1. **R Expressions & Data Structures**

Program:

Expression <- function(x,y)

{

obj <- list(X=x, Y=y, Result=NULL)

class(obj) <- "Expression"

return(obj)

}

evaluate.Expression <- function(obj)

{

obj$Result <- obj$X^2 + obj$Y^2

return(obj)

}

print.Expression <- function(obj)

{

cat("X:", obj$X, " Y:", obj$Y, "\n")

cat("Result (x^2 + y^2):", obj$Result, "\n")

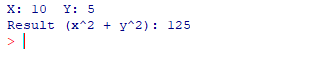
}

exp1 <- Expression(10,5)

exp1 <- evaluate.Expression(exp1)

print(exp1)

**Output:**



**2. Manipulation of Vectors and Matrices**

Program:

setClass("MathData",

slots = list(Vec = "numeric", Mat = "matrix"))

setGeneric("vectorOps", function(object) standardGeneric("vectorOps"))

setMethod("vectorOps", "MathData", function(object){

cat("Vector Elements:", object@Vec, "\n")

cat("First Element:", object@Vec[1], "\n")

cat("Subset (2nd to 4th):", object@Vec[2:4], "\n")

cat("Sum of Vector:", sum(object@Vec), "\n")

cat("Mean of Vector:", mean(object@Vec), "\n")

cat("Sorted Vector:", sort(object@Vec), "\n\n")

})

setGeneric("matrixOps", function(object) standardGeneric("matrixOps"))

setMethod("matrixOps", "MathData", function(object){

cat("Matrix:\n"); print(object@Mat)

cat("\nTranspose:\n"); print(t(object@Mat))

cat("\nFirst Row:", object@Mat[1,], "\n")

cat("Second Column:", object@Mat[,2], "\n")

cat("\nMatrix Multiplication (Mat %\*% t(Mat)):\n")

print(object@Mat %\*% t(object@Mat))

})

vec\_input <- as.numeric(unlist(strsplit(readline(prompt="Enter vector elements separated by space: "), " ")))

mat\_input <- matrix(as.numeric(unlist(strsplit(readline(prompt="Enter 9 numbers for 3x3 matrix: "), " "))),

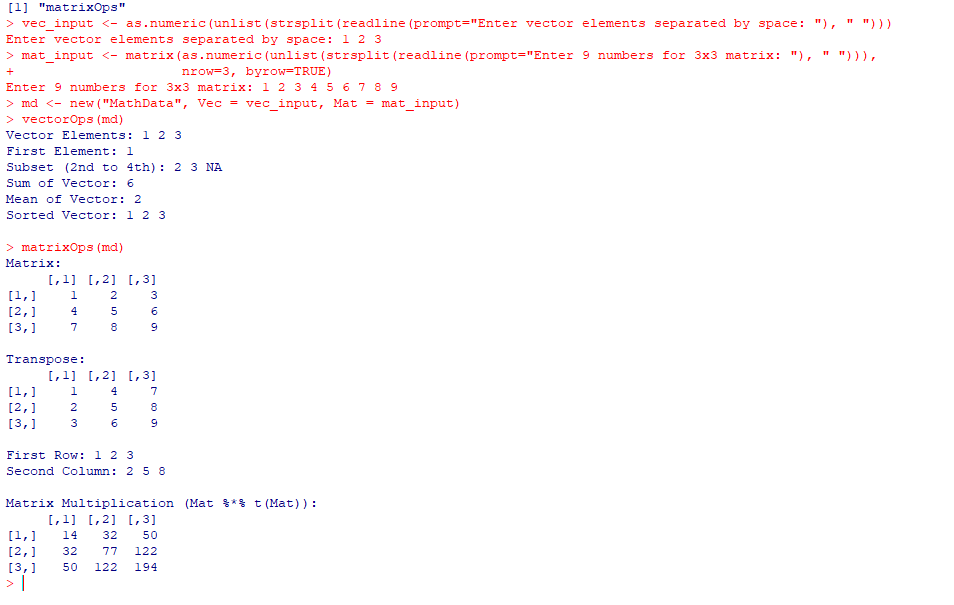
nrow=3, byrow=TRUE)

md <- new("MathData", Vec = vec\_input, Mat = mat\_input)

vectorOps(md)

matrixOps(md)

**Output:**



**3.** **Operators on Factors in R**

Program:

Category <- function(items){

obj <- list(Items = factor(items)) # Store as factor

class(obj) <- "Category" # Assign S3 class

return(obj)

}

analyze.Category <- function(obj){

cat("Factor Items:\n"); print(obj$Items)

cat("\nLevels of Factor:\n"); print(levels(obj$Items))

cat("\nFrequency Count:\n"); print(summary(obj$Items))

cat("\nSorted Factor:\n"); print(sort(obj$Items))

cat("\nReordered Factor (reverse levels):\n")

new\_fac <- factor(obj$Items, levels = rev(levels(obj$Items)))

print(new\_fac)

}

user\_items <- unlist(strsplit(readline(prompt="Enter categories separated by space: "), " "))

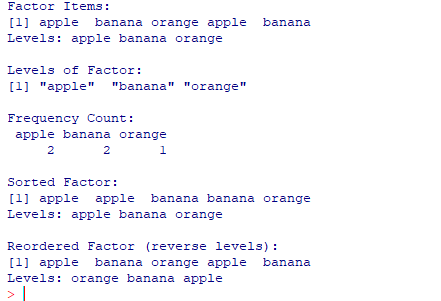
catObj <- Category(user\_items)

analyze.Category(catObj)

**Input:**

Enter categories separated by space: apple banana orange apple banana

**Output:**



**4. Data Frames in R**

Program:

StudentData <- function(ids, names, marks){

obj <- list(

DF = data.frame(ID = ids, Name = names, Marks = marks)

)

class(obj) <- "StudentData" # Assign S3 class

return(obj)

}

analyze.StudentData <- function(obj){

cat("Original Data Frame:\n")

print(obj$DF)

cat("\nAccessing 1st Row:\n")

print(obj$DF[1, ])

cat("\nAccessing 'Name' Column:\n")

print(obj$DF$Name)

cat("\nAdding Grade Column:\n")

obj$DF$Grade <- ifelse(obj$DF$Marks >= 90, "A",

ifelse(obj$DF$Marks >= 75, "B", "C"))

print(obj$DF)

cat("\nFiltering Students with Marks >= 80:\n")

print(subset(obj$DF, Marks >= 80))

cat("\nStatistical Summary of Marks:\n")

print(summary(obj$DF$Marks))

}

ids <- as.integer(unlist(strsplit(readline("Enter student IDs separated by space: "), " ")))

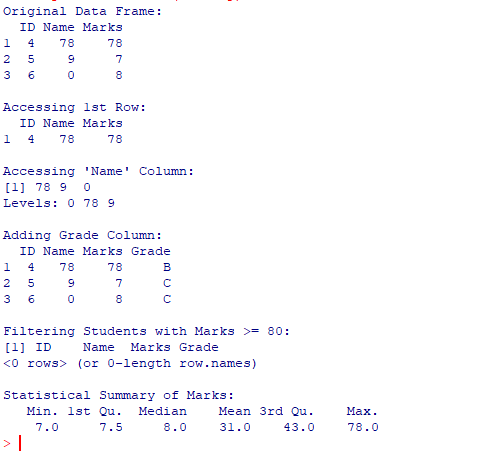
names <- unlist(strsplit(readline("Enter student names separated by space: "), " "))

marks <- as.numeric(unlist(strsplit(readline("Enter student marks separated by space: "), " ")))

studObj <- StudentData(ids, names, marks)

analyze.StudentData(studObj)

**Output:**



**5. Lists and operators**

Program:

Document<-function(text){

obj<-list(Text=text,Words=unlist(strsplit(tolower(text),"\\s+")))

class(obj)<-"Document"

return(obj)

}

wordcount.Document<-function(obj){

tbl<-table(obj$Words)

return(tbl)

}

print.Document<-function(obj){

cat("Document:",obj$Text,"\n")

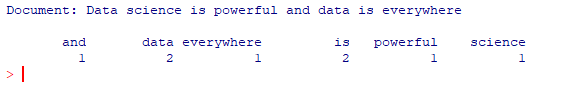
print(wordcount.Document(obj))

}

doc1<-Document("Data science is powerful and data is everywhere")

print(doc1)

**Output:**



**6. Working with Looping Statements in R**

Program:

LoopDemo <- function(nums, factNum){

obj <- list(

Numbers = nums,

FactorialNumber = factNum

)

class(obj) <- "LoopDemo"

return(obj)

}

process <- function(obj) {

UseMethod("process")

}

process.LoopDemo <- function(obj){

total <- 0

for(i in obj$Numbers){

total <- total + i

}

cat("Sum of Numbers using for loop:", total, "\n")

n <- obj$FactorialNumber

fact <- 1

while(n > 0){

fact <- fact \* n

n <- n - 1

}

cat("Factorial using while loop:", fact, "\n")

cat("Numbers generated using repeat loop:\n")

x <- 1

repeat{

cat(x, " ")

x <- x + 2

if(x > 20) break

}

cat("\n")

}

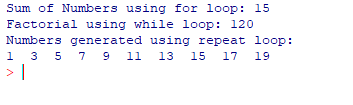
nums <- c(1, 2, 3, 4, 5)

factNum <- 5

loopObj <- LoopDemo(nums, factNum)

process(loopObj)

**Output:**



**7. Graphs in R**

data(mtcars)

hist(mtcars$mpg,

main = "Histogram of Miles per Gallon",

xlab = "Miles per Gallon",

col = "lightblue", border = "black")

boxplot(mtcars$hp,

main = "Boxplot of Horsepower",

ylab = "Horsepower",

col = "lightgreen")

plot(mtcars$hp, mtcars$mpg,

main = "Scatterplot of MPG vs Horsepower",

xlab = "Horsepower",

ylab = "Miles per Gallon",

col = "blue", pch = 19)

plot(mtcars$mpg,

type = "o",

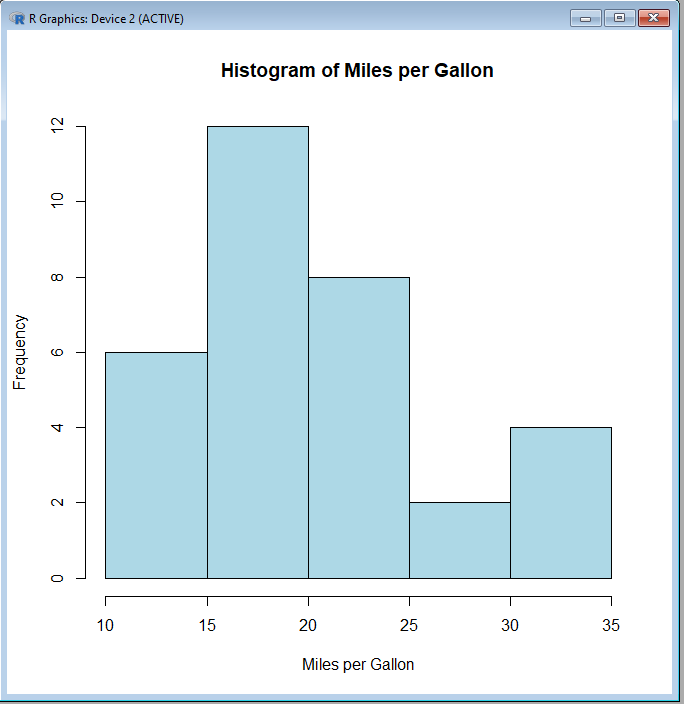
main = "Line Chart of MPG across Cars",

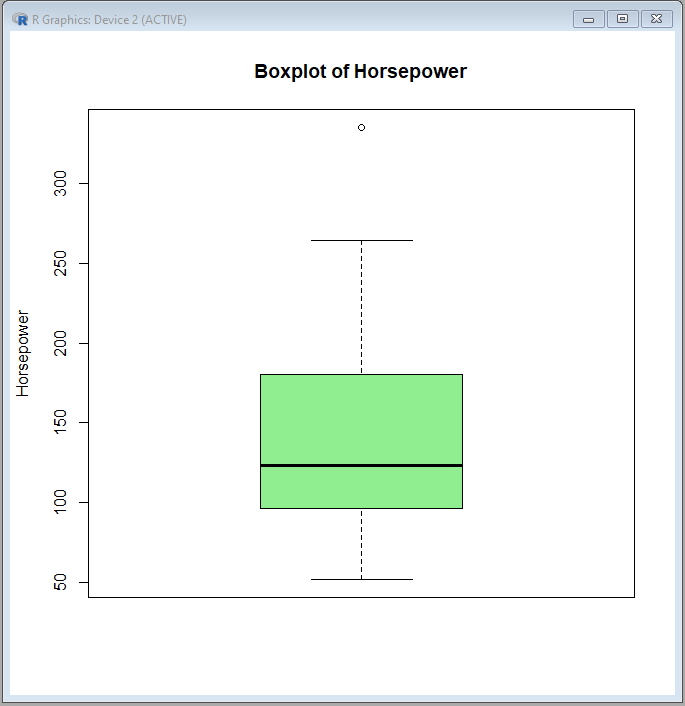
xlab = "Car Index",

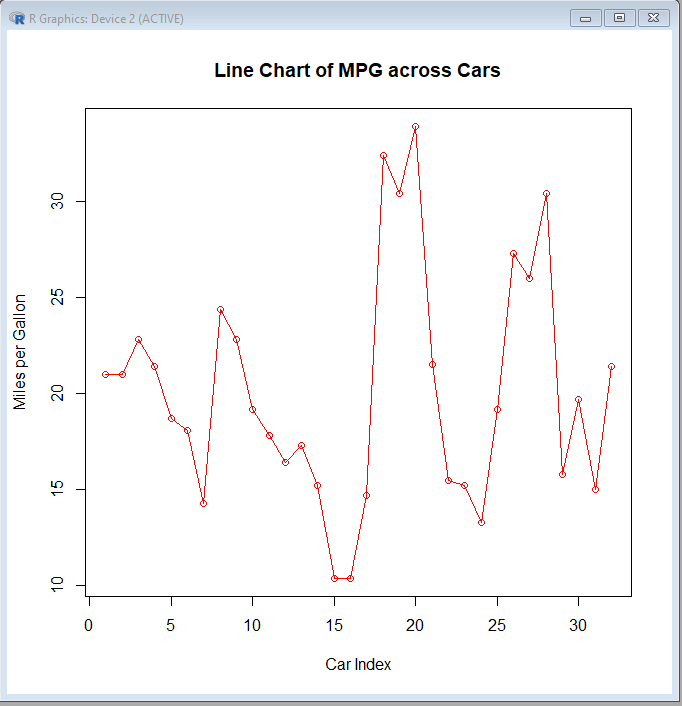
ylab = "Miles per Gallon",

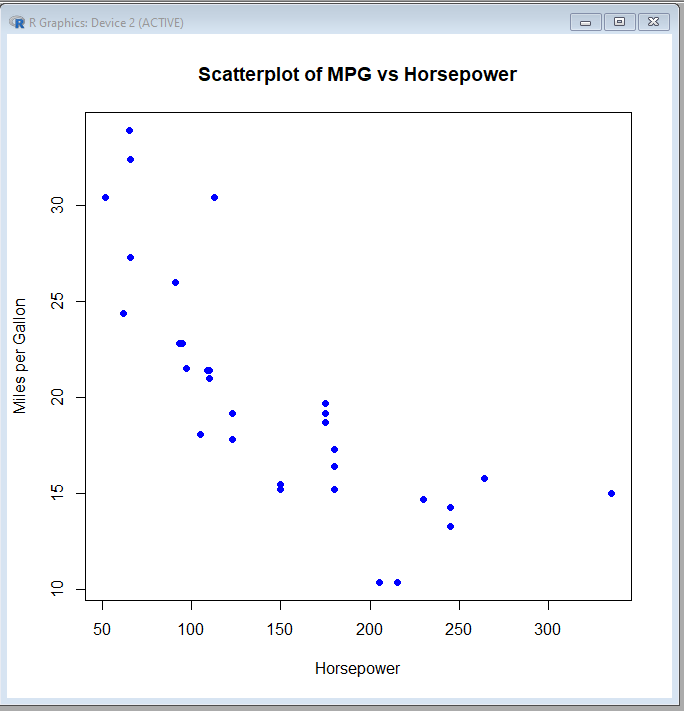
col = "red")

**Output:**









**8. 3D plots in R**

library(scatterplot3d)

x <- 1:10

y <- x^2

z <- x^3

scatterplot3d(x, y, z,

main = "3D Scatter Plot",

xlab = "X axis",

ylab = "Y axis",

zlab = "Z axis",

color = "blue",

pch = 19)

scatterplot3d(x, y, z, angle = 55)

s3d <- scatterplot3d(x, y, z)

fit <- lm(z ~ x + y) # linear model

s3d$plane3d(fit)

z <- outer(x, y, function(x, y) cos(sqrt(x^2 + y^2)))

persp(x, y, z,

col = "lightblue",

theta = 30, phi = 20, # rotation angles

expand = 0.5,

xlab = "X", ylab = "Y", zlab = "Z",

main = "3D Surface Plot (persp)")

**Output:**

