**Week -11:**

**79.Generate 100 random numbers from a normal distribution.**

random\_numbers <- rnorm(100, mean = 0, sd = 1)

print(random\_numbers)

**80. Generate reproducible results using set.Seed().**

set.seed(123) # Setting seed for reproducibility

random\_numbers <- rnorm(100, mean = 0, sd = 1)

print(random\_numbers)

**81. Combine multiple plots using gridExtra.**

install.packages("ggplot2")

install.packages("gridExtra")

p1 <- ggplot(mtcars, aes(x = mpg)) + geom\_histogram(binwidth = 2, fill = "blue")

p2 <- ggplot(mtcars, aes(x = hp, y = mpg)) + geom\_point(color = "red")

grid.arrange(p1, p2, ncol = 2) # Arrange plots side by side

**82. Generate 100 random numbers from a normal distribution using R**

random\_numbers <- rnorm(100, mean = 50, sd = 10)

print(random\_numbers)

**83. Write a for loop to iterate over a vector.**

# Create a vector

numbers <- c(10, 20, 30, 40, 50)

for (i in 1:length(numbers)) {

print(numbers[i])

}

**84. Combine two data frames by a common column.**

df1 <- data.frame(ID = c(1, 2, 3), Name = c("raj", "ravi", "rahul"))

df2 <- data.frame(ID = c(1, 2, 3), Age = c(25, 30, 35))

merged\_df <- merge(df1, df2, by = "ID") # Merge by common column "ID"

print(merged\_df)

**85. Calculate the difference in days between two dates.**

date1 <- as.Date("2024-01-01")

date2 <- as.Date("2025-01-01")

days\_diff <- as.numeric(difftime(date2, date1, units = "days"))

print(days\_diff)

**86. Remove rows with any missing values.**

df <- data.frame(A = c(1, NA, 3), B = c(4, 5, NA))

clean\_df <- na.omit(df) # Removes rows with missing values

print(clean\_df)

**Assessment-12**

**Automate a weekly report generation process in R.**

**Week-12:**

**87. Create a factor with levels "low", "medium", and "high" using R**

**# Create a factor variable**

levels\_vector <- factor(c("low", "medium", "high", "low", "high", "medium"),

levels = c("low", "medium", "high"))

# Print the factor variable

print(levels\_vector)

**88. Print the sum of the first 100 natural numbers.**

sum\_100 <- sum(1:100)

print(sum\_100)

**89. Create a vector of even numbers from 1 to 20 and calculate their mean**

# Create a vector of even numbers

even\_numbers <- seq(2, 20, by = 2)

# Calculate the mean

mean\_even <- mean(even\_numbers)

# Print results

print(even\_numbers)

print(mean\_even)

**90. Create a 3x3 matrix and calculate its determinant.**

# Create a 3x3 matrix

matrix\_3x3 <- matrix(c(2, 4, 6, 1, 3, 5, 7, 8, 9), nrow = 3, byrow = TRUE)

# Calculate the determinant

det\_value <- det(matrix\_3x3)

# Print results

print(matrix\_3x3)

print(det\_value)

91. Create a list containing a vector, a matrix, and a data frame. Access the second element.

# Create a vector

vec <- c(1, 2, 3, 4, 5)

# Create a 2x2 matrix

mat <- matrix(c(1, 2, 3, 4), nrow = 2)

# Create a data frame

df <- data.frame(Name = c("madhu", "Gopi"), Age = c(25, 30))

# Create a list containing all three elements

my\_list <- list(Vector = vec, Matrix = mat, DataFrame = df)

# Access the second element (Matrix)

second\_element <- my\_list[[2]]

# Print the second element

print(second\_element)

92. Filter rows in a data frame where age is greater than 25.

# Create a sample data frame

df <- data.frame(Name = c("sita", "geetha", "Satya", "ram"),

Age = c(22, 30, 27, 24))

# Filter rows where Age > 25

filtered\_df <- df[df$Age > 25, ]

# Print filtered data frame

print(filtered\_df)

93. **Assessment-12**

**How do you split a string into words in R?**

# Define a string

text <- "R programming is powerful"

# Split the string into words

words <- strsplit(text, " ")

# Print the result

print(words)

**Week-13:**

**93. Calculate the difference in days between two dates.**

# Define two dates

date1 <- as.Date("2024-01-01")

date2 <- as.Date("2025-10-01")

# Calculate difference in days

days\_diff <- as.numeric(difftime(date2, date1, units = "days"))

print(days\_diff)

# Alternative method using subtraction

days\_diff\_alt <- as.numeric(date2 - date1)

print(days\_diff\_alt) # Using subtraction

**94. Fit a linear regression model.**

# Create sample data

data <- data.frame(

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

y = c(2, 4, 5, 4, 5)

)

# Fit a linear regression model (y ~ x)

model <- lm(y ~ x, data = data)

# Print summary of the model

summary(model)

**95. Read a CSV file and display the first 5 rows.**

**96. Filter rows and calculate the mean of a column.**

# Create a sample data frame

df <- data.frame(Name = c("pandu", "geetha", "sudha", "raghav"),

Age = c(22, 30, 27, 24),

Salary = c(40000, 50000, 55000, 42000))

# Filter rows where Age > 25

filtered\_df <- df[df$Age > 25, ]

# Calculate the mean of the "Salary" column

mean\_salary <- mean(filtered\_df$Salary)

# Print results

print(filtered\_df)

print(mean\_salary)

**97. Create a histogram of random numbers generated from a normal distribution.**

# Generate 1000 random numbers from a normal distribution

random\_numbers <- rnorm(1000, mean = 50, sd = 10)

# Create a histogram

hist(random\_numbers,

main = "Histogram of Normally Distributed Numbers",

xlab = "Value",

col = "lightblue",

border = "black")

data <- data.frame(Value = rnorm(1000, mean = 50, sd = 10))

# Create a histogram

ggplot(data, aes(x = Value)) +

geom\_histogram(binwidth = 2, fill = "blue", color = "black") +

ggtitle("Histogram of Normally Distributed Numbers") +

xlab("Value") +

ylab("Frequency")

**98.Filter rows in a data frame where age is greater than 25.**

# Create a sample data frame

df <- data.frame(Name = c("mahesh", "madhu", "kamal", "bindu"),

Age = c(22, 30, 27, 24))

# Filter rows where Age > 25

filtered\_df <- df[df$Age > 25, ]

# Print the filtered data frame

print(filtered\_df)

**library(dplyr)**

filtered\_df <- df %>%

filter(Age > 25)

print(filtered\_df)

**99. Create a for loop to iterate over a vector and print each element.**

# Create a vector

numbers <- c(10, 20, 30, 40, 50)

for (i in 1:length(numbers)) {

print(numbers[i])

}

**100. Write a function to check if a number is prime.**

is\_prime <- function(n) {

if (n <= 1) {

return(FALSE) # 0 and 1 are not prime

}

for (i in 2:sqrt(n)) {

if (n %% i == 0) {

return(FALSE) # Divisible by a number other than 1 and itself

}

}

return(TRUE) # Prime number

}

# Test the function

print(is\_prime(7)) # TRUE

print(is\_prime(10)) # FALSE

**Assessment-13**

**How do you install and load packages in R?**

**1. Installing a Package**

Use install.packages() to install a package from CRAN:

Ex:install.packages("ggplot2") # Installs the ggplot2 package

**2. Loading a Package**

Use library() to load the installed package:

library(ggplot2) # Loads the ggplot2 package