**Problem Statement 16:** Program to check if the input year is a leap year or not.

**Objective:** To check if the input year is a leap year or not.

**R Script:**

year <- as.integer(readline("Enter a year: "))

if (year %% 400 == 0) {

print("It is a leap year.")

} else if (year %% 100 == 0) {

print("It is not a leap year.")

} else if (year %% 4 == 0) {

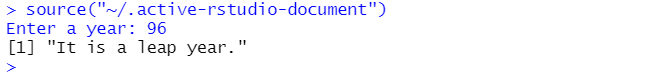
print("It is a leap year.")

} else {

print("It is not a leap year.")

}

**Output:**



**Problem Statement 17:**

Create, access, modify and delete following data structures in R

a) Vectors

b) Lists

c) Data Frame

d) Factor

e) Matrix

**Objective:**

To understand the working of different data structures in R.

**R Script + Output:**

**a) Vector:**

# CREATE VECTOR

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

print(vec)

# ACCESS ELEMENT

print(vec[1])

# MODIFY ELEMENT

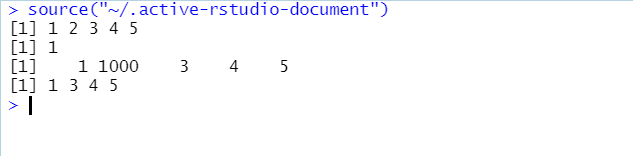
vec[2] <- 1000

print(vec)

# DELETE ELEMENT

vec <- vec[-2]

print(vec)



**b) Lists:**

# CREATE

lis <- list(a = 1, b = 2)

print(lis)

# ACCESS ELEMENT

print(lis$a)

# MODIFY ELEMENT

lis$a <- 1000

print(lis)

# DELETE ELEMENT

lis <- lis[-1]

print(lis)



**c) Data Frame:**

# CREATE

df <- data.frame(a = c(1,2,3), b = c(11,12,13))

print(df)

# ACCESS ELEMENT

print(df$a)

# MODIFY ELEMENT

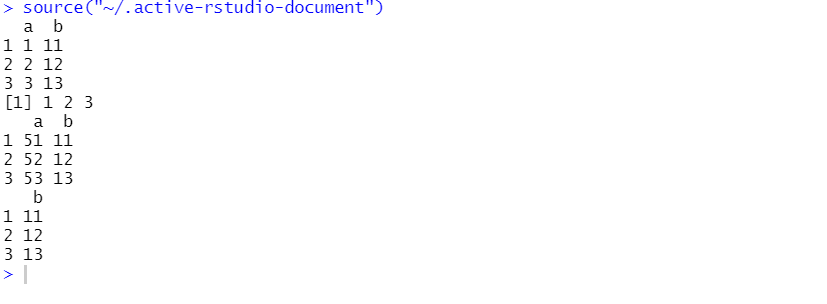
df$a <- 51:53

print(df)

# DELETE ELEMENT

df <- df[-1]

print(df)



**d) Factor:**

# CREATE

fac <- factor(c("a", "b", "c", "d", "e"))

print(fac)

# ACCESS ELEMENT

print(fac[1])

# MODIFY ELEMENT

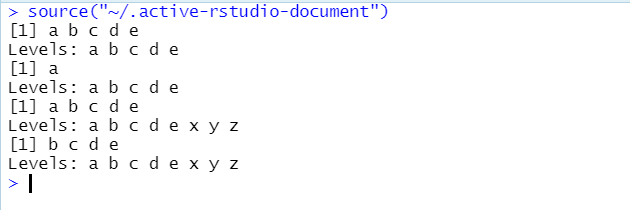
levels(fac) <- c("a", "b", "c", "d", "e", "x", "y", "z")

print(fac)

# DELETE ELEMENT

fac <- fac[-1]

print(fac)



**e) Matrix:**

# CREATE

mat <- matrix(1:12, nrow = 3)

print(mat)

# ACCESS ELEMENT

print(mat[2,3])

# MODIFY ELEMENT

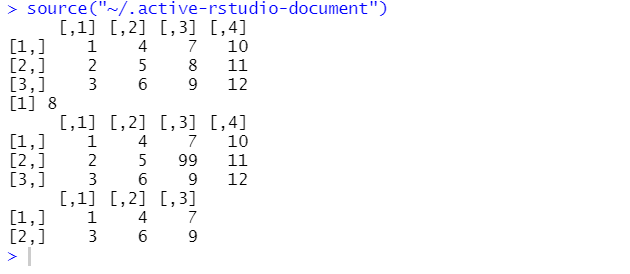
mat[2,3] <- 99

print(mat)

# DELETE ELEMENT

mat <- mat[-2,-4]

print(mat)



**Problem Statement 18:** Create a function to print squares of numbers in sequence.

**Objective:** To create a function to print squares of numbers in sequence.

**R Script + Output:**

print\_squares <- function(n) {

for (i in 1:n) {

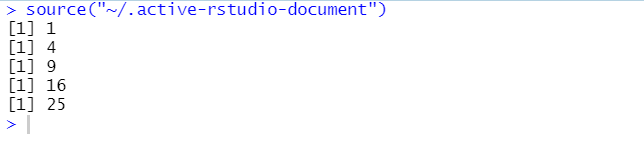
sq <- i^2

print(sq)

}

}

print\_squares(5)



**Problem Statement 19:** Demonstrate various Numerical, Character and Statistical functions used in R.

**Objective:**  To demonstrate various Numerical, Character and Statistical functions used in R.

**R Script + Output:**

# Numerical Functions

a <- 10.3

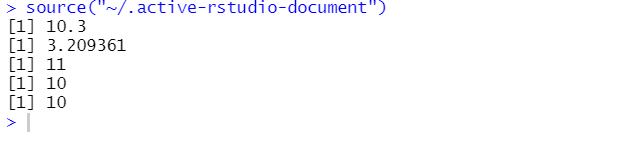
print(abs(a))

print(sqrt(a))

print(ceiling(a))

print(floor(a))

print(round(a))



# Character functions

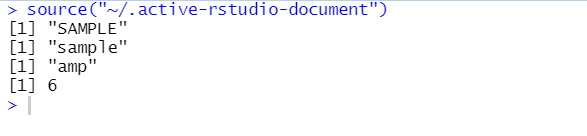
a <- "sample"

print(toupper(a))

print(tolower(a))

print(substr(a, 2, 4))

print(nchar(a))



# Statistical functions

a <- c(1:10)

print(mean(a))

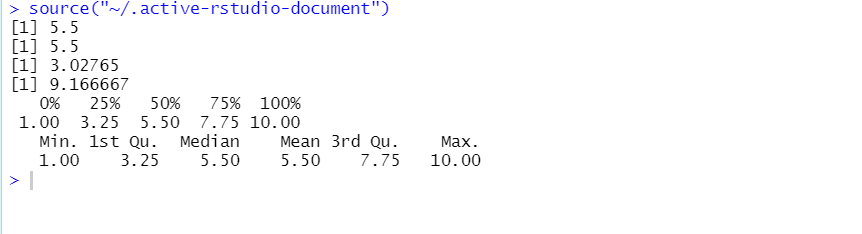
print(median(a))

print(sd(a))

print(var(a))

print(quantile(a))

print(summary(a))



**Problem Statement 20:**

The numbers below are the first ten days of rainfall amounts in 1996.Read them in

to a vector using the c() function 0.1, 0.6, 33.8, 1.9, 9.6, 4.3, 33.7, 0.3, 0.0, 0.1

a. What was the mean rainfall, how about the standard deviation?

b. Calculate the cumulative rainfall (’running total’) over these ten days.

Confirm that the last value of the vector that this produces is equal to the

total sum of the rainfall.

c. Which day saw the highest rainfall?

**R Script + Output:**

# Read in the rainfall amounts

rainfall <- c(0.1, 0.6, 33.8, 1.9, 9.6, 4.3, 33.7, 0.3, 0.0, 0.1)

# Calculate the mean rainfall

mean\_rainfall <- mean(rainfall)

print(mean\_rainfall)

# Calculate the standard deviation of the rainfall

sd\_rainfall <- sd(rainfall)

print(sd\_rainfall)

# Calculate the cumulative rainfall

cumulative\_rainfall <- cumsum(rainfall)

print(cumulative\_rainfall)

# Confirm that the last value of the cumulative rainfall vector is equal to the total sum of the rainfall

sum\_rainfall <- sum(rainfall)

if (cumulative\_rainfall[length(cumulative\_rainfall)] == sum\_rainfall) {

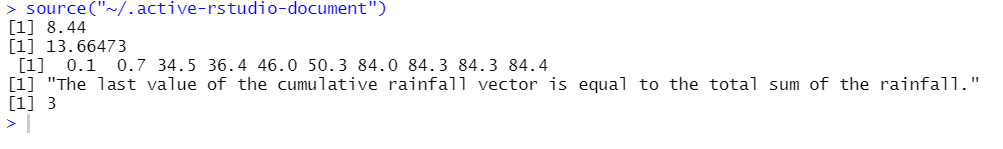
print("The last value of the cumulative rainfall vector is equal to the total sum of the rainfall.")

}

# Find the day with the highest rainfall

max\_day <- which.max(rainfall)

print(max\_day)



**Problem Statement 21:**

Demonstrate the various function used for Graphical Analysis like creating box

plot, scatters plot, line graph and pie charts and bar chart.

**Objective:**

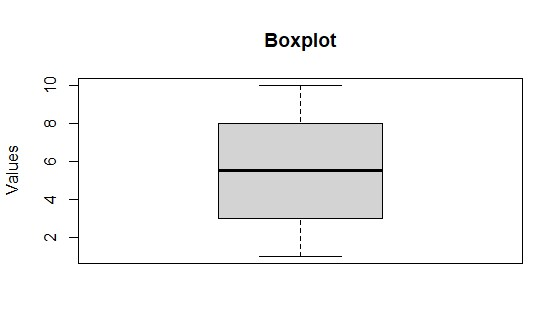
To demonstrate the various function used for Graphical Analysis like creating box

plot, scatters plot, line graph and pie charts and bar chart.

Box Plot:

x = sample(1:10)

boxplot(x, main="Boxplot", ylab="Values")

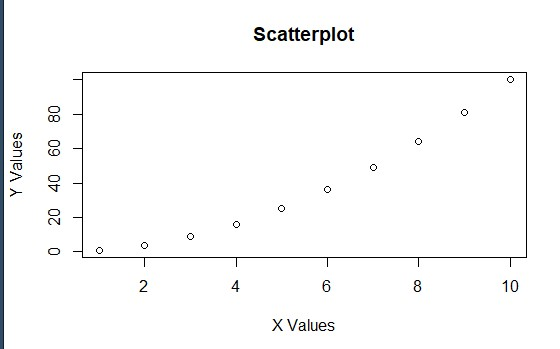


Scatter Plot:

x = c(1,2,3,4,5,6,7,8,9,10)

y = c(1,4,9,16,25,36,49,64,81,100)

plot(x, y, main="Scatterplot", xlab="X Values", ylab="Y Values")

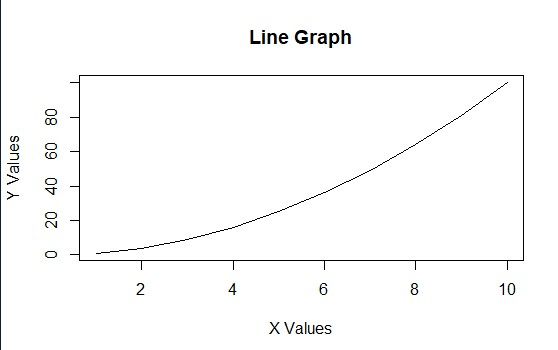


Line Graph:

x = c(1,2,3,4,5,6,7,8,9,10)

y = c(1,4,9,16,25,36,49,64,81,100)

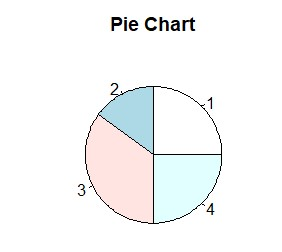
plot(x, y, type="l", main="Line Graph", xlab="X Values", ylab="Y Values")



Pie Chart:

x = c(25,15,35,25)

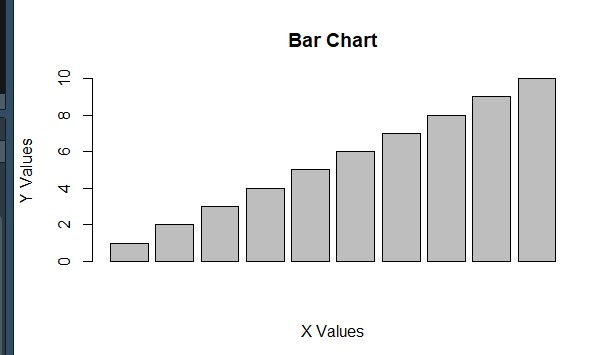
pie(x, main="Pie Chart")



Bar Chart:

x = c(1,2,3,4,5,6,7,8,9,10)

barplot(x, main="Bar Chart", xlab="X Values", ylab="Y Values")



**Problem Statement 22:** Demonstrate Implementation of ANOVA in R Studio.

**Objective:** To demonstrate Implementation of ANOVA in R Studio.

**R Script:**

# Create the data frame

heights <- data.frame(group = c(rep("men", 10), rep("women", 10), rep("children", 10)),

height = c(175, 180, 165, 170, 173, 176, 177, 182, 170, 173,

160, 162, 155, 156, 158, 159, 164, 161, 170, 172,

120, 122, 125, 126, 128, 130, 135, 131, 140, 142))

# Perform the ANOVA test

aov\_test <- aov(height ~ group, data = heights)

# Print the summary of the ANOVA test

print(summary(aov\_test))

**Output:**

Df Sum Sq Mean Sq F value Pr(>F)

group 2 10395 5198 141.8 4.78e-15 \*\*\*

Residuals 27 990 37

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1