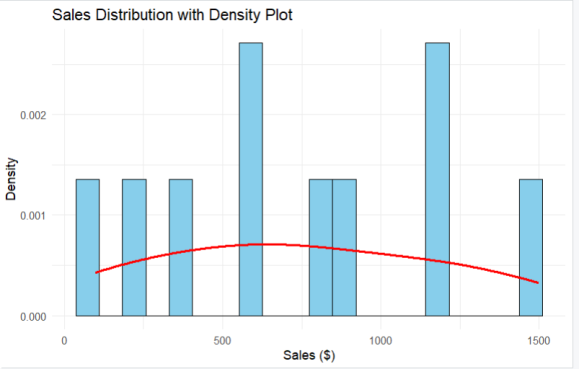
* + Q1: 25th percentile of Profit.
  + Q3: 75th percentile of Profit.
  + IQR: Difference between Q3 and Q1.

- Box plot can be used here for visualization.



**7. Analyze the distribution of sales and decide whether the data follows a normal distribution.**





**8. Compare sales trends across different regions. Suggest an effective visualization.**

**Effective visualization- A Grouped Bar chart.**



**9. Visualize the relationship between customer age and product rating.**



**Experiment 5**

**Exercise 5.1**

# %%

# Advertising Budget (in $1000) Monthly Sales (in $1000)

data <- data.frame(

AdvertisingBudget = c(1,2,3,4,5),

MonthlySales = c(4,5,7,8,11)

)

# Takign AdvertisingBudget as Independent cause vibes also cause more budget = more sales

# %% md

#Perform a linear regression to predict sales based on the advertising budget using R

# %%

n = length(data$AdvertisingBudget)

y\_sum = sum(data$MonthlySales)

x\_sum = sum(data$AdvertisingBudget)

print(n \* sum(data$AdvertisingBudget \* data$AdvertisingBudget))

**Output- [1] 275**

# %%

slope = (n \* sum(data$AdvertisingBudget \* data$MonthlySales) - x\_sum \* y\_sum) / (n \* sum(data$AdvertisingBudget \* data$AdvertisingBudget) - x\_sum \* x\_sum)

intercept = (y\_sum - slope \* x\_sum) / n

# %%

print(intercept)

**Output –[1] 1.9**

print(slope)

**Output - [1] 1.7**

# %%

model <- lm(MonthlySales ~ AdvertisingBudget, data = data)

print(model)

**Call:**

**lm(formula = MonthlySales ~ AdvertisingBudget, data = data)**

**Coefficients:**

**(Intercept) AdvertisingBudget**

**1.9 1.7**

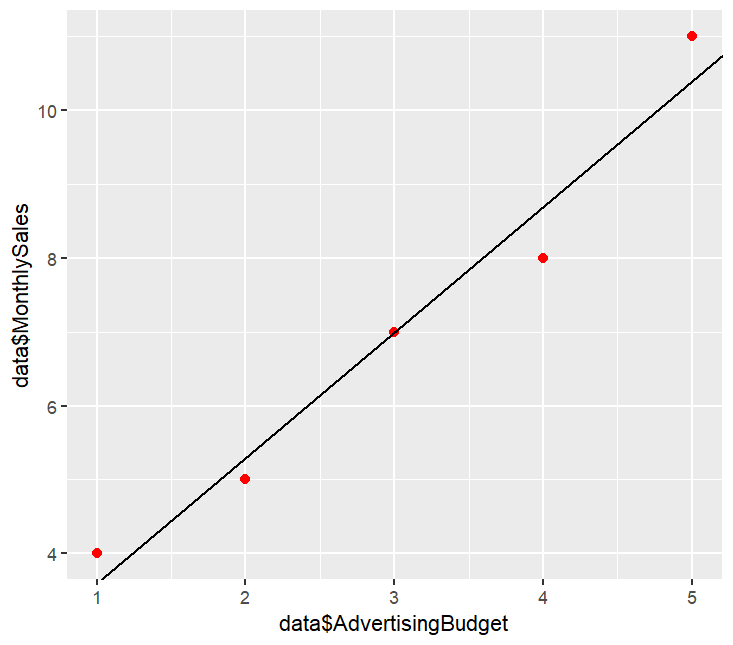
# %%

library(ggplot2)

ggplot(data, aes(x = data$AdvertisingBudget, y = data$MonthlySales)) +

geom\_point(color="red", size=2) +

geom\_abline(mapping=NULL,slope=slope, intercept=intercept)



Exercise 5.2

# %%

data <- read.csv("/home/asus/content/Notes/Semester 4/FDN Lab/Experiments/Experiment 5/house\_prices.csv")

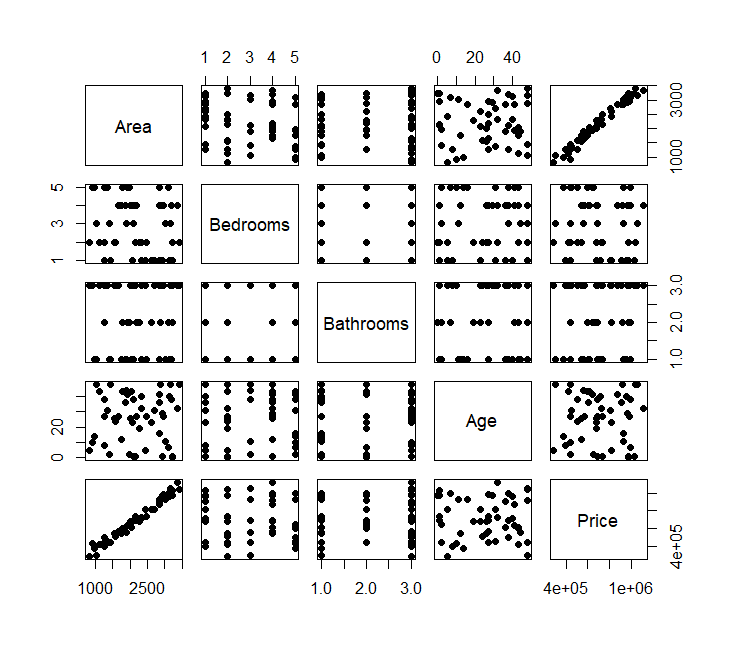
# %%

summary(data)

| **Area Bedrooms Bathrooms Age Price**  **Min. : 821 Min. :1.00 Min. :1.0 Min. : 0.00 Min. : 285302**  **1st Qu.:1592 1st Qu.:2.00 1st Qu.:1.0 1st Qu.:11.25 1st Qu.: 541287**  **Median :2130 Median :3.00 Median :2.5 Median :27.00 Median : 690313**  **Mean :2196 Mean :2.96 Mean :2.2 Mean :25.76 Mean : 708433**  **3rd Qu.:2866 3rd Qu.:4.00 3rd Qu.:3.0 3rd Qu.:39.50 3rd Qu.: 921673**  **Max. :3412 Max. :5.00 Max. :3.0 Max. :48.00 Max. :1115614** |
| --- |
|  |
| |  | | --- | |

# %%

pairs(data, pch = 19) #scatterplot matrices



# %% md

#Fit a linear regression model using lm() where Price is the dependent variable.

# %%

lm(Age ~ Price, data=data)

models <- list()

for (col in names(data)[names(data) != "Price"]) {

formula <- as.formula(paste("Price ~", col))

models[[col]] <- lm(formula = formula, data = data)

}

# %%

print(names(models[["Area"]]))

**[1] "coefficients" "residuals" "effects" "rank" "fitted.values"**

**[6] "assign" "qr" "df.residual" "xlevels" "call"**

**[11] "terms" "model"**

# %%

library(ggplot2)

# %%

for(col in names(data)[names(data) != "Price"]){

p <- ggplot(data, aes\_string(y = "Price", x = col)) +

geom\_point(color="red", size=2) +

geom\_abline(color="blue", intercept=coef(models[[col]])[1], slope=coef(models[[col]])[2]) +

labs(title = paste("Price vs", col), x = col, y = "Price" )

ggsave(paste("/home/asus/content/Notes/Semester 4/FDN Lab/Experiments/Experiment 5/", col,".jpeg"))

}

# %%

library(dplyr)

# %%

# Create a summary dataframe

model\_performance <- data.frame(

Predictor = names(models),

RMSE = sapply(models, function(m) sqrt(mean((data$Price - predict(m))^2))),

R\_squared = sapply(models, function(m) summary(m)$r.squared)

)

# %%

# Sort by best R-squared (descending)

model\_performance

arrange(desc(R\_squared))

print()

**model\_performance**

**Predictor RMSE R\_squared**

**Area Area 37173.09 9.719968e-01**

**Bedrooms Bedrooms 221310.14 7.450071e-03**

**Bathrooms Bathrooms 220209.36 1.729922e-02**

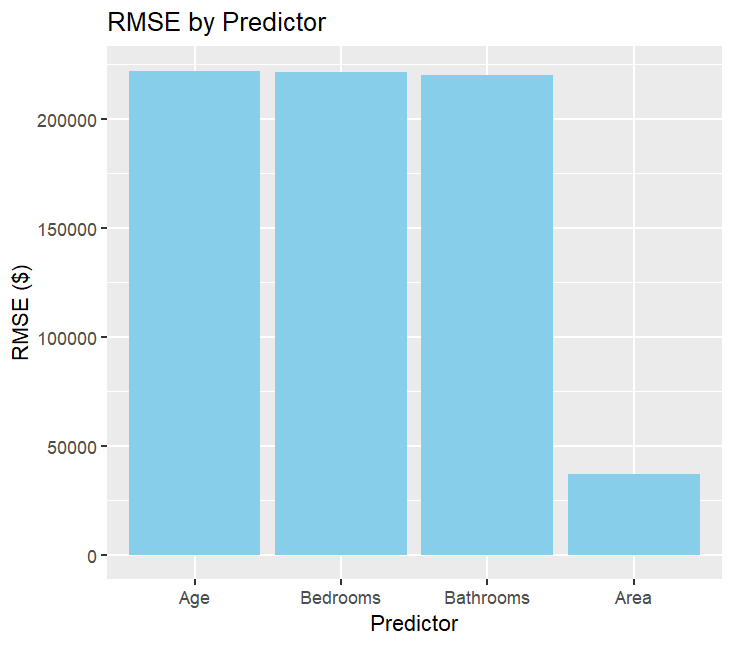
**Age Age 222136.31 2.569724e-05**

# %%

ggplot(model\_performance, aes(x = reorder(Predictor, -RMSE), y = RMSE)) +

geom\_bar(stat = "identity", fill = "skyblue") +

labs(title = "RMSE by Predictor", x = "Predictor", y = "RMSE ($)")



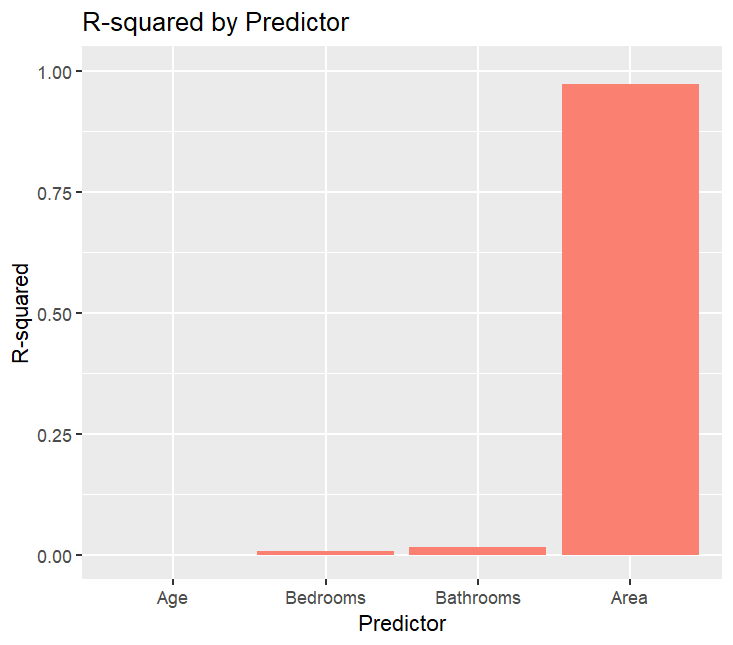
# %%

ggplot(model\_performance, aes(x = reorder(Predictor, R\_squared), y = R\_squared)) +

geom\_bar(stat = "identity", fill = "salmon") +

labs(title = "R-squared by Predictor", x = "Predictor", y = "R-squared") +

ylim(0, 1)



**Experiment 6**

**Exercise 6.1**

library(ggplot2)

library(dplyr)

library(cluster)

# %%

set.seed(1)

mall\_data <- read.csv("C:/Users/naman/OneDrive/Desktop/Data science lab/New folder/Experiments/Experiment 6/Mall\_Customers.csv")

# %%

head(mall\_data)

summary(mall\_data)

**head(mall\_data)**

**CustomerID Gender Age Annual.Income..k.. Spending.Score..1.100.**

**1 1 Male 19 15 39**

**2 2 Male 21 15 81**

**3 3 Female 20 16 6**

**4 4 Female 23 16 77**

**5 5 Female 31 17 40**

**6 6 Female 22 17 76**

**> summary(mall\_data)**

**CustomerID Gender Age Annual.Income..k..**

**Min. : 1.00 Length:200 Min. :18.00 Min. : 15.00**

**1st Qu.: 50.75 Class :character 1st Qu.:28.75 1st Qu.: 41.50**

**Median :100.50 Mode :character Median :36.00 Median : 61.50**

**Mean :100.50 Mean :38.85 Mean : 60.56**

**3rd Qu.:150.25 3rd Qu.:49.00 3rd Qu.: 78.00**

**Max. :200.00 Max. :70.00 Max. :137.00**

**Spending.Score..1.100.**

**Min. : 1.00**

**1st Qu.:34.75**

**Median :50.00**

**Mean :50.20**

**3rd Qu.:73.00**

**Max. :99.00**

# %%

colnames(mall\_data) <- c("CustomerID", "Gender", "Age", "AnnualIncome", "SpendingScore")

# %%

ggplot(mall\_data, aes(x = AnnualIncome, y = SpendingScore)) +

geom\_point(aes(color = Gender), alpha = 0.7) +

labs(title = "Annual Income vs Spending Score by Gender",

x = "Annual Income (k$)",

y = "Spending Score (1-100)") +

theme\_minimal()



# %%

ggplot(mall\_data, aes(x = Age)) +

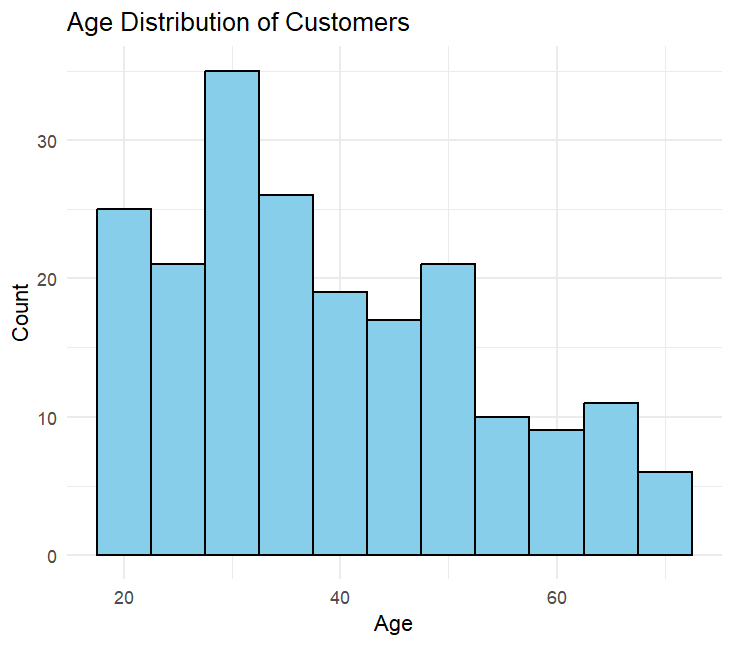
geom\_histogram(binwidth = 5, fill = "skyblue", color = "black") +

labs(title = "Age Distribution of Customers",

x = "Age",

y = "Count") +

theme\_minimal()



# %%

ggplot(mall\_data, aes(x = Gender, y = SpendingScore, fill = Gender)) +

geom\_boxplot() +

labs(title = "Spending Score Distribution by Gender",

y = "Spending Score (1-100)") +

theme\_minimal()

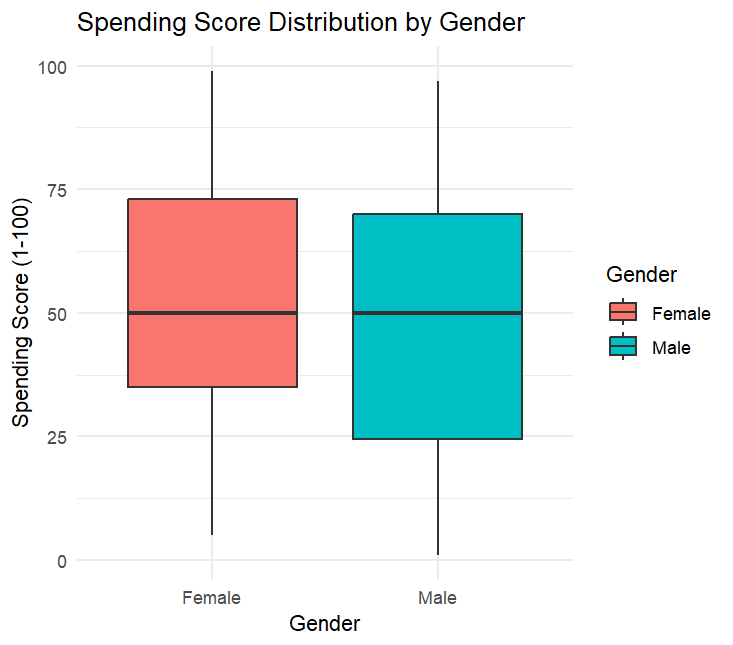
# ===================================

# Part 2

# ===================================

# %%

data\_for\_clustering <- mall\_data[, c("AnnualIncome", "SpendingScore")]



# %%

wss <- function(k) {

kmeans(data\_for\_clustering, k, nstart = 10)$tot.withinss

}

# %%

k\_values <- 1:10

wss\_values <- sapply(k\_values, wss)

# %%

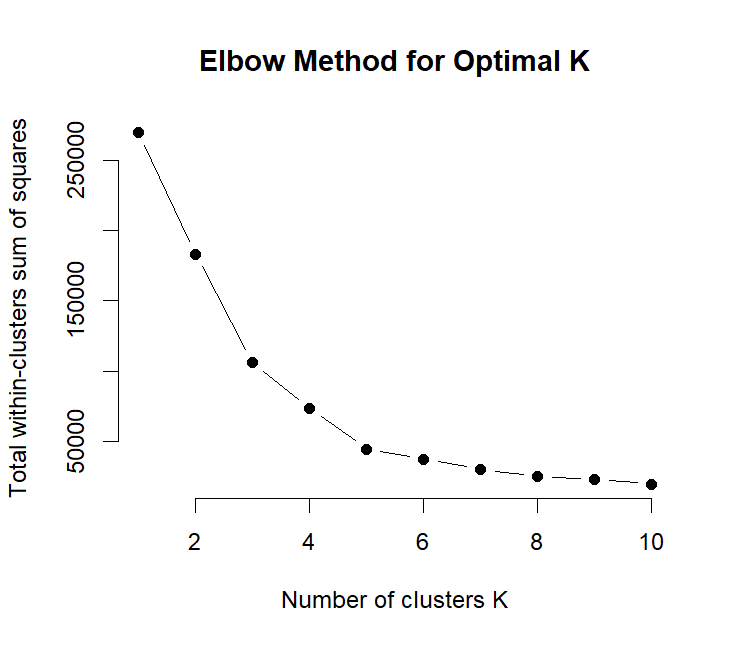
plot(k\_values, wss\_values,

type = "b", pch = 19, frame = FALSE,

xlab = "Number of clusters K",

ylab = "Total within-clusters sum of squares",

main = "Elbow Method for Optimal K")



# %%

avg\_sil <- function(k) {

km.res <- kmeans(data\_for\_clustering, centers = k, nstart = 25)

ss <- silhouette(km.res$cluster, dist(data\_for\_clustering))

mean(ss[, 3])

}

# %%

k\_values <- 2:10

avg\_sil\_values <- sapply(k\_values, avg\_sil)

# %%

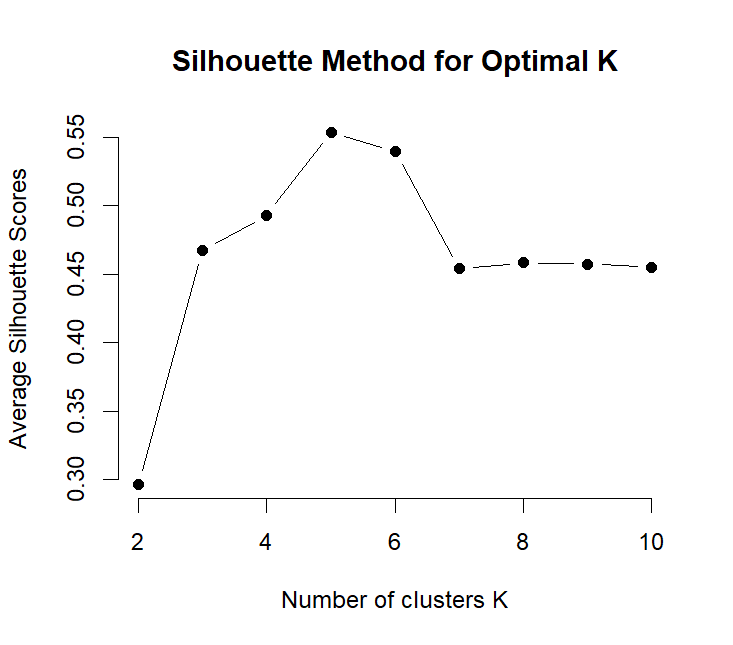
plot(k\_values, avg\_sil\_values,

type = "b", pch = 19, frame = FALSE,

xlab = "Number of clusters K",

ylab = "Average Silhouette Scores",

main = "Silhouette Method for Optimal K")



# %%

final\_k <- 5

# %%

kmeans\_result <- kmeans(data\_for\_clustering, centers = final\_k, nstart = 25)

mall\_data$Cluster <- as.factor(kmeans\_result$cluster)

# %%

ggplot(mall\_data, aes(x = AnnualIncome, y = SpendingScore, color = Cluster)) +

geom\_point(size = 3, alpha = 0.7) +

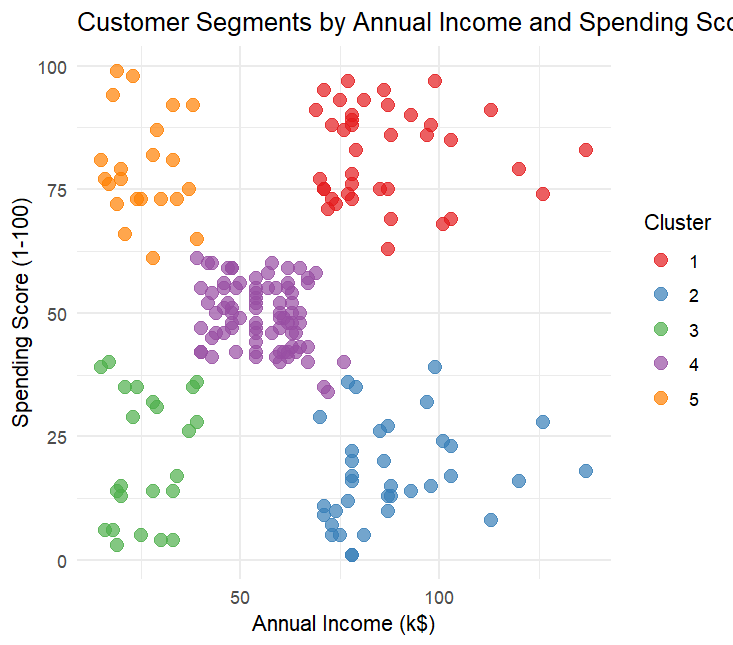
scale\_color\_brewer(palette = "Set1") +

labs(title = "Customer Segments by Annual Income and Spending Score",

x = "Annual Income (k$)",

y = "Spending Score (1-100)") +

theme\_minimal()



# %%

cluster\_stats <- mall\_data %>%

group\_by(Cluster) %>%

summarise(

Count = n(),

Avg\_Age = mean(Age),

Avg\_Income = mean(AnnualIncome),

Avg\_Spending = mean(SpendingScore),

Female\_Pct = sum(Gender == "Female") / n() \* 100

)

# %%

# Printing Stats

print(cluster\_stats)

| **print(cluster\_stats)**  **# A tibble: 5 × 6**  **Cluster Count Avg\_Age Avg\_Income Avg\_Spending Female\_Pct**  ***<fct>* *<int>* *<dbl>* *<dbl>* *<dbl>* *<dbl>***  **1 1 39 32.7 86.5 82.1 53.8**  **2 2 35 41.1 88.2 17.1 45.7**  **3 3 23 45.2 26.3 20.9 60.9**  **4 4 81 42.7 55.3 49.5 59.3**  **5 5 22 25.3 25.7 79.4 59.1** |
| --- |
|  |
| | **>** | | --- | |

# %%

ggplot(cluster\_stats, aes(x = Avg\_Income, y = Avg\_Spending, size = Count, color = Cluster)) +

geom\_point() +

scale\_size(range = c(5, 15)) +

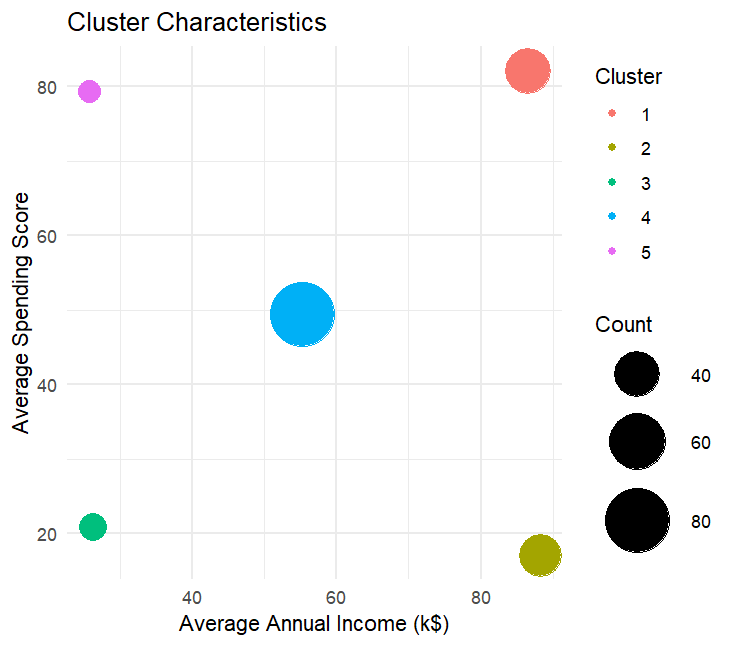
labs(title = "Cluster Characteristics",

x = "Average Annual Income (k$)",

y = "Average Spending Score") +

theme\_minimal()

# %%



**Experiment 6.2**

library(tidyverse)

library(cluster)

# %%

set.seed(1)

# %%

happiness <- read\_csv("C:/Users/naman/OneDrive/Desktop/Data science lab/New folder/Experiments/Experiment 6/archive(11)/2019.csv")

# %%

head(happiness)

summary(happiness)

**head(happiness)**

**# A tibble: 6 × 9**

**`Overall rank` Country Score `GDP per capita` `Social support` Healthy life expecta…¹**

***<dbl>* *<chr>* *<dbl>* *<dbl>* *<dbl>* *<dbl>***

**1 1 Finland 7.77 1.34 1.59 0.986**

**2 2 Denmark 7.6 1.38 1.57 0.996**

**3 3 Norway 7.55 1.49 1.58 1.03**

**4 4 Iceland 7.49 1.38 1.62 1.03**

**5 5 Netherla… 7.49 1.40 1.52 0.999**

**6 6 Switzerl… 7.48 1.45 1.53 1.05**

**# ℹ abbreviated name: ¹​`Healthy life expectancy`**

**# ℹ 3 more variables: `Freedom to make life choices` <dbl>, Generosity <dbl>,**

**# `Perceptions of corruption` <dbl>**

**> summary(happiness)**

**Overall rank Country Score GDP per capita Social support**

**Min. : 1.00 Length:156 Min. :2.853 Min. :0.0000 Min. :0.000**

**1st Qu.: 39.75 Class :character 1st Qu.:4.545 1st Qu.:0.6028 1st Qu.:1.056**

**Median : 78.50 Mode :character Median :5.380 Median :0.9600 Median :1.272**

**Mean : 78.50 Mean :5.407 Mean :0.9051 Mean :1.209**

**3rd Qu.:117.25 3rd Qu.:6.184 3rd Qu.:1.2325 3rd Qu.:1.452**

**Max. :156.00 Max. :7.769 Max. :1.6840 Max. :1.624**

**Healthy life expectancy Freedom to make life choices Generosity**

**Min. :0.0000 Min. :0.0000 Min. :0.0000**

**1st Qu.:0.5477 1st Qu.:0.3080 1st Qu.:0.1087**

**Median :0.7890 Median :0.4170 Median :0.1775**

**Mean :0.7252 Mean :0.3926 Mean :0.1848**

**3rd Qu.:0.8818 3rd Qu.:0.5072 3rd Qu.:0.2482**

**Max. :1.1410 Max. :0.6310 Max. :0.5660**

**Perceptions of corruption**

**Min. :0.0000**

**1st Qu.:0.0470**

**Median :0.0855**

**Mean :0.1106**

**3rd Qu.:0.1412**

**Max. :0.4530**

# %%

# Filtering out the text cols

features <- c("Overall rank", "Score", "GDP per capita", "Social support", "Healthy life expectancy", "Freedom to make life choices", "Generosity", "Perceptions of corruption")

happiness <- happiness %>%

select(all\_of(features)) %>%

na.omit()

# %%

features <- c("Overall rank", "Country", "Score", "GDP per capita", "Social support", "Healthy life expectancy", "Freedom to make life choices", "Generosity", "Perceptions of corruption")

happiness\_country <- happiness %>%

select(all\_of(features))

# %%

#SCAAAAAAAAAAAAAAAALIng

happiness\_scaled <- scale(happiness)

# %%

# total within-cluster sum of squares

wss <- function(k) {

kmeans(happiness\_scaled, k, nstart = 10)$tot.withinss

}

# %%

k\_values <- 1:10

wss\_values <- map\_dbl(k\_values, wss)

# %%

library(ggplot2)

ggplot(data.frame(k = k\_values, wss = wss\_values), aes(k, wss)) +

geom\_line() + geom\_point() +

scale\_x\_continuous(breaks = k\_values) +

labs(title = "Elbow Method for Optimal Number of Clusters",

x = "Number of clusters K",

y = "Total within-cluster sum of squares") +

theme\_minimal()



# %%

avg\_sil <- function(k) {

km.res <- kmeans(happiness\_scaled, centers = k, nstart = 25)

ss <- silhouette(km.res$cluster, dist(happiness\_scaled))

mean(ss[, 3])

}

# %%

k\_values <- 2:10

avg\_sil\_values <- map\_dbl(k\_values, avg\_sil)

# %%

ggplot(data.frame(k = k\_values, silhouette = avg\_sil\_values), aes(k, silhouette)) +

geom\_line() + geom\_point() +

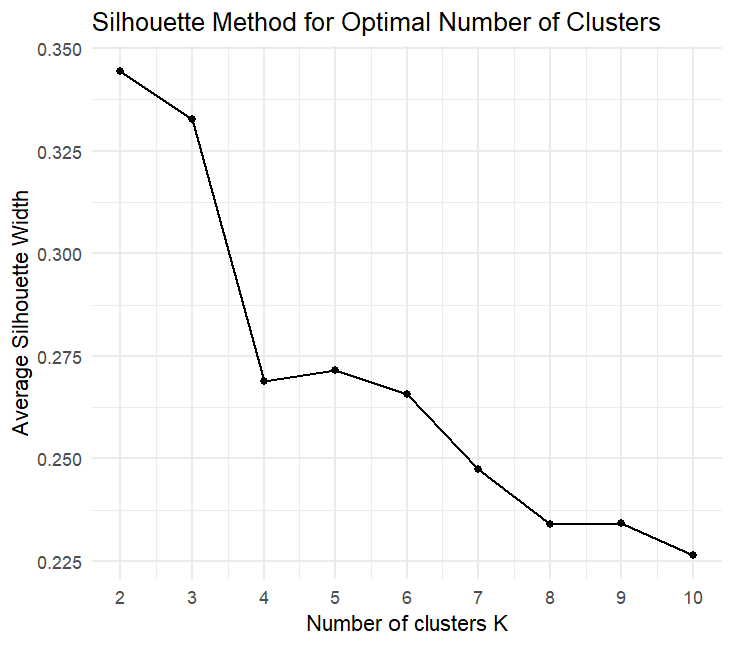
scale\_x\_continuous(breaks = k\_values) +

labs(title = "Silhouette Method for Optimal Number of Clusters",

x = "Number of clusters K",

y = "Average Silhouette Width") +

theme\_minimal()



# %%

optimal\_k <- 5

# %%

kmeans\_result <- kmeans(happiness\_scaled, centers = optimal\_k, nstart = 25)

# %%

# Add cluster assignments to original data

happiness$Cluster <- as.factor(kmeans\_result$cluster)

# %%

# Perform PCA for visualization

pca\_result <- prcomp(happiness\_scaled, scale. = TRUE)

pca\_df <- as.data.frame(pca\_result$x[, 1:2])

pca\_df$Cluster <- happiness$Cluster

# %%

# Plot clusters in PCA space

ggplot(pca\_df, aes(x = PC1, y = PC2, color = Cluster)) +

geom\_point(size = 3, alpha = 0.7) +

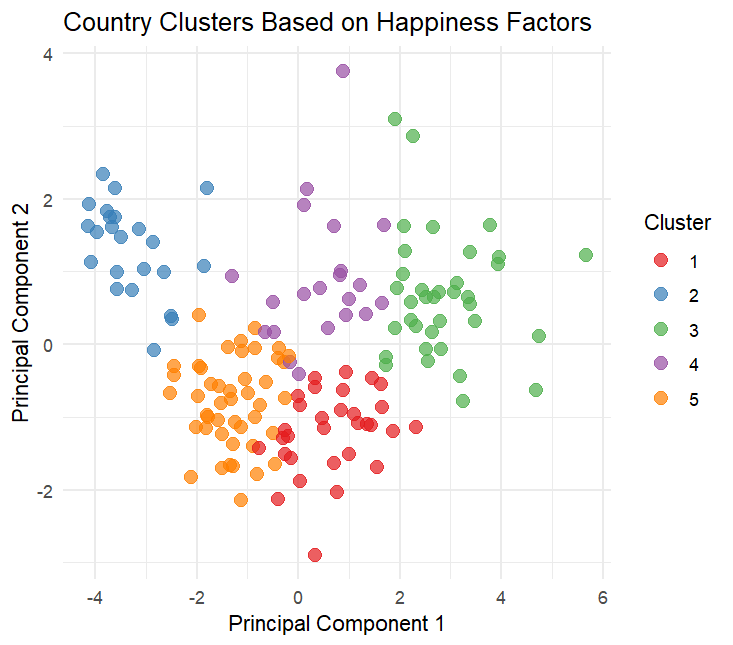
scale\_color\_brewer(palette = "Set1") +

labs(title = "Country Clusters Based on Happiness Factors",

x = "Principal Component 1",

y = "Principal Component 2") +

theme\_minimal()



# %%

# Prepare data for parallel coordinates plot

cluster\_means <- happiness %>%

group\_by(Cluster) %>%

summarise(across(where(is.numeric), mean))

# %%

cluster\_means\_long <- cluster\_means %>%

pivot\_longer(cols = -Cluster, names\_to = "Feature", values\_to = "Mean\_Value")

# %%

ggplot(cluster\_means\_long, aes(x = Feature, y = Mean\_Value, group = Cluster, color = Cluster)) +

geom\_line(size = 1.5) +

geom\_point(size = 3) +

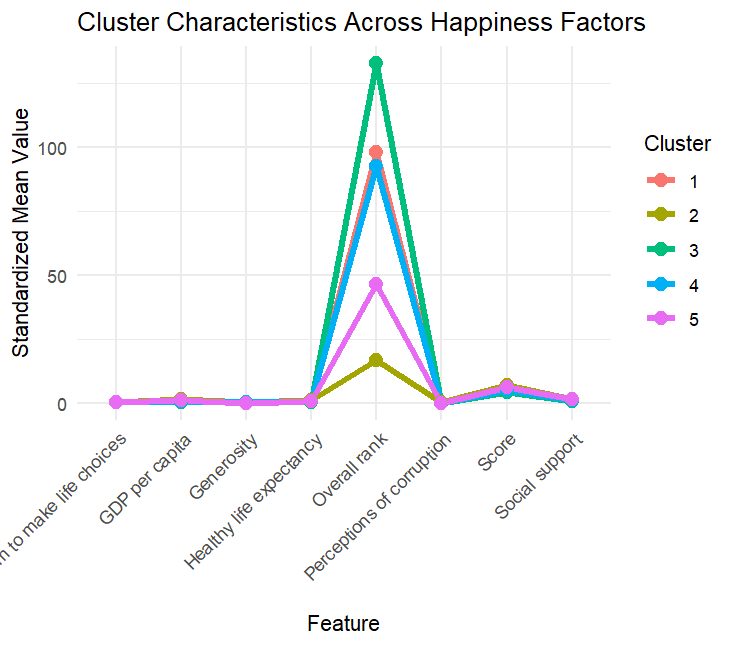
labs(title = "Cluster Characteristics Across Happiness Factors",

y = "Standardized Mean Value") +

theme\_minimal() +

theme(axis.text.x = element\_text(angle = 45, hjust = 1))

# %%



# Calculate and display cluster characteristics

cluster\_profiles <- happiness %>%

group\_by(Cluster) %>%

summarise(

Count = n(),

Avg\_GDP = mean(`GDP per capita`, na.rm = TRUE),

Avg\_Social = mean(`Social support`, na.rm = TRUE),

Avg\_Health = mean(`Healthy life expectancy`, na.rm = TRUE),

Avg\_Freedom = mean(`Freedom to make life choices`, na.rm = TRUE),

Avg\_Generosity = mean(Generosity, na.rm = TRUE),

Avg\_Corruption = mean(`Perceptions of corruption`, na.rm = TRUE)

)

# %%

# Print cluster profiles

print(cluster\_profiles)

| **print(cluster\_profiles)**  **# A tibble: 5 × 8**  **Cluster Count Avg\_GDP Avg\_Social Avg\_Health Avg\_Freedom Avg\_Generosity Avg\_Corruption**  ***<fct>* *<int>* *<dbl>* *<dbl>* *<dbl>* *<dbl>* *<dbl>* *<dbl>***  **1 1 31 0.968 1.20 0.759 0.259 0.106 0.065**  **2 2 23 1.40 1.49 0.990 0.548 0.275 0.281**  **3 3 36 0.351 0.824 0.387 0.301 0.203 0.102**  **4 4 21 0.771 1.17 0.661 0.457 0.290 0.0761**  **5 5 45 1.12 1.39 0.867 0.449 0.130 0.0779** |
| --- |
|  |
| | **>** | | --- | |

# %%

cluster\_profiles\_long <- cluster\_profiles %>%

select(-Count) %>%

pivot\_longer(cols = -Cluster, names\_to = "Feature", values\_to = "Mean\_Value")

# %%

ggplot(cluster\_profiles\_long, aes(x = Feature, y = Mean\_Value, fill = as.factor(Cluster))) +

geom\_col(position = "dodge") +

labs(title = "Average Feature Values by Cluster",

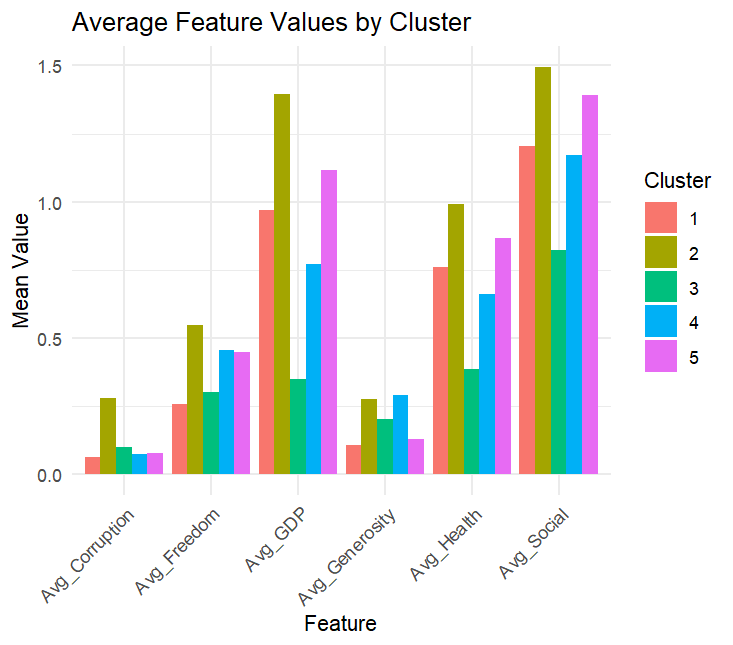
y = "Mean Value",

fill = "Cluster") +

theme\_minimal() +

theme(axis.text.x = element\_text(angle = 45, hjust = 1))

# %%



**Exercise 6.3**

library(tidyverse)

library(cluster)

library(gridExtra)

library(ggplot2)

# %%

set.seed(1)

# %%

happiness <- read\_csv("C:/Users/naman/OneDrive/Desktop/Data science lab/New folder/Experiments/Experiment 6/archive(11)/2019.csv")

# %%

numeric\_data <- happiness %>%

select(

"Overall rank", "Score", "GDP per capita", "Social support", "Healthy life expectancy", "Freedom to make life choices", "Generosity", "Perceptions of corruption"

) %>%

scale()

# %%

rownames(numeric\_data) <- happiness$`Country`

# %%

wcss <- map\_dbl(1:10, ~ kmeans(numeric\_data, ., nstart = 25)$tot.withinss)

# %%

avg\_sil <- map\_dbl(2:10, ~ {

km <- kmeans(numeric\_data, ., nstart = 25)

silhouette\_score <- silhouette(km$cluster, dist(numeric\_data))

mean(silhouette\_score[, 3])

})

# %%

elbow\_plot <- ggplot(data.frame(K = 1:10, WCSS = wcss), aes(K, WCSS)) +

geom\_line(color = "steelblue", size = 1.2) +

geom\_point(color = "red", size = 3) +

labs(title = "Elbow Method (Optimal K)", x = "Number of Clusters (K)", y = "WCSS") +

theme\_minimal()

silhouette\_plot <- ggplot(data.frame(K = 2:10, Silhouette = avg\_sil), aes(K, Silhouette)) +

geom\_line(color = "steelblue", size = 1.2) +

geom\_point(color = "red", size = 3) +

labs(title = "Silhouette Score (Optimal K)", x = "Number of Clusters (K)", y = "Avg. Silhouette Width") +

theme\_minimal()

grid.arrange(elbow\_plot, silhouette\_plot, ncol = 2)

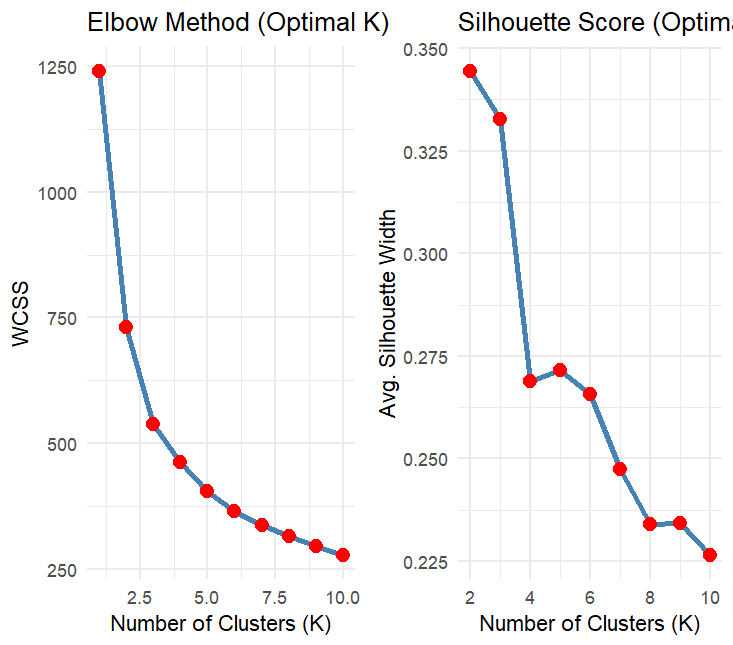
# %%

k2 <- kmeans(numeric\_data, centers = 2, nstart = 25)

k3 <- kmeans(numeric\_data, centers = 3, nstart = 25)

k4 <- kmeans(numeric\_data, centers = 4, nstart = 25)

k5 <- kmeans(numeric\_data, centers = 5, nstart = 25)



# %%

happiness$Cluster\_K2 <- as.factor(k2$cluster)

happiness$Cluster\_K3 <- as.factor(k3$cluster)

happiness$Cluster\_K4 <- as.factor(k4$cluster)

happiness$Cluster\_K5 <- as.factor(k5$cluster)

# %%

plot\_cluster\_means <- function(km\_result, title) {

centers <- as.data.frame(km\_result$centers)

centers$Cluster <- factor(rownames(centers))

centers\_long <- centers %>%

pivot\_longer(cols = -Cluster, names\_to = "Feature", values\_to = "Mean\_Value")

ggplot(centers\_long, aes(x = Feature, y = Mean\_Value, fill = Cluster)) +

geom\_bar(stat = "identity", position = "dodge") +

labs(title = title, y = "Standardized Mean Value", x = "") +

theme\_minimal() +

theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +

scale\_fill\_brewer(palette = "Set1")

}

# %%

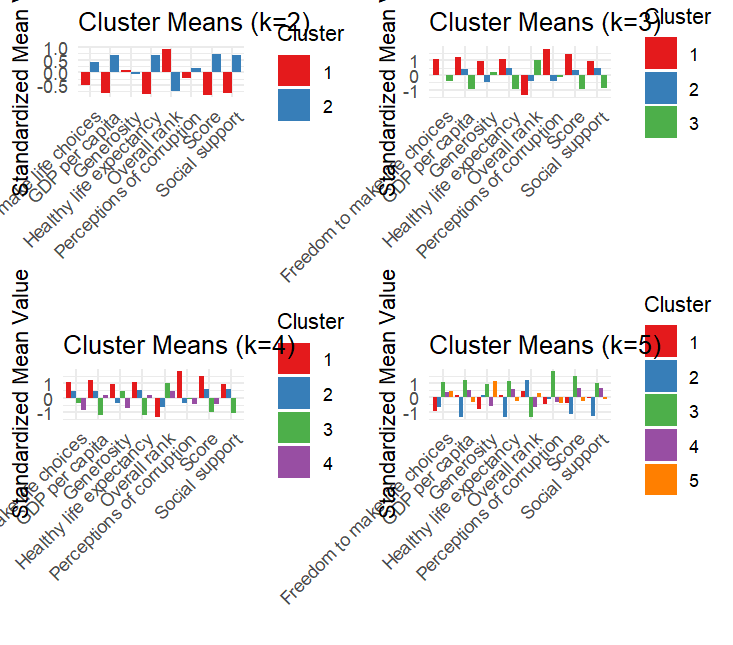
p2 <- plot\_cluster\_means(k2, "Cluster Means (k=2)")

p3 <- plot\_cluster\_means(k3, "Cluster Means (k=3)")

p4 <- plot\_cluster\_means(k4, "Cluster Means (k=4)")

p5 <- plot\_cluster\_means(k5, "Cluster Means (k=5)")

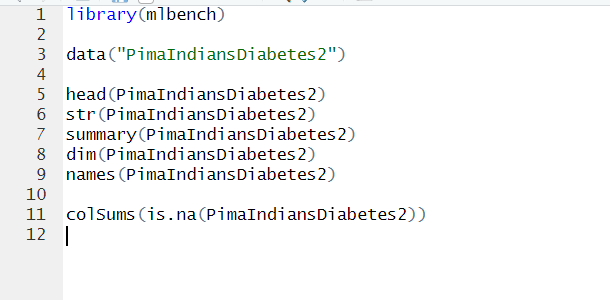
grid.arrange(p2, p3, p4, p5, ncol = 2)

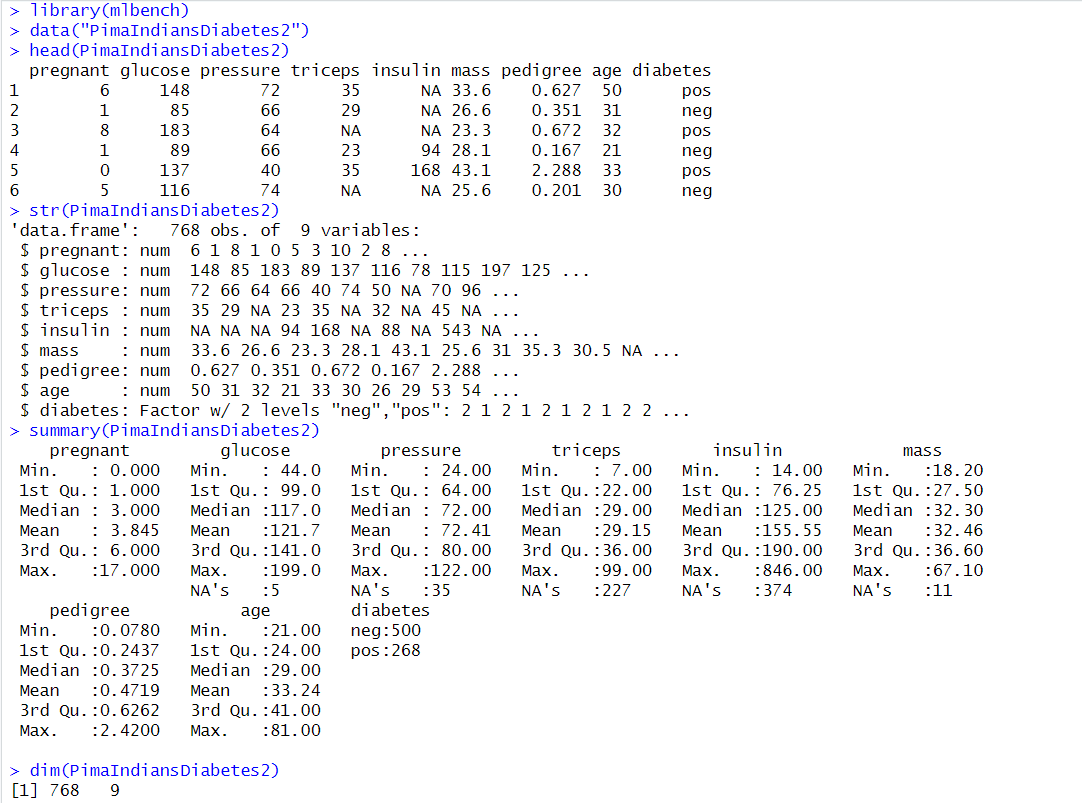


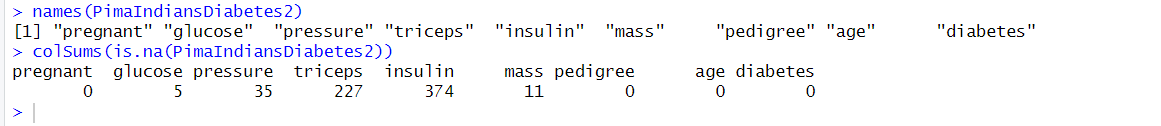
**EXPERIMENT-7**

1. Load and explore the dataset.

(i). data("PimaIndiansDiabetes2") # Alternatively use Heart dataset from UCI

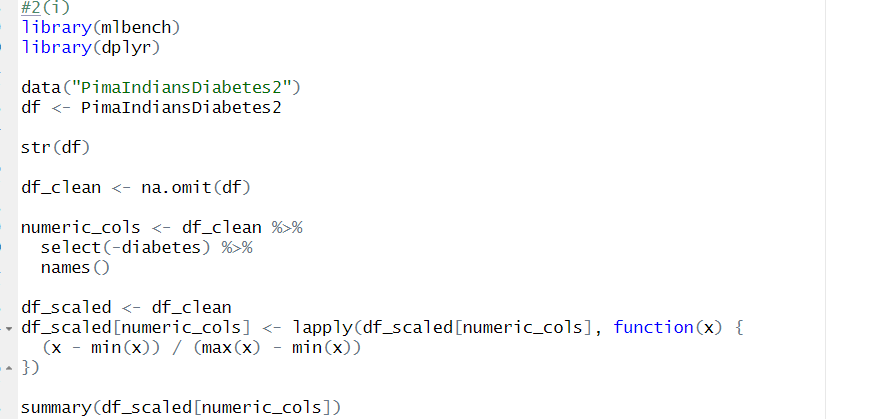


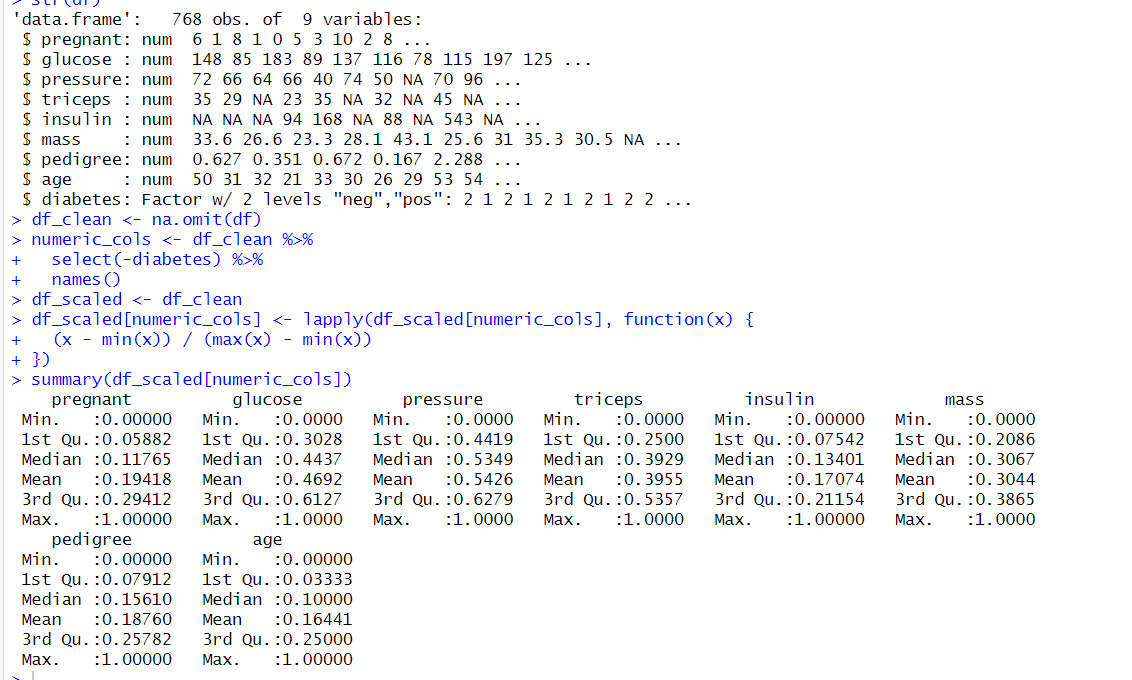




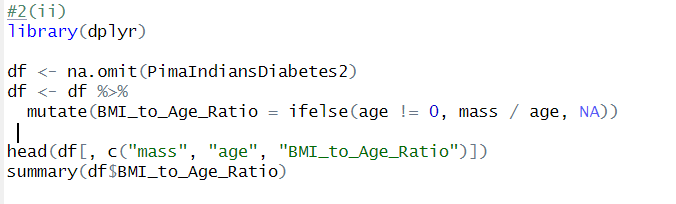
2. Perform feature engineering (categorical encoding, scaling, transformations).

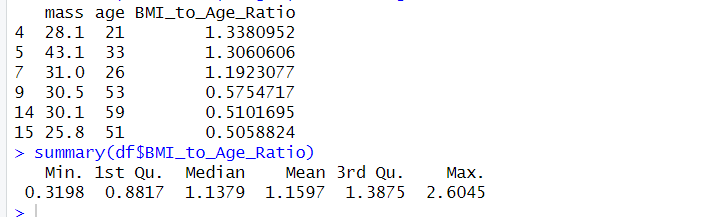
(i). Normalize numerical features



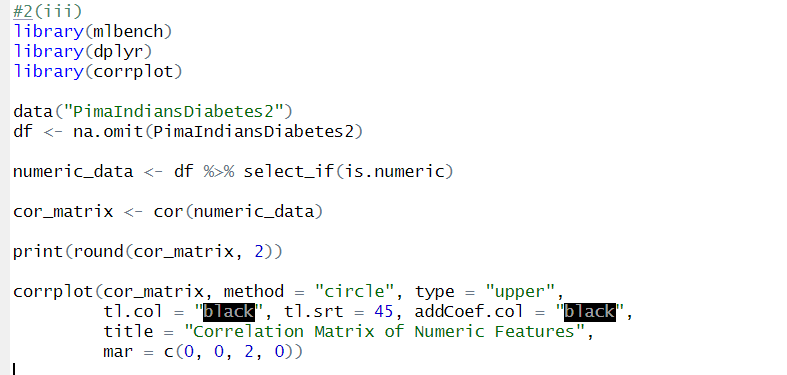


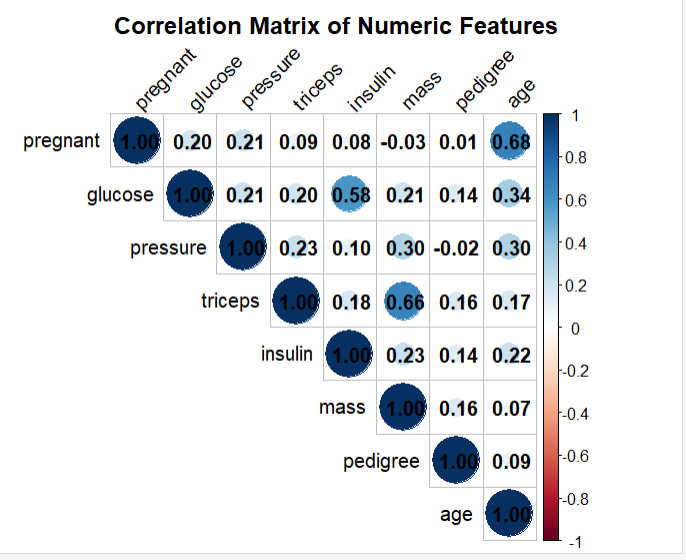
(ii). Create a new feature: BMI-to-Age Ratio



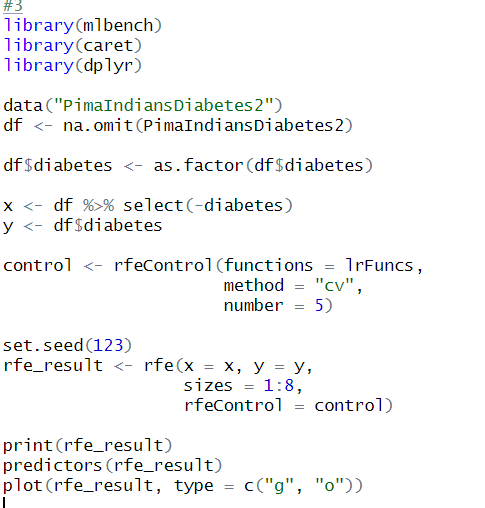


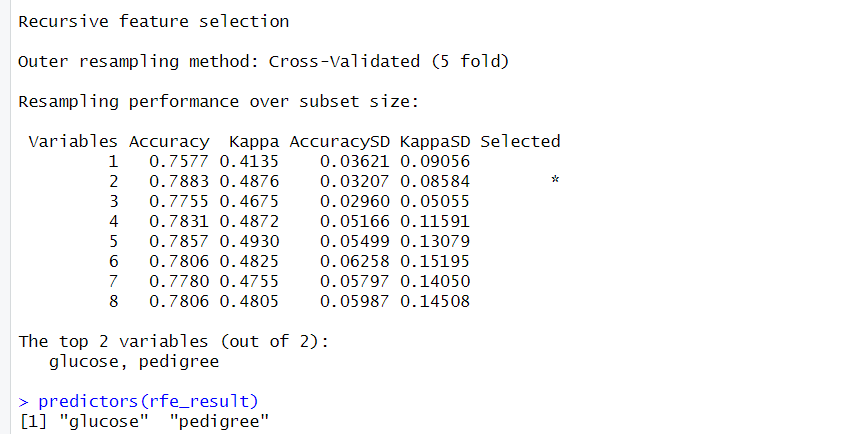
(iii). Correlation matrix

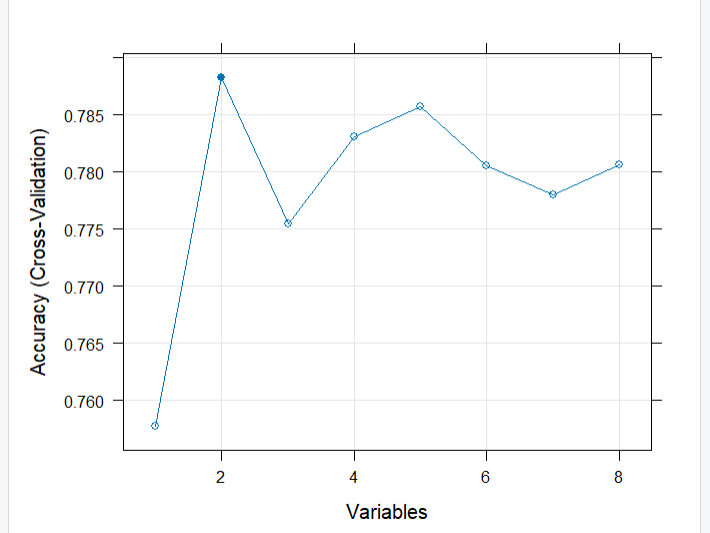




1. Use correlation, recursive feature elimination (RFE), and LASSO for feature selection.
2. Feature Selection - Recursive Feature Elimination

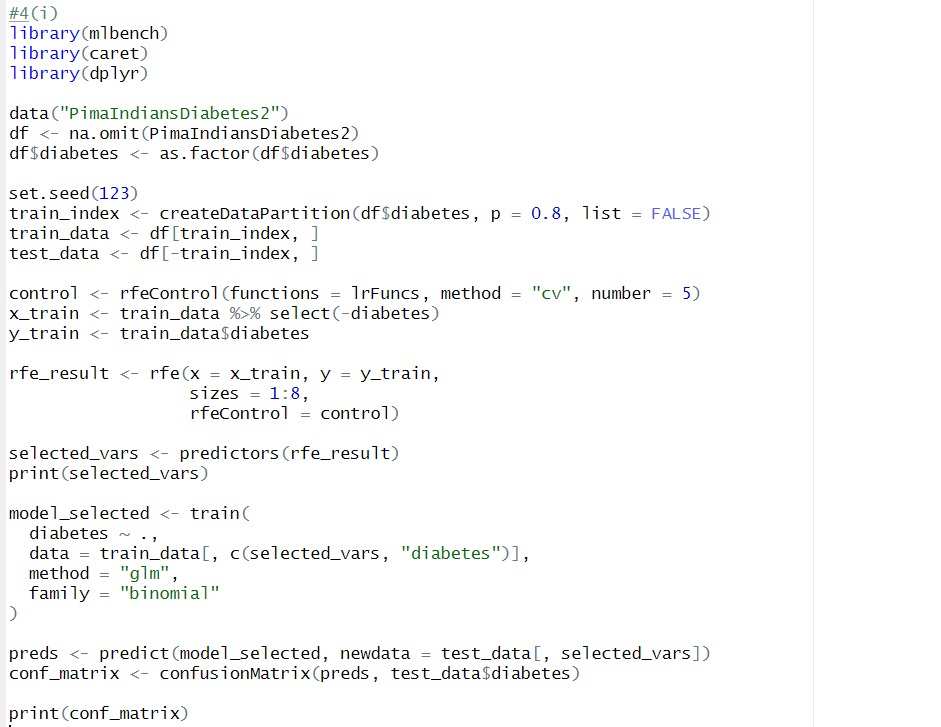


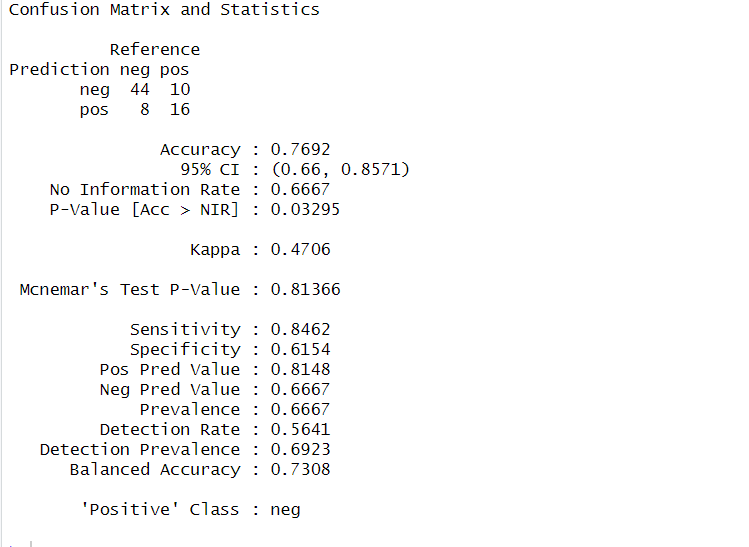




4. Train and compare model performance (with and without engineered/selected features).

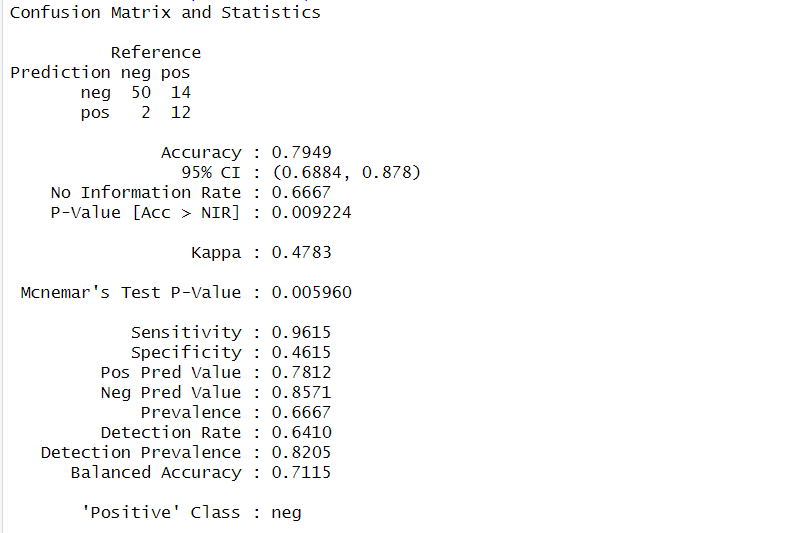
(i). Train model with selected features





(ii). "glucose", "mass", "bmi\_age\_ratio"





**EXPERIMENT- 8**

Exercise: Build and evaluate a logistic regression model to predict passenger survival on the Titanic.

∙ Apply preprocessing to dataset

∙ Fit and evaluate a logistic regression model

∙ Understand how categorical and continuous variables affect model performance

Expected Output

∙ Model Summary

∙ Confusion Matrix

∙ Accuracy and other metrics

