**UNIT 3: Working with Data Frames, File Operations in R**

**1. Data Frame**

customer\_data <- data.frame(

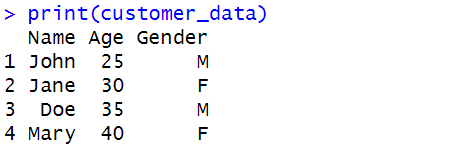
Name = c("John", "Jane", "Doe", "Mary"),

Age = c(25, 30, 35, 40),

Gender = c("M", "F", "M", "F")

)

print(customer\_data) **Output:**

****

**2. Extract Height Column**

df <- data.frame(

age = c(25, 30, 35, 40),

height = c(170, 165, 180, 175)

)

height\_vector <- df$height

print(height\_vector)

**Output:**

****

**3. Melt function**

library(reshape2)

sale <- data.frame(

region = c("North", "South", "East", "West"),

Q1 = c(1000, 1500, 1200, 1300),

Q2 = c(1100, 1600, 1300, 1400),

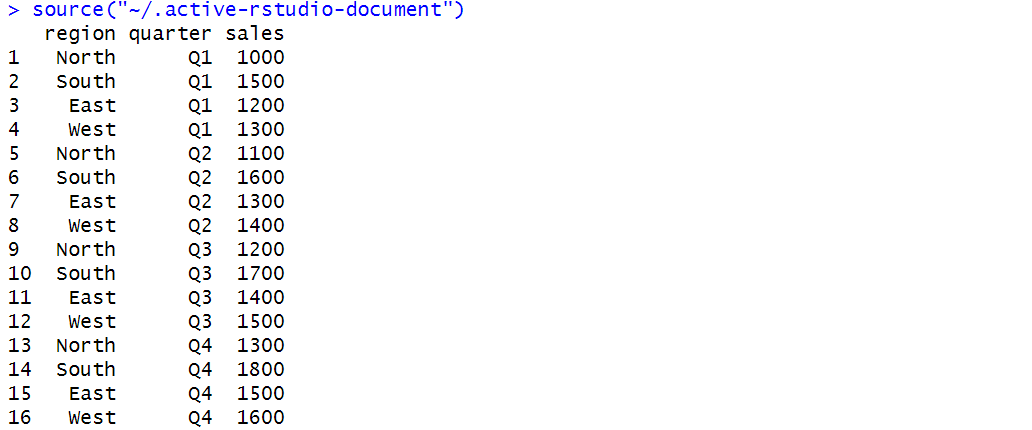
Q3 = c(1200, 1700, 1400, 1500),

Q4 = c(1300, 1800, 1500, 1600)

)

sale\_long <- melt(sale, id.vars = "region", variable.name = "quarter", value.name = "sales")

print(sale\_long) **Output:**

****

**4. Air Quality Data**

data("airquality")

is\_df <- is.data.frame(airquality)

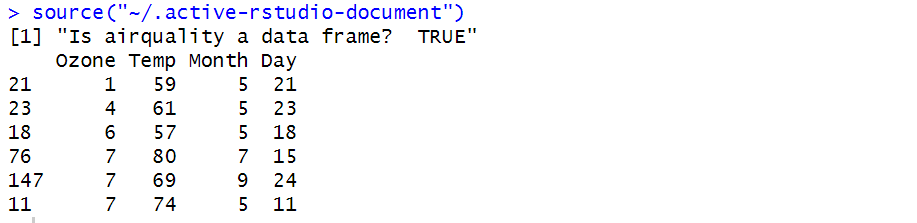
print(paste("Is airquality a data frame? ", is\_df))

airquality\_ordered <- airquality[order(airquality[,1], airquality[,2], na.last = TRUE), ]

airquality\_cleaned <- airquality\_ordered[, !(names(airquality\_ordered) %in% c("Solar.R", "Wind"))]

print(head(airquality\_cleaned))

**Output:**

****

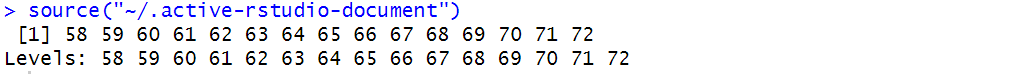
**5. Women Dataset**

data("women")

height\_factor <- factor(women$height)

print(height\_factor)

**Output:**

****

**6. Iris Dataset**

data("iris")

print(dim(iris))

print(str(iris))

print(summary(iris))

print(sapply(iris[, 1:4], sd))

library(dplyr)

iris\_stats <- iris %>%

group\_by(Species) %>%

summarise(across(where(is.numeric), list(mean = mean, sd = sd)))

print(iris\_stats)

print(quantile(iris$Sepal.Width))

print(quantile(iris$Sepal.Length))

iris1 <- iris %>%

mutate(Sepal.Length.Cate = cut(Sepal.Length, breaks = quantile(Sepal.Length, probs = seq(0, 1, 0.25)),

include.lowest = TRUE, labels = c("Short", "Medium", "Long", "Very Long")))

iris\_avg <- iris1 %>%

group\_by(Species, Sepal.Length.Cate) %>%

summarise(across(where(is.numeric), mean, na.rm = TRUE))

print(iris\_avg)

iris\_mean <- iris1 %>%

group\_by(Species, Sepal.Length.Cate) %>%

summarise(across(where(is.numeric), mean, na.rm = TRUE))

print(iris\_mean)

library(tidyr)

iris\_pivot <- iris1 %>%

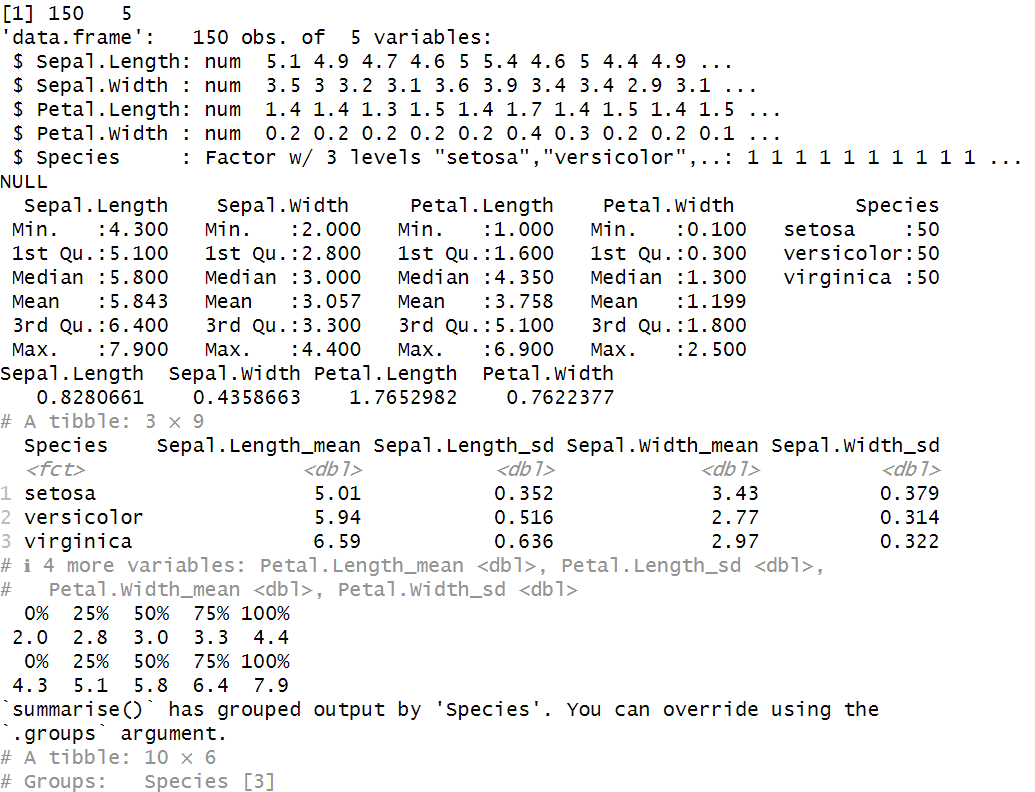
group\_by(Species, Sepal.Length.Cate) %>%

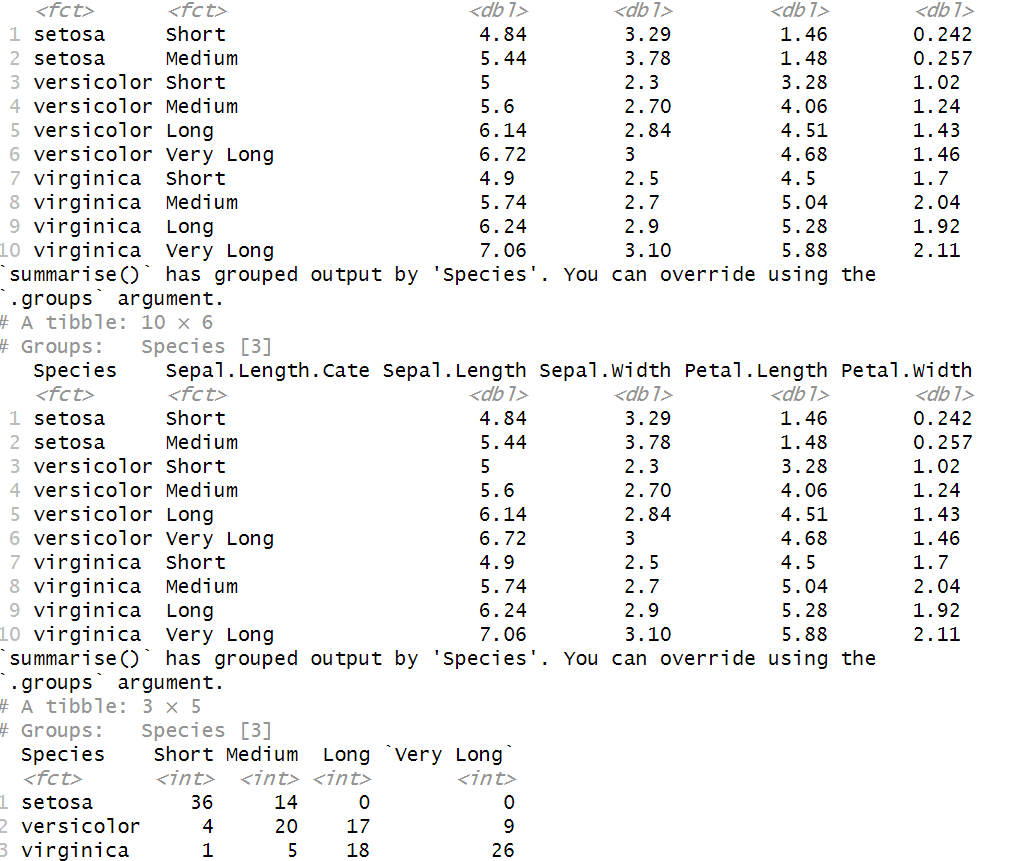
summarise(Count = n()) %>%

pivot\_wider(names\_from = Sepal.Length.Cate, values\_from = Count, values\_fill = 0)

print(iris\_pivot)

**Output:**

****

****

**7. Iris dataset into 80% training and 20% testing**

library(nnet)

library(caret)

data("iris")

set.seed(123)

trainIndex <- createDataPartition(iris$Species, p = 0.8, list = FALSE)

trainData <- iris[trainIndex, ]

testData <- iris[-trainIndex, ]

model <- multinom(Species ~ Petal.Length + Petal.Width, data = trainData)

pred\_probs <- predict(model, testData, type = "prob")

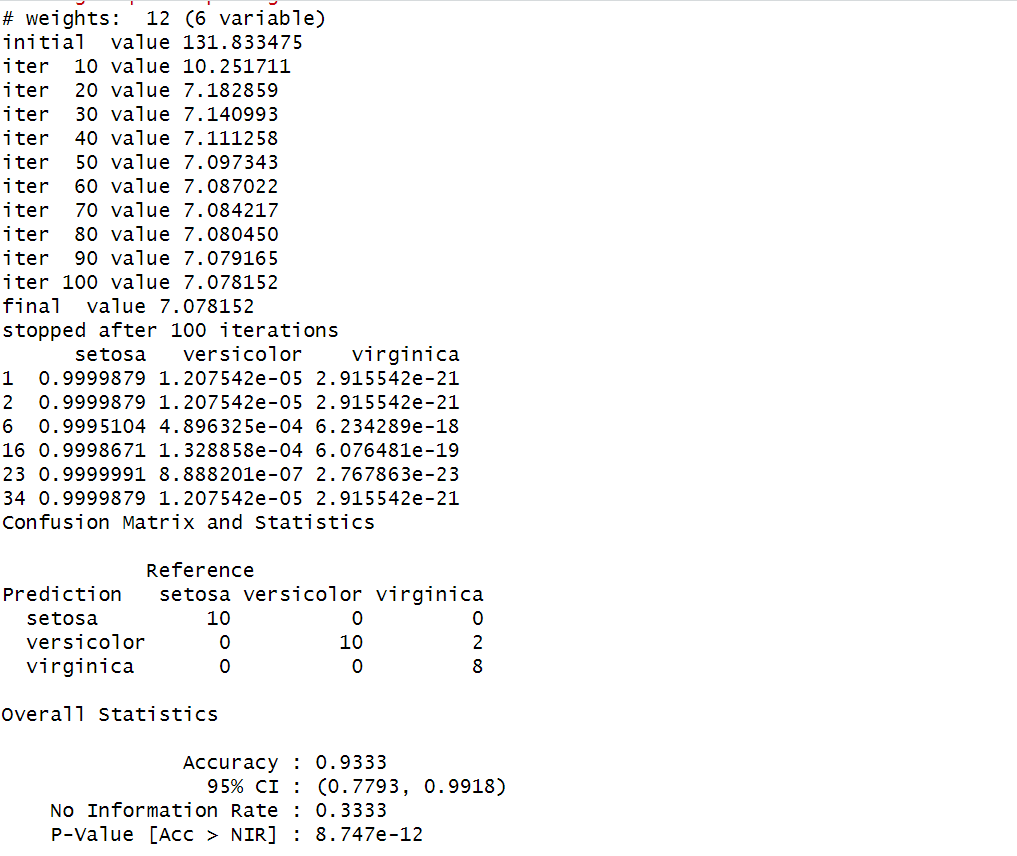
print(head(pred\_probs))

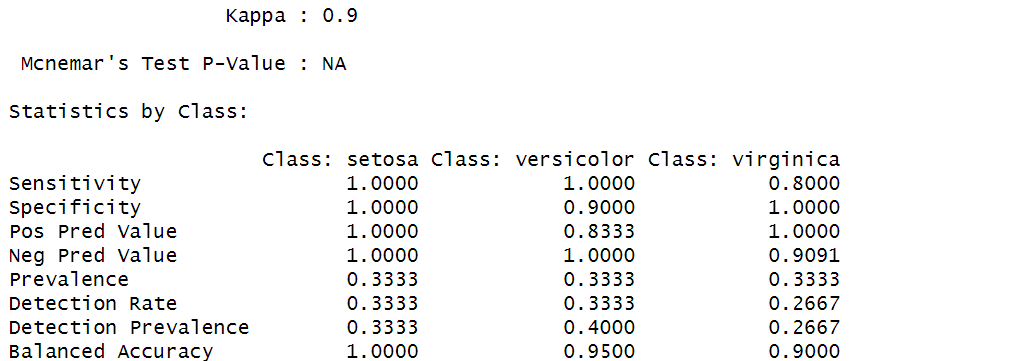
pred\_species <- predict(model, testData)

conf\_matrix <- confusionMatrix(pred\_species, testData$Species)

print(conf\_matrix)

**Output:**

****

****

**8. Air quality dataset and perform the required tasks**

data("airquality")

mean\_temp <- sum(airquality$Temp, na.rm = TRUE) / length(na.omit(airquality$Temp))

print(paste("Mean Temperature:", mean\_temp))

first\_five\_rows <- head(airquality, 5)

print(first\_five\_rows)

airquality\_subset <- airquality[, !(names(airquality) %in% c("Temp", "Wind"))]

print(head(airquality\_subset)) # Display first few rows

coldest\_day <- airquality[which.min(airquality$Temp), ]

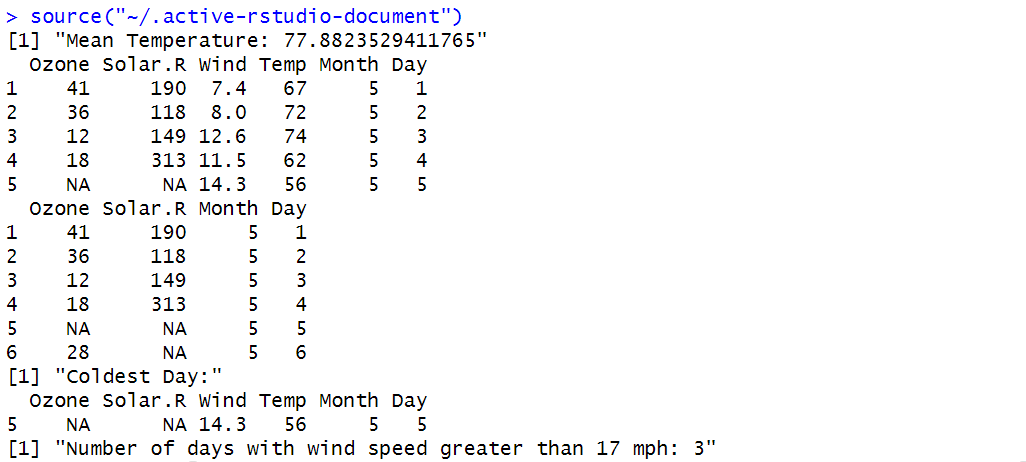
print("Coldest Day:")

print(coldest\_day)

wind\_high\_days <- sum(airquality$Wind > 17, na.rm = TRUE)

print(paste("Number of days with wind speed greater than 17 mph:", wind\_high\_days))

**Output:**

****

**9. Multiple Regression Model using the built-in ChickWeight dataset**

data("ChickWeight")

model <- lm(weight ~ Time + as.factor(Diet), data = ChickWeight) # Convert Diet to a factor

summary(model)

new\_data <- data.frame(Time = 10, Diet = factor(1, levels = levels(ChickWeight$Diet)))

predicted\_weight <- predict(model, new\_data)

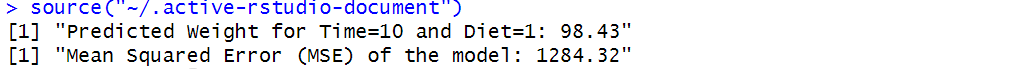
print(paste("Predicted Weight for Time=10 and Diet=1:", round(predicted\_weight, 2)))

actual\_weights <- ChickWeight$weight

predicted\_weights <- predict(model, ChickWeight)

mse <- mean((actual\_weights - predicted\_weights)^2, na.rm = TRUE)

print(paste("Mean Squared Error (MSE) of the model:", round(mse, 2))) **Output:**

****

**10. Titanic dataset**

library(ggplot2)

library(dplyr)

titanic\_df <- as.data.frame(Titanic)

ggplot(titanic\_df, aes(x = Class, y = Freq, fill = Survived)) +

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

labs(title = "Survival Count by Passenger Class", x = "Passenger Class", y = "Count") +

theme\_minimal()

ggplot(titanic\_df, aes(x = Class, y = Freq, fill = Survived)) +

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

facet\_wrap(~Sex) + # Split by Gender

labs(title = "Survival Count by Passenger Class and Gender", x = "Passenger Class", y = "Count") +

theme\_minimal()

if (!requireNamespace("datasets", quietly = TRUE)) install.packages("datasets")

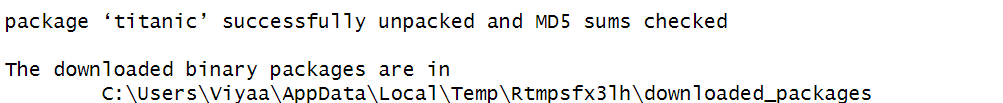
data("Titanic")

if (!requireNamespace("titanic", quietly = TRUE)) install.packages("titanic")

library(titanic)

titanic\_data <- titanic::titanic\_train

**Output:**

****

**11. Box Plot and Bar Plot**

data\_set <- c(6, 47, 49, 15, 43, 41, 7, 39, 43, 41, 36)

print(quantile(data\_set))

boxplot(data\_set, main = "Boxplot", col = "lightblue", notch = TRUE, horizontal = TRUE)

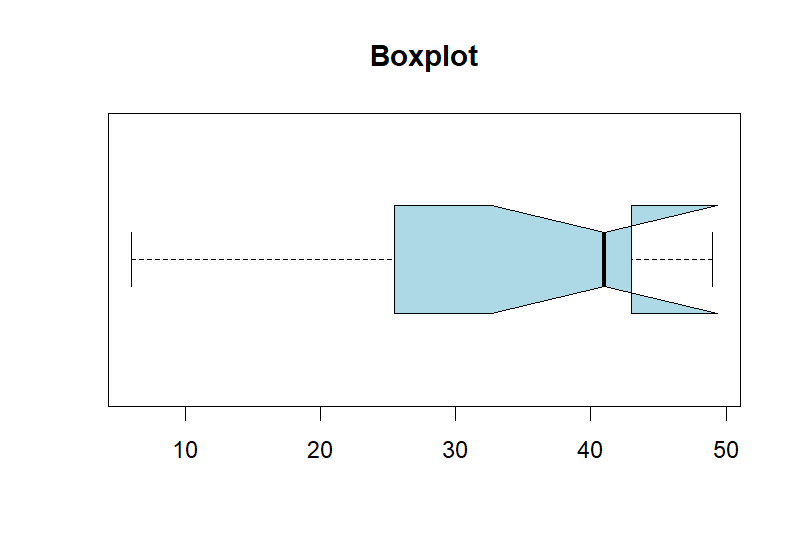
temp <- c(35, 42, 38, 25, 28, 36, 40)

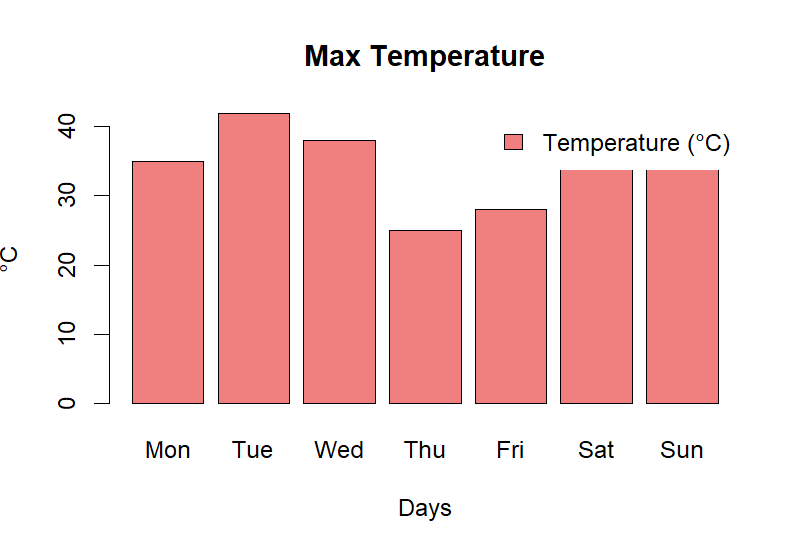
days <- c("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun")

barplot(temp, names.arg = days, col = "lightcoral", main = "Max Temperature", xlab = "Days", ylab = "°C")

legend("topright", legend = "Temperature (°C)", fill = "lightcoral", box.lty = 0)

****

****

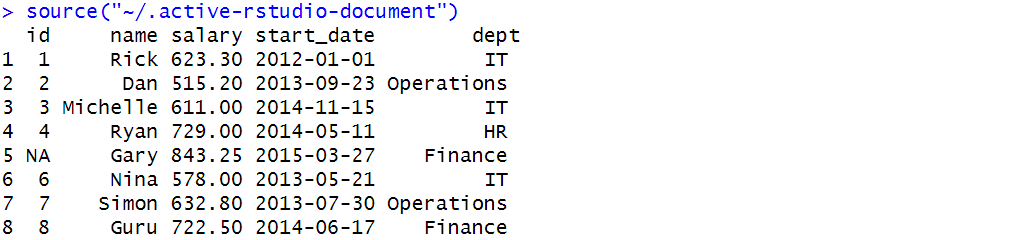
****

**12. Read .csv File**

data <- read.csv("input.csv", header = TRUE, sep = ",")

print(data)

**Output:**

****

**13. Titanic dataset Bar chart**

library(ggplot2)

library(dplyr)

titanic <- as.data.frame(Titanic) # Convert to a data frame

titanic <- titanic[rep(1:nrow(titanic), titanic$Freq), -5] # Expand rows by frequency

ggplot(titanic, aes(x = Class, fill = factor(Survived))) +

geom\_bar(position = "dodge") +

labs(title = "Survival Count Based on Passenger Class",

x = "Passenger Class",

y = "Count",

fill = "Survived") +

scale\_fill\_manual(values = c("red", "blue")) +

theme\_minimal()

ggplot(titanic, aes(x = Class, fill = factor(Sex))) +

geom\_bar(position = "dodge") +

labs(title = "Survival Count Based on Class and Gender",

x = "Passenger Class",

y = "Count",

fill = "Gender") +

scale\_fill\_manual(values = c("pink", "lightblue")) +

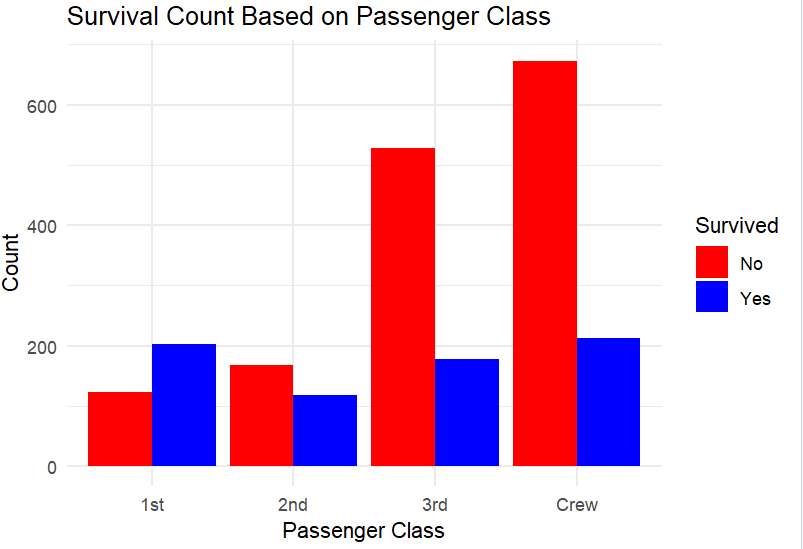
theme\_minimal()

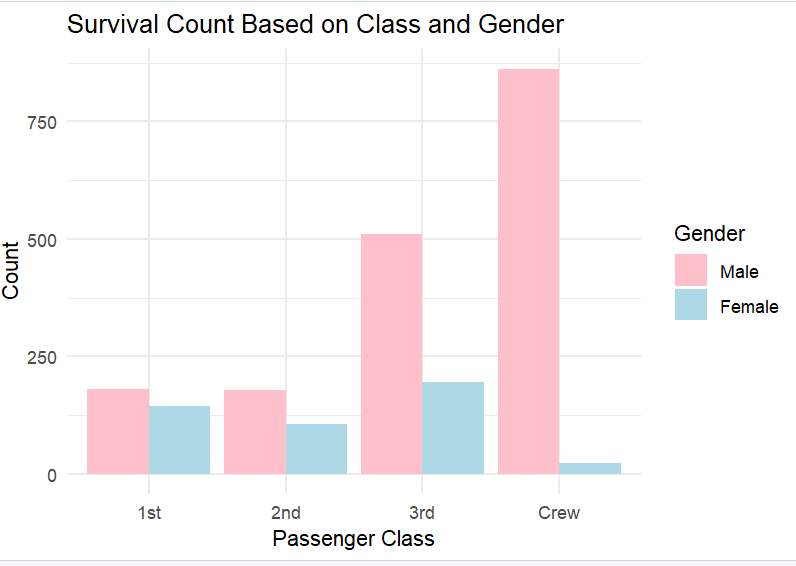
if ("datasets" %in% installed.packages()) {

data("Titanic")

}

**Output:**

****

****

**15. Stock Dataset**

product\_names <- c("Laptop", "Smartphone", "Headphones", "Tablet", "Smartwatch")

product\_prices <- c(800, 500, 100, 300, 200) # Prices in dollars

product\_quantities <- c(10, 25, 50, 15, 30) # Quantities in stock

product\_data <- data.frame(

Name = product\_names,

Price = product\_prices,

Quantity = product\_quantities

)

print("Product Data:")

print(product\_data)

average\_price <- mean(product\_data$Price)

print(paste("Average Price: $", round(average\_price, 2)))

high\_stock\_products <- product\_data$Name[product\_data$Quantity > 20]

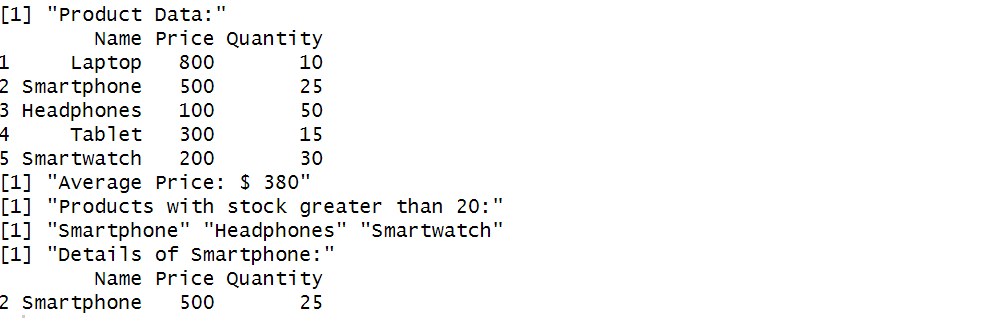
print("Products with stock greater than 20:")

print(high\_stock\_products)

smartphone\_info <- product\_data[product\_data$Name == "Smartphone", ]

print("Details of Smartphone:")

print(smartphone\_info)  
**Output:**

****

**16. Customer Information**

customer\_info <- data.frame(

CustomerID = c(101, 102, 103, 104),

Name = c("Alice", "Bob", "Charlie", "David"),

Email = c("alice@example.com", "bob@example.com", "charlie@example.com", "david@example.com")

)

purchase\_history <- data.frame(

CustomerID = c(101, 102, 104, 105), # Note: ID 105 is not in customer\_info

Product = c("Laptop", "Smartphone", "Tablet", "Headphones"),

Amount = c(800, 500, 300, 100)

)

merged\_data <- merge(customer\_info, purchase\_history, by = "CustomerID")

print("Merged Data (Inner Join):")

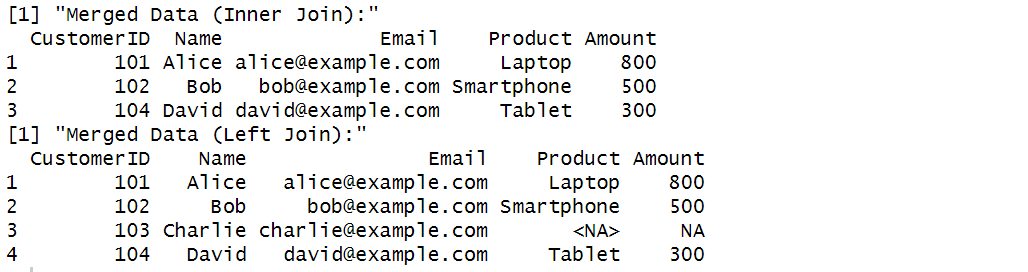
print(merged\_data)

merged\_left <- merge(customer\_info, purchase\_history, by = "CustomerID", all.x = TRUE)

print("Merged Data (Left Join):")

print(merged\_left)

**Output:**

****

**17. Advertising Data**

advertising\_data <- data.frame(

Month = c(1, 2, 3, 4, 5, 6),

Spends = c(1000, 4000, 5000, 4500, 3000, 4000),

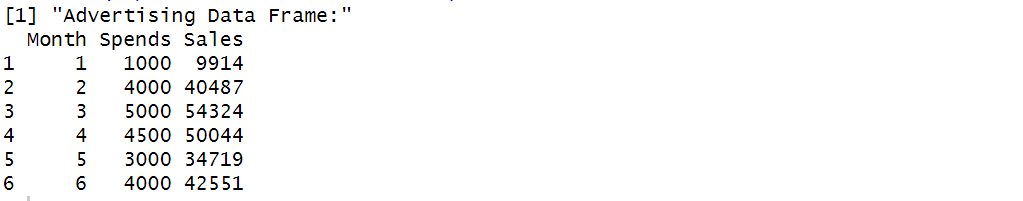
Sales = c(9914, 40487, 54324, 50044, 34719, 42551)

)

print("Advertising Data Frame:")

print(advertising\_data)

**Output:**

****

**library(ggplot2)**

ggplot(advertising\_data, aes(x = Spends, y = Sales)) +

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

geom\_smooth(method = "lm", color = "violet", se = FALSE) +

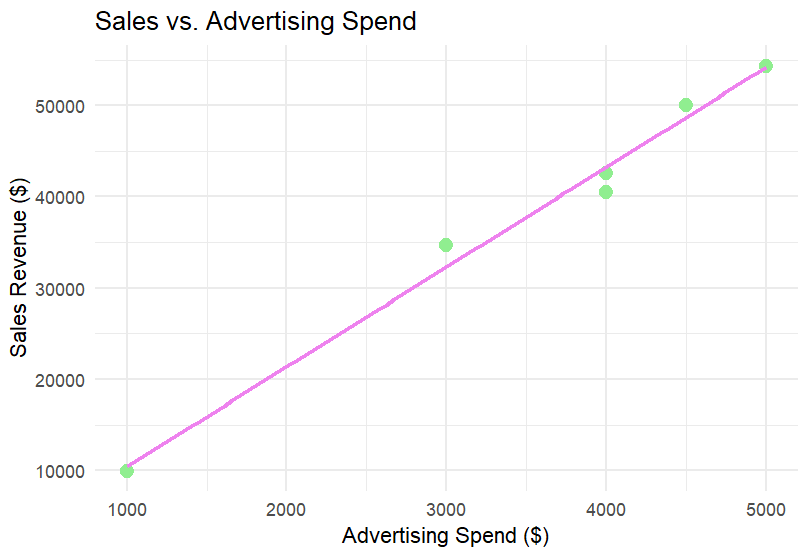
labs(title = "Sales vs. Advertising Spend",

x = "Advertising Spend ($)",

y = "Sales Revenue ($)") +

theme\_minimal()

**Output:**

****

**18. Height Weight data**

library(ggplot2)

height <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)

weight <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)

data <- data.frame(Height = height, Weight = weight)

print("Height-Weight Data:")

print(data)

ggplot(data, aes(x = Height, y = Weight)) +

geom\_point(color = "blue", size = 3) + # Scatter plot points

geom\_smooth(method = "lm", color = "red", se = FALSE) + # Regression line

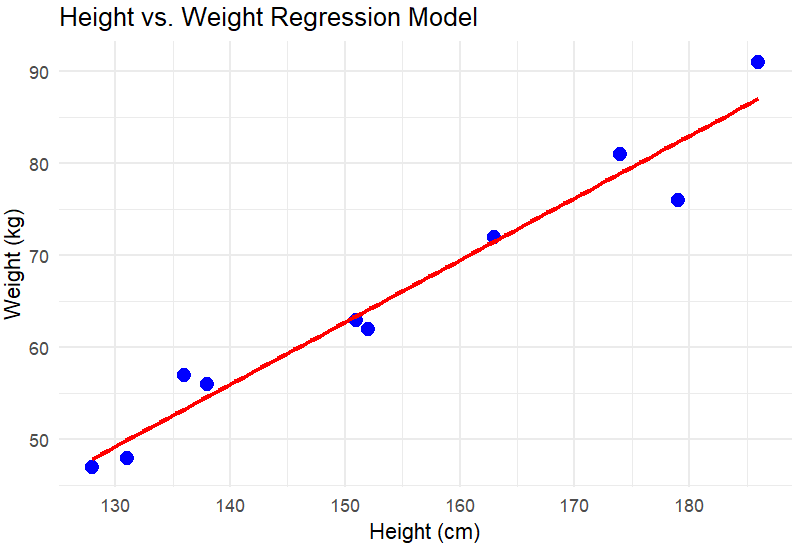
labs(title = "Height vs. Weight Regression Model",

x = "Height (cm)",

y = "Weight (kg)") +

theme\_minimal()

**Output:**

****

**19. Mean Median Mode**employee\_data <- data.frame(

EmployeeID = 1:10,

Salary = c(50000, 55000, 60000, 52000, 70000, 45000, 48000, 53000, 60000, 50000) # Salary in dollars

)

print("Employee Salary Data:")

print(employee\_data)

mean\_salary <- mean(employee\_data$Salary)

print(paste("Mean Salary: $", mean\_salary))

median\_salary <- median(employee\_data$Salary)

print(paste("Median Salary: $", median\_salary))

get\_mode <- function(x) {

uniq\_vals <- unique(x)

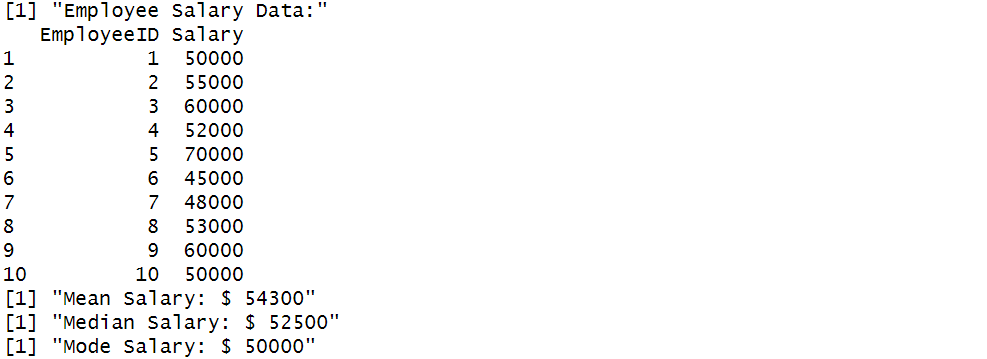
uniq\_vals[which.max(tabulate(match(x, uniq\_vals)))]

}

mode\_salary <- get\_mode(employee\_data$Salary)

print(paste("Mode Salary: $", mode\_salary))

**Output:**

****

**20. Temperature Dataset**

daily\_temps <- c(22, 24, 19, 25, 21, 20, 23, 26, 28, 22,

24, 23, 27, 21, 19, 20, 22, 24, 25, 26,

23, 22, 21, 27, 28, 25, 24, 23, 26, 22)

print("Daily Temperature Data:")

print(daily\_temps)

mean\_temp <- mean(daily\_temps)

print(paste("Mean Temperature: ", round(mean\_temp, 2), "°C"))

median\_temp <- median(daily\_temps)

print(paste("Median Temperature: ", median\_temp, "°C"))

get\_mode <- function(x) {

uniq\_vals <- unique(x)

uniq\_vals[which.max(tabulate(match(x, uniq\_vals)))]

}

mode\_temp <- get\_mode(daily\_temps)

print(paste("Mode Temperature: ", mode\_temp, "°C"))

range\_temp <- range(daily\_temps)

temp\_range <- diff(range\_temp) # Difference between max and min

print(paste("Temperature Range: ", temp\_range, "°C"))

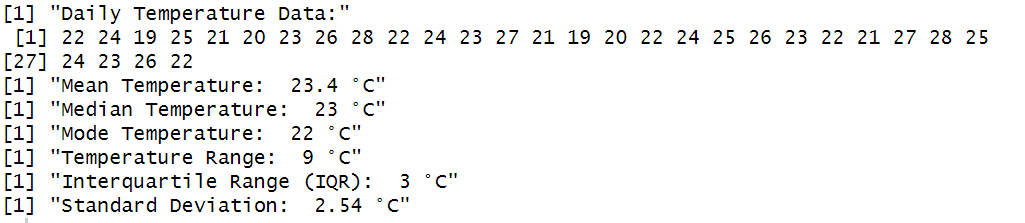
iqr\_temp <- IQR(daily\_temps)

print(paste("Interquartile Range (IQR): ", iqr\_temp, "°C"))

std\_dev\_temp <- sd(daily\_temps)

print(paste("Standard Deviation: ", round(std\_dev\_temp, 2), "°C"))

**Output:**

****