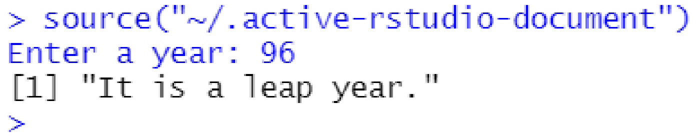
Problem Statement 16: Program to check if the input year is a leap year or not.

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:



```
> source("~/.active-rstudio-document")
Enter a year: 96
[1] "It is a leap year."
>
```

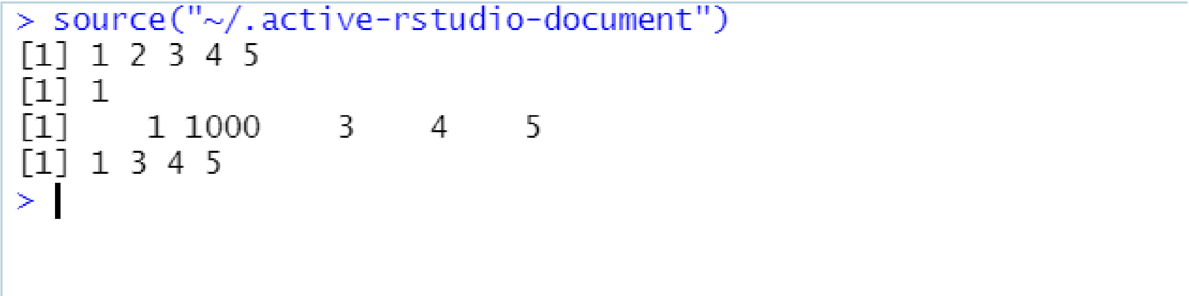
Problem Statement 17:
Create, access, modify and delete following data structures in R
a) Vectors
b) Lists
c) Data Frame
d) Factor
e) Matrix

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)
```

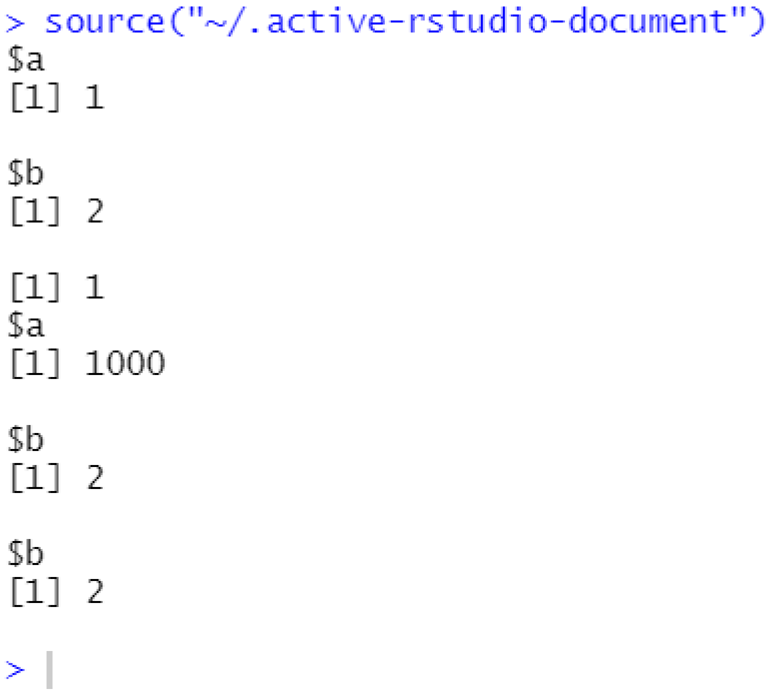


```
> source("~/.active-rstudio-document")
[1] 1 2 3 4 5
[1] 1
[1]    1 1000    3    4    5
[1] 1 3 4 5
> 
```

b) Lists:

```
# CREATE
lis <- list(a = 1, b = 2)
print(lis)

# ACCESS ELEMENT
print(lis$a)
```

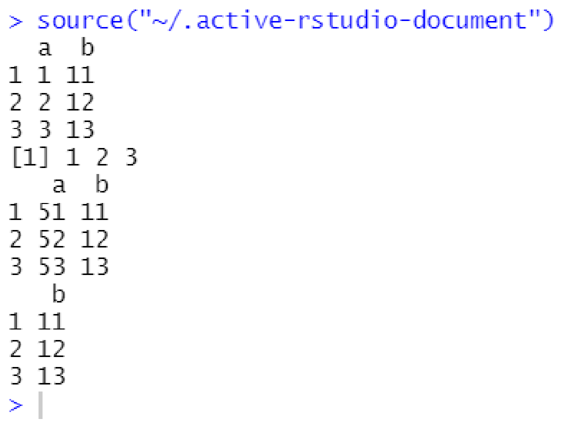
```
# MODIFY ELEMENT
lis$a <- 1000
print(lis)

# DELETE ELEMENT
lis <- lis[-1]
print(lis)
```



```
> source("~/.active-rstudio-document")
$a
[1] 1

$b
[1] 2

[1] 1
$a
[1] 1000

$b
[1] 2

$b
[1] 2

> 
```

c) Data Frame:

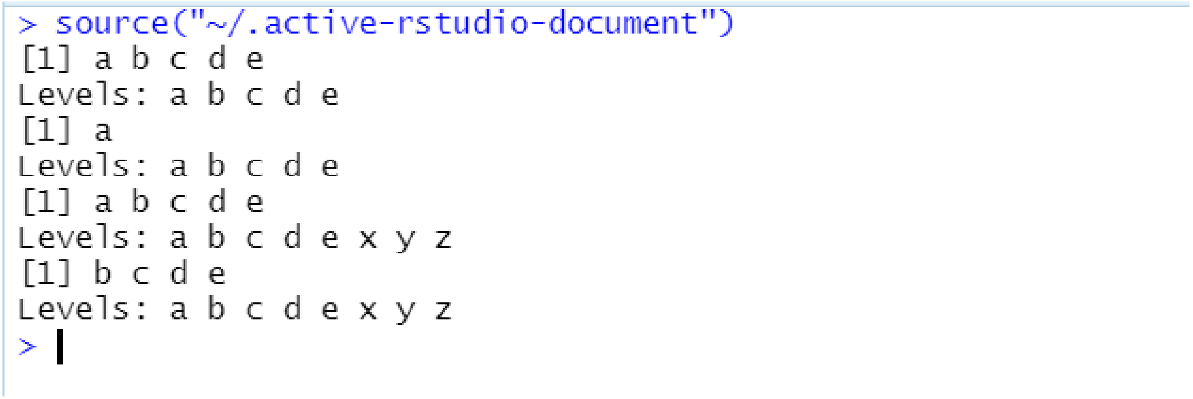
```
# CREATE
df <- data.frame(a = c(1,2,3), b = c(11,12,13))
print(df)

# ACCESS ELEMENT
print(df$a)

# MODIFY ELEMENT
df$a <- 51:53
print(df)

# DELETE ELEMENT
df <- df[-1]
print(df)
```



```
> source("~/.active-rstudio-document")
  a  b
1 1 11
2 2 12
3 3 13
[1] 1 2 3
   a  b
1 51 11
2 52 12
3 53 13
   b
1 11
2 12
3 13
> 
```

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)
```

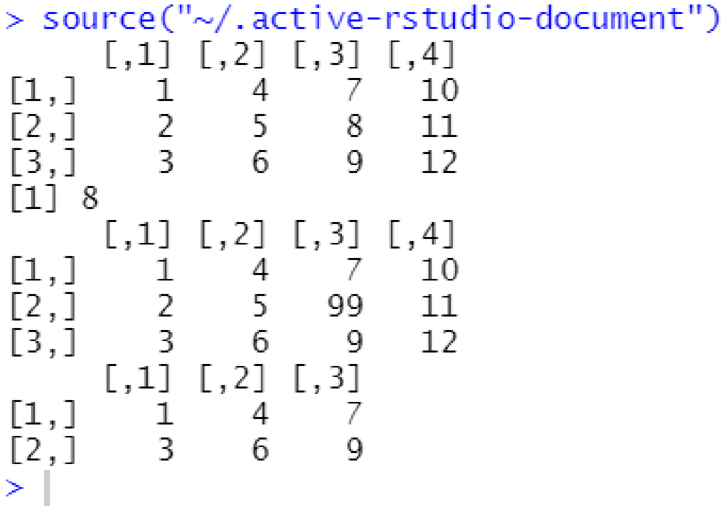


```
> source("~/.active-rstudio-document")
[1] a b c d e
Levels: a b c d e
[1] a
Levels: a b c d e
[1] a b c d e
Levels: a b c d e x y z
[1] b c d e
Levels: a b c d e x y z
> 
```

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)
```



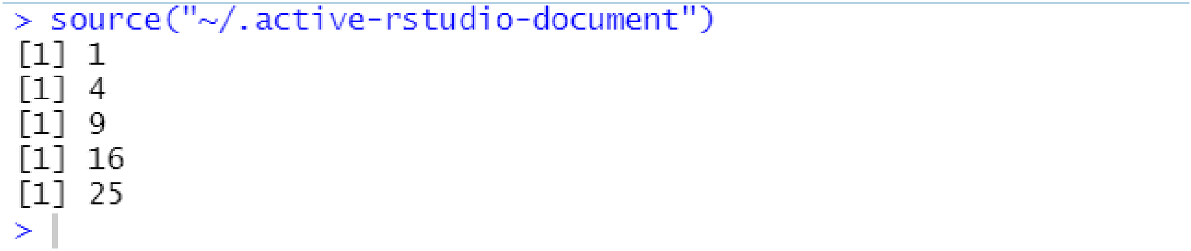
```
> source("~/.active-rstudio-document")
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
[1] 8
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5   99   11
[3,]    3    6    9   12
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    3    6    9
> 
```

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)
```



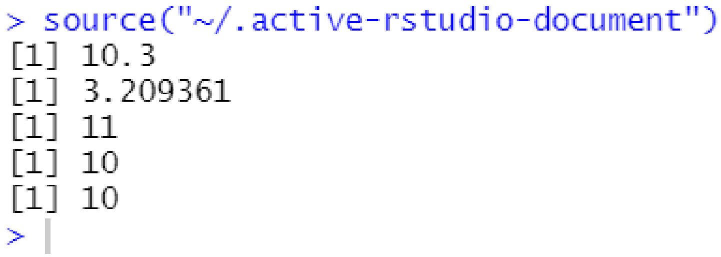
```
> source("~/.active-rstudio-document")
[1] 1
[1] 4
[1] 9
[1] 16
[1] 25
> 
```

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.

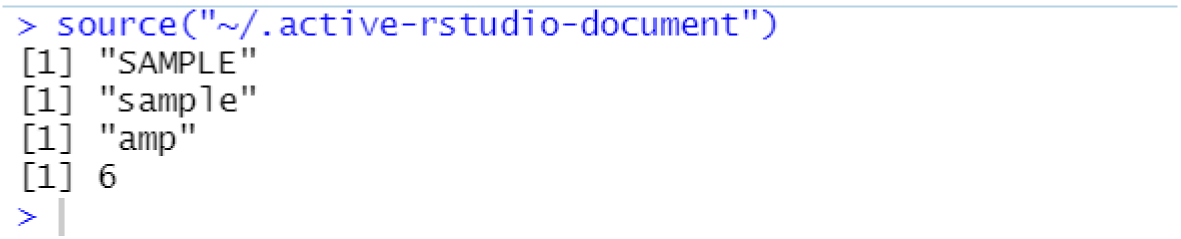
R Script + Output:

```
# Numerical Functions
a <- 10.3

print(abs(a))
print(sqrt(a))
print(ceiling(a))
print(floor(a))
print(round(a))
```



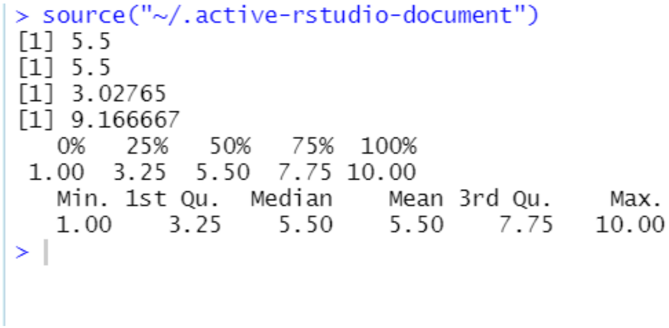
```
> source("~/.active-rstudio-document")
[1] 10.3
[1] 3.209361
[1] 11
[1] 10
[1] 10
> 
```

```
# Character functions

a <- "sample"

print(toupper(a))
print(tolower(a))
print(substr(a, 2, 4))
print(nchar(a))
```



```
> source("~/.active-rstudio-document")
[1] "SAMPLE"
[1] "sample"
[1] "amp"
[1] 6
> 
```

```
# Statistical functions

a <- c(1:10)

print(mean(a))
print(median(a))
print(sd(a))
print(var(a))
print(quantile(a))
print(summary(a))
```



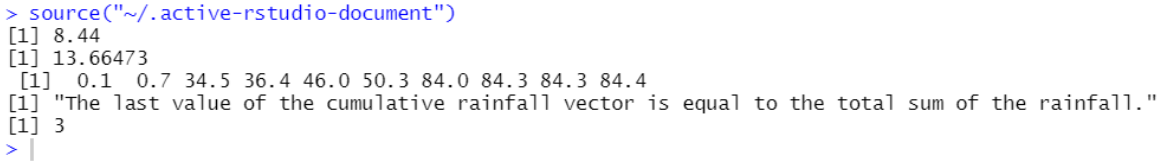
```
> source("~/.active-rstudio-document")
[1] 5.5
[1] 5.5
[1] 3.02765
[1] 9.166667
   0%   25%   50%   75%  100% 
 1.00  3.25  5.50  7.75 10.00 
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   1.00    3.25    5.50    5.50    7.75   10.00 
> 
```

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)
```



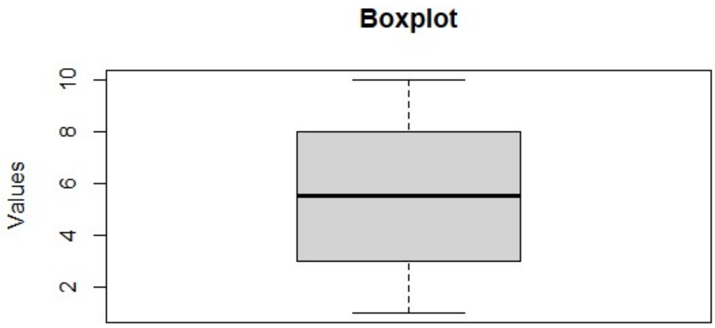
```
> source("~/.active-rstudio-document")
[1] 8.44
[1] 13.66473
 [1]  0.1  0.7 34.5 36.4 46.0 50.3 84.0 84.3 84.3 84.4
[1] "The last value of the cumulative rainfall vector is equal to the total sum of the rainfall."
[1] 3
> 
```

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.

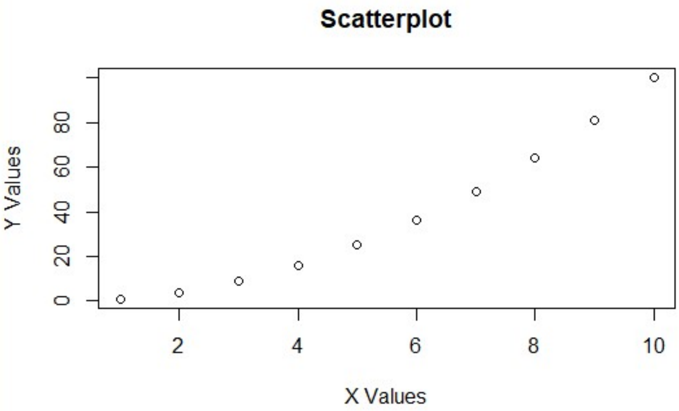
Box Plot:

```
x = sample(1:10)
boxplot(x, main="Boxplot", ylab="Values")
```



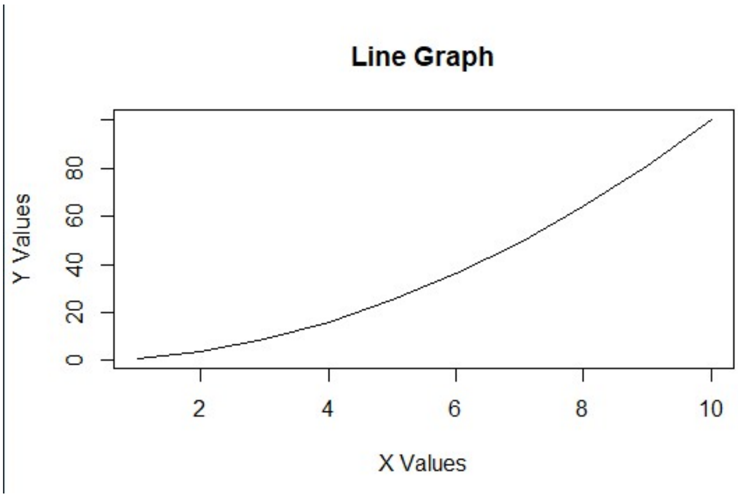
Scatter Plot:

```
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")
```



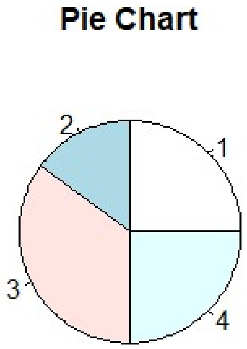
Line Graph:

```
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")
```



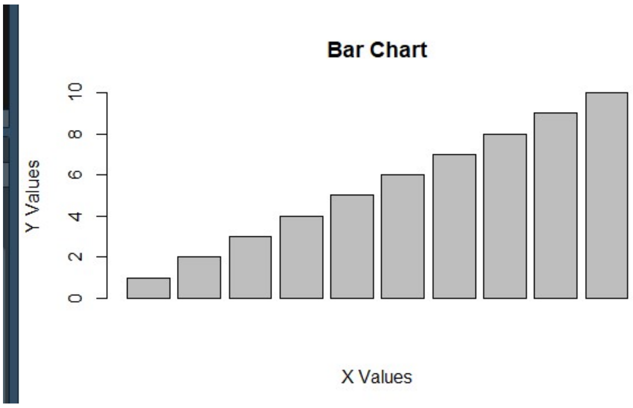
Pie Chart:

```
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
```