**Week 5:**

**32. Data Import/Export (.csv, .xls, .txt) in R**

# Import CSV file

data\_csv <- read.csv("C:/Users/SYS28/Desktop/Data1.csv")

head(data\_csv)

# Import Excel file (requires readxl package)

install.packages("xlsx")

library(xlsx)

data\_xls <-read.xlsx("C:/Users/SYS28/Documents/data1.xlsx",sheetIndex=1)

head(data\_xls)

# Import Text file

data="C:/Users/SYS28/Documents/test.txt"

data\_txt <-read.table(data)

# Export DataFrame to CSV

write.csv(data\_csv, "output.csv")

# Export DataFrame to TXT

write.table(data\_txt, "output.txt", row.names = FALSE)

**33. Matrix Operations: Addition, Subtraction, Multiplication, Inverse, Transpose, Division**

# Get input matrix from user

mat1 <- matrix(c(1, 2, 3, 4), nrow = 2)

mat2 <- matrix(c(5, 6, 7, 8), nrow = 2)

# Addition

mat\_add <- mat1 + mat2

# Subtraction

mat\_sub <- mat1 - mat2

# Multiplication (element-wise)

mat\_mul <- mat1 \* mat2

# Matrix Multiplication (dot product)

mat\_dot <- mat1 %\*% mat2

# Transpose

mat\_transpose <- t(mat1)

# Inverse (if square and non-singular)

mat\_inverse <- solve(mat1)

# Division (element-wise)

mat\_div <- mat1 / mat2

# Print results

print(mat\_add)

print(mat\_sub)

print(mat\_mul)

print(mat\_dot)

print(mat\_transpose)

print(mat\_inverse)

print(mat\_div)

**34. Statistical Operations (Mean, Median, Mode, Standard Deviation)**

# Sample Data

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

# Mean

mean\_value <- mean(data)

# Median

median\_value <- median(data)

# Mode (custom function since R has no built-in mode)

mode\_function <- function(x) {

unique\_x <- unique(x)

unique\_x[which.max(tabulate(match(x, unique\_x)))]

}

mode\_value <- mode\_function(data)

# Standard Deviation

sd\_value <- sd(data)

# Print results

print(paste("Mean:", mean\_value))

print(paste("Median:", median\_value))

print(paste("Mode:", mode\_value))

print(paste("Standard Deviation:", sd\_value))

**35. Data Pre-processing (Handling Missing Values)**

# Sample Data with NA (missing values)

data <- data.frame(A = c(1, 2, NA, 4, 5), B = c(10, NA, 30, 40, 50))

# Remove rows with NA

data\_clean <- na.omit(data)

# Replace NA with mean

data$A[is.na(data$A)] <- mean(data$A, na.rm = TRUE)

# Print cleaned data

print(data\_clean)

print(data)

**36. Principal Component Analysis (PCA)**

# Load dataset

data(iris)

# Perform PCA (excluding categorical column)

pca\_result <- prcomp(iris[, -5], center = TRUE, scale. = TRUE)

# Print PCA summary

summary(pca\_result)

# Plot PCA

plot(pca\_result, type = "l")

**37. Simple Linear Regression in R**

# Load dataset

data(mtcars)

# Build regression model (Predict mpg using hp)

model <- lm(mpg ~ hp, data = mtcars)

# Summary of model

summary(model)

# Plot regression line

plot(mtcars$hp, mtcars$mpg, main = "Linear Regression", xlab = "Horsepower", ylab = "MPG")

abline(model, col = "red")

**38. K-Means Clustering (Iris Dataset)**

# Load dataset

data(iris)

# Perform K-Means clustering (3 clusters)

set.seed(123)

clusters <- kmeans(iris[, -5], centers = 3)

# Plot clusters

library(ggplot2)

ggplot(iris, aes(Petal.Length, Petal.Width, color = as.factor(clusters$cluster))) +

geom\_point(size = 3) +

**ggtitle("K-Means Clustering of Iris Dataset")**

**39. KNN Classification for Disease Diagnosis**

# Install & Load required library

install.packages("class")

library(class)

# Sample dataset (Diabetes dataset from UCI)

data <- iris

data$Species <- as.factor(data$Species)

# Split data

set.seed(123)

index <- sample(1:nrow(data), 0.7 \* nrow(data))

train <- data[index, ]

test <- data[-index, ]

# Apply KNN model

knn\_pred <- knn(train = train[, -5], test = test[, -5], cl = train$Species, k = 3)

# Confusion matrix

table(test$Species, knn\_pred)

**Assessment-5: Calculate Mean, Median, Variance, and Standard Deviation of mpg in mtcars**

# Load dataset

data(mtcars)

# Calculate required statistics

mean\_mpg <- mean(mtcars$mpg)

median\_mpg <- median(mtcars$mpg)

variance\_mpg <- var(mtcars$mpg)

sd\_mpg <- sd(mtcars$mpg)

# Print results

print(paste("Mean of MPG:", mean\_mpg))

print(paste("Median of MPG:", median\_mpg))

print(paste("Variance of MPG:", variance\_mpg))

print(paste("Standard Deviation of MPG:", sd\_mpg))