Batch: H2_1
Roll No.: 16010122151
Experiment No. 1

Title : Exploring R for Data Science

Aim: To understand basics of R - Operators, built-in functions, Data types, Data manipulation in R, R packages for Data Science

Expected Outcome of Experiment:
CO : Students should write

Books/ Journals/ Websites referred:
1. https://cran.r-project.org/
2. Students should write
3. Students should write

What is R?

- R is a scripting/programming language and environment for statistical computing, data science and graphics.
- R is a successor of the proprietary statistical computing programming language S.
- It is an important tool for computational statistics, visualization and data science.

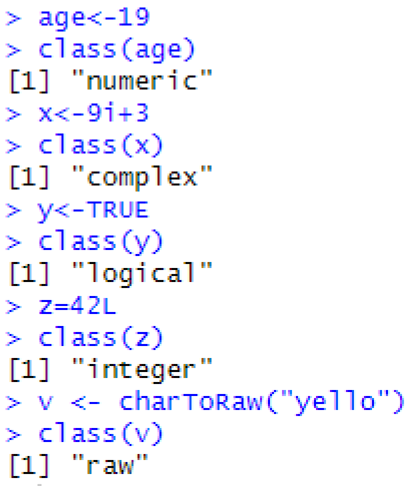
Why R?

It provides techniques for various statistical analyses like classical tests and classification, time-series analysis, clustering, linear and non-linear modelling and graphical operations.

It has superior support for graphics.

Reasons for learning R:
- Free, Open source
- Great visualization
- Cross-platform compatibility
- Advanced statistics
- Integration with other programming languages
- Supportive open source community
- Easy extensibility via packages

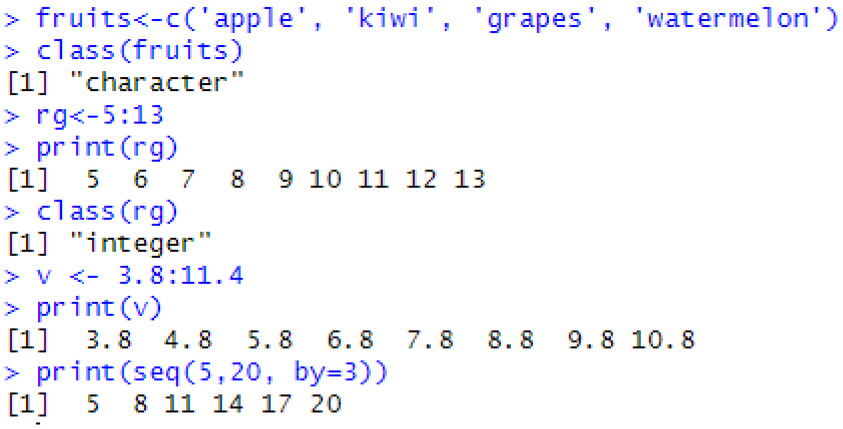
1. **Exploring the atomic datatypes supported by R-Logical, Numeric-integer, Character, Double, Complex, Raw**



```
> age<-19
> class(age)
[1] "numeric"
> x<-9i+3
> class(x)
[1] "complex"
> y<-TRUE
> class(y)
[1] "logical"
> z=42L
> class(z)
[1] "integer"
> v <- charToRaw("yello")
> class(v)
[1] "raw"
```

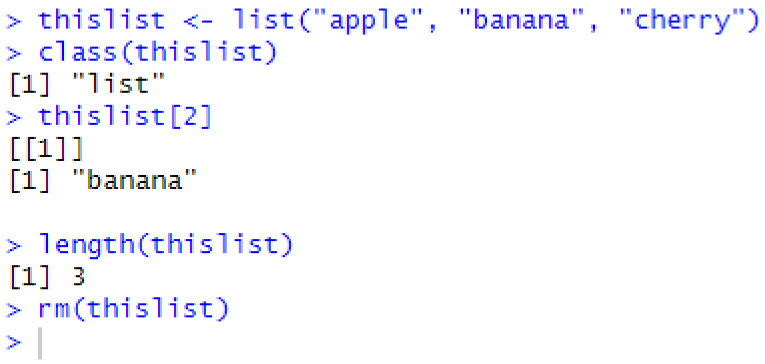
2. **Exploring data manipulation of different data objects of R- Vectors- Matrices, Factors, List, Array, Data Frames**

Vectors:



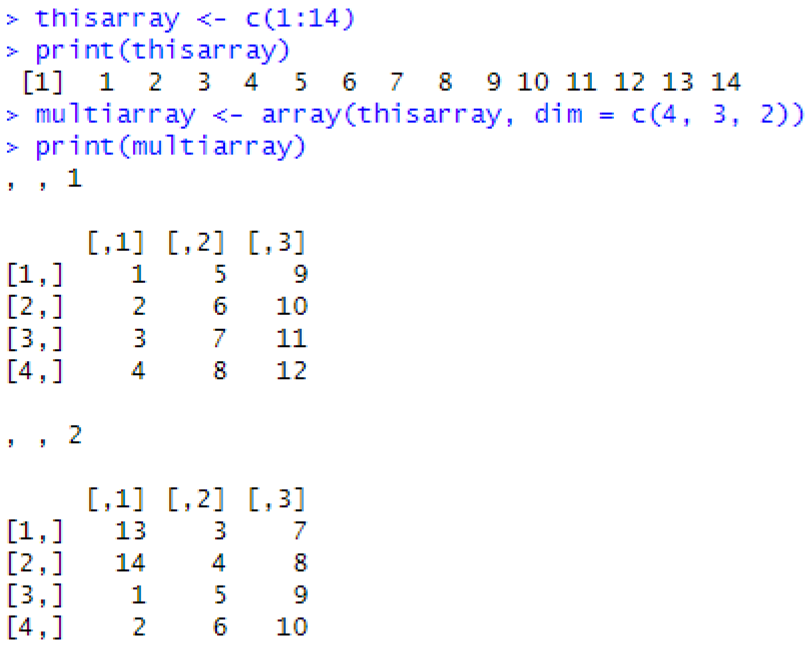
```
> fruits<-c('apple', 'kiwi', 'grapes', 'watermelon')
> class(fruits)
[1] "character"
> rg<-5:13
> print(rg)
[1]  5  6  7  8  9 10 11 12 13
> class(rg)
[1] "integer"
> v <- 3.8:11.4
> print(v)
[1]  3.8  4.8  5.8  6.8  7.8  8.8  9.8 10.8
> print(seq(5,20, by=3))
[1]  5  8 11 14 17 20
```

Lists:



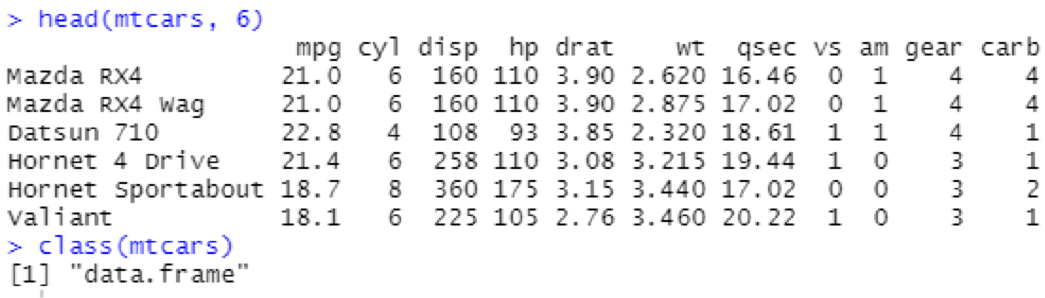
```
> thislist <- list("apple", "banana", "cherry")
> class(thislist)
[1] "list"
> thislist[2]
[[1]]
[1] "banana"

> length(thislist)
[1] 3
> rm(thislist)
> 
```

Array & Multiarray(Matrices):



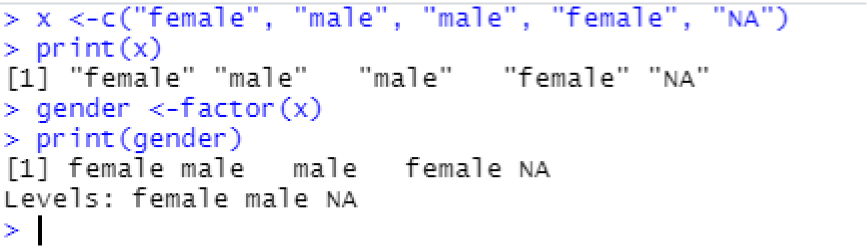
```
> thisarray <- c(1:14)
> print(thisarray)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14
> multiarray <- array(thisarray, dim = c(4, 3, 2))
> print(multiarray)
, , 1

     [,1] [,2] [,3]
[1,]    1    5    9
[2,]    2    6   10
[3,]    3    7   11
[4,]    4    8   12

, , 2

     [,1] [,2] [,3]
[1,]   13    3    7
[2,]   14    4    8
[3,]    1    5    9
[4,]    2    6   10
```

Data Frames:



```
> head(mtcars, 6)
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
> class(mtcars)
[1] "data.frame"
```

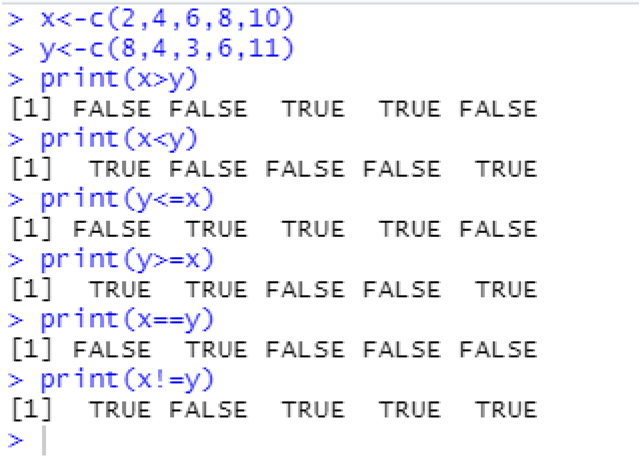
Factors:



```
> x <-c("female", "male", "male", "female", "NA")
> print(x)
[1] "female" "male"   "male"   "female" "NA"
> gender <-factor(x)
> print(gender)
[1] female male   male   female NA
Levels: female male NA
> 
```

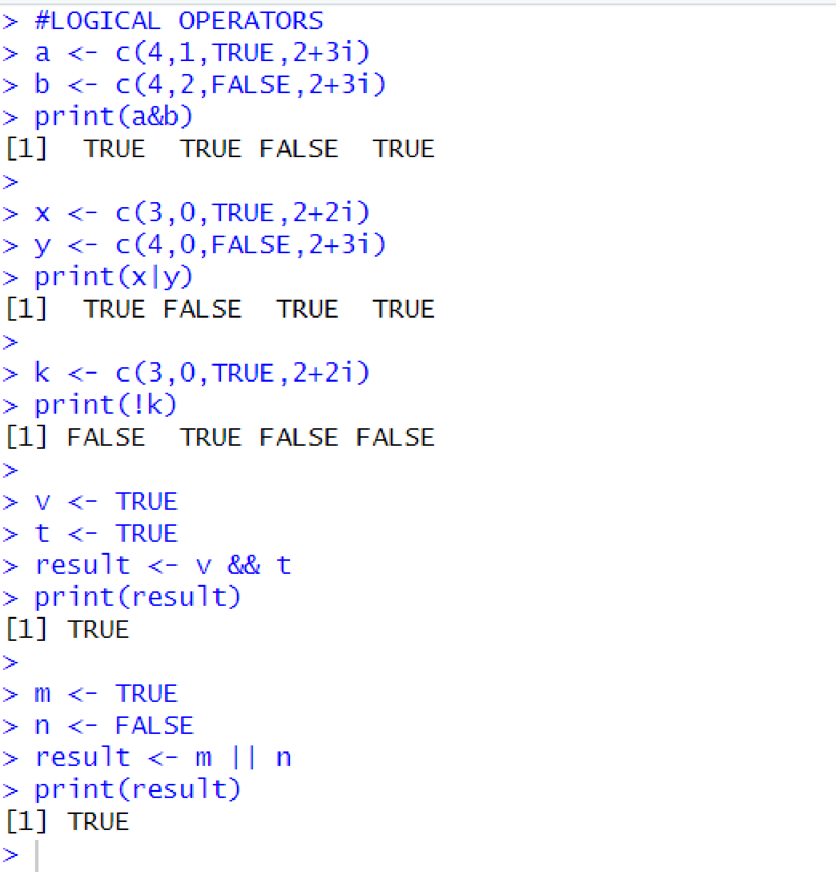
3. Exploring Operators and built-in functions and writing user-defined functions in R

Relational Operators:



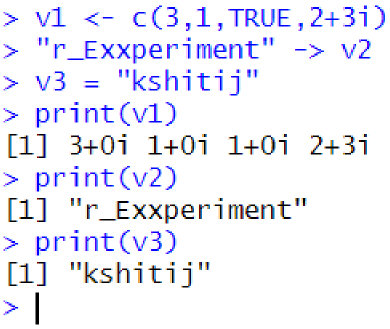
```
> x<-c(2,4,6,8,10)
> y<-c(8,4,3,6,11)
> print(x>y)
[1] FALSE FALSE  TRUE  TRUE FALSE
> print(x<y)
[1]  TRUE FALSE FALSE FALSE  TRUE
> print(y<=x)
[1] FALSE  TRUE  TRUE  TRUE FALSE
> print(y>=x)
[1]  TRUE  TRUE FALSE FALSE  TRUE
> print(x==y)
[1] FALSE  TRUE FALSE FALSE FALSE
> print(x!=y)
[1]  TRUE FALSE  TRUE  TRUE  TRUE
> 
```

Logical Operators:



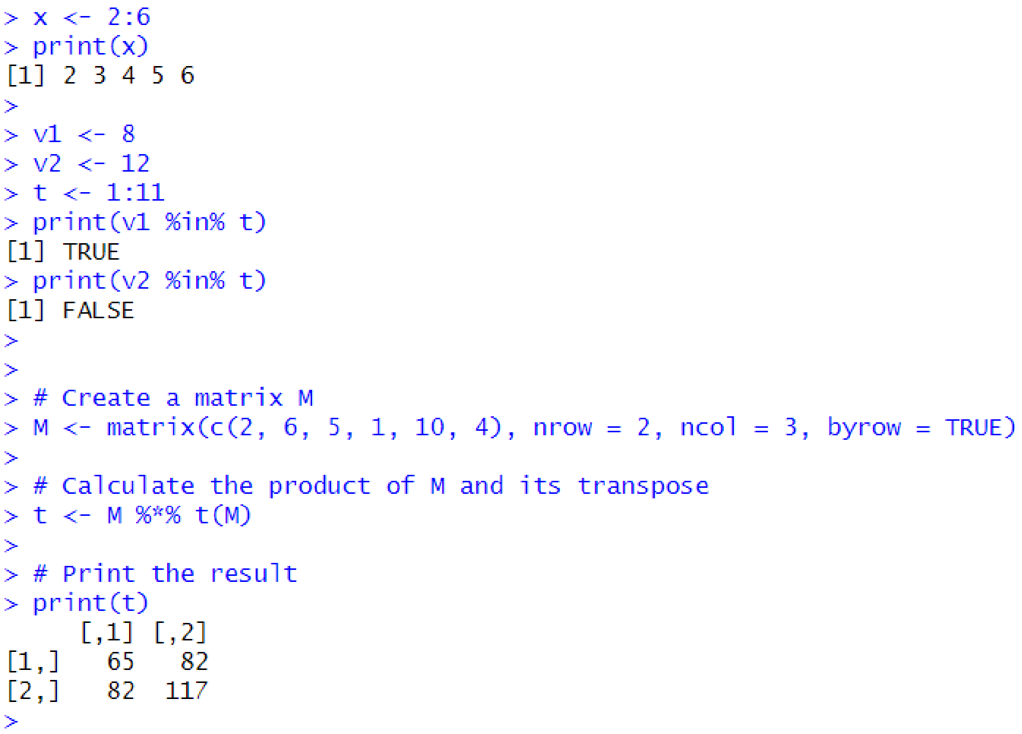
```
> #LOGICAL OPERATORS
> a <- c(4,1,TRUE,2+3i)
> b <- c(4,2,FALSE,2+3i)
> print(a&b)
[1]  TRUE  TRUE FALSE  TRUE
> 
> x <- c(3,0,TRUE,2+2i)
> y <- c(4,0,FALSE,2+3i)
> print(x|y)
[1]  TRUE FALSE  TRUE  TRUE
> 
> k <- c(3,0,TRUE,2+2i)
> print(!k)
[1] FALSE  TRUE FALSE FALSE
> 
> v <- TRUE
> t <- TRUE
> result <- v && t
> print(result)
[1] TRUE
> 
> m <- TRUE
> n <- FALSE
> result <- m || n
> print(result)
[1] TRUE
> 
```

Assignment Operators:



```
> v1 <- c(3,1,TRUE,2+3i)
> "r_Exxperiment" -> v2
> v3 = "kshitij"
> print(v1)
[1] 3+0i 1+0i 1+0i 2+3i
> print(v2)
[1] "r_Exxperiment"
> print(v3)
[1] "kshitij"
> 
```

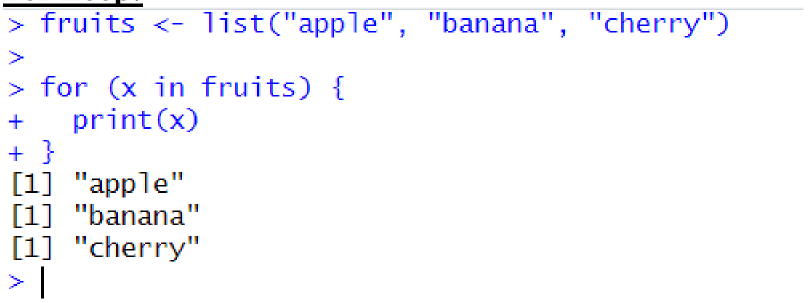
Miscellaneous Operators:



```
> x <- 2:6
> print(x)
[1] 2 3 4 5 6
> 
> v1 <- 8
> v2 <- 12
> t <- 1:11
> print(v1 %in% t)
[1] TRUE
> print(v2 %in% t)
[1] FALSE
> 
> 
> # Create a matrix M
> M <- matrix(c(2, 6, 5, 1, 10, 4), nrow = 2, ncol = 3, byrow = TRUE)
> 
> # Calculate the product of M and its transpose
> t <- M %*% t(M)
> 
> # Print the result
> print(t)
     [,1] [,2]
[1,]   65   82
[2,]   82  117
> 
```

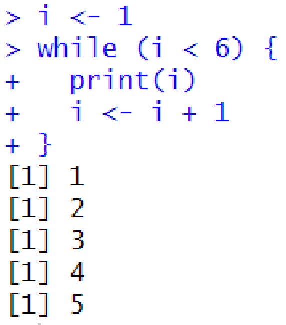
4. Using Looping constructs in R

For Loop:



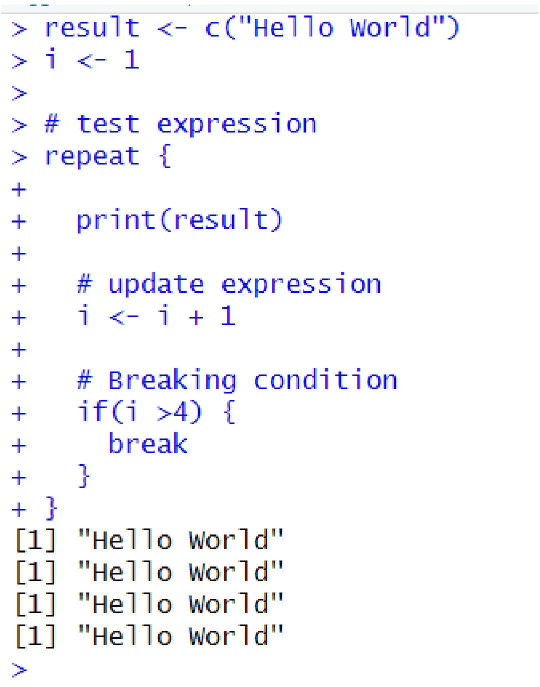
```
> fruits <- list("apple", "banana", "cherry")
> 
> for (x in fruits) {
+   print(x)
+ }
[1] "apple"
[1] "banana"
[1] "cherry"
> 
```

While Loop:



```
> i <- 1
> while (i < 6) {
+   print(i)
+   i <- i + 1
+ }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

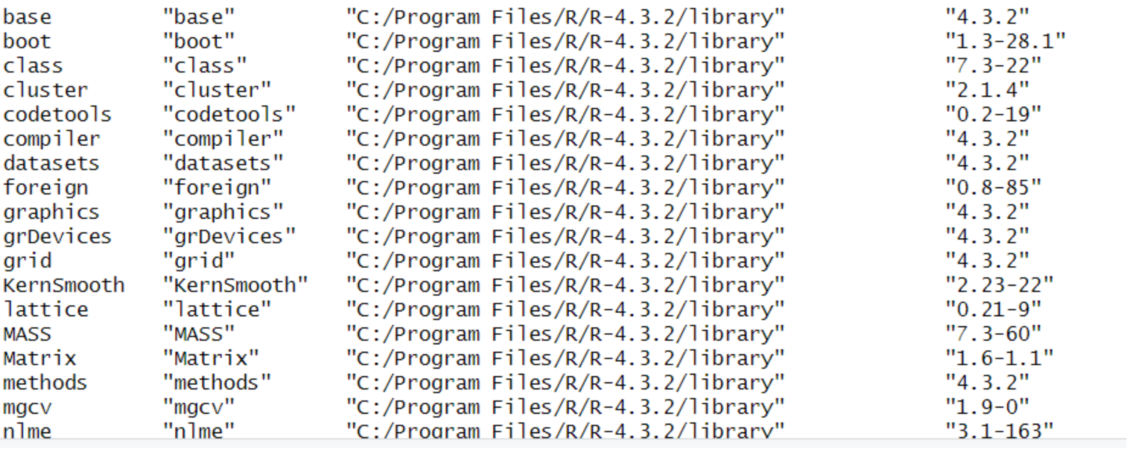
Repeat Loop:



```
> result <- c("Hello World")
> i <- 1
>
> # test expression
> repeat {
+
+   print(result)
+
+   # update expression
+   i <- i + 1
+
+   # Breaking condition
+   if(i >4) {
+     break
+   }
+ }
[1] "Hello World"
[1] "Hello World"
[1] "Hello World"
[1] "Hello World"
>
```

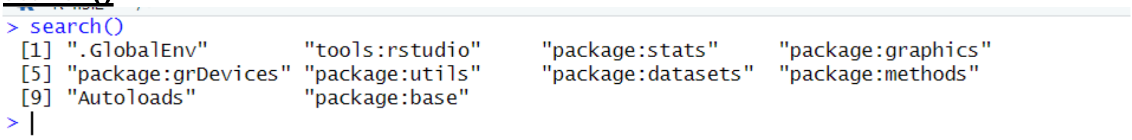
5. Exploring any Packages in R (any graphic package)

installed.packages()



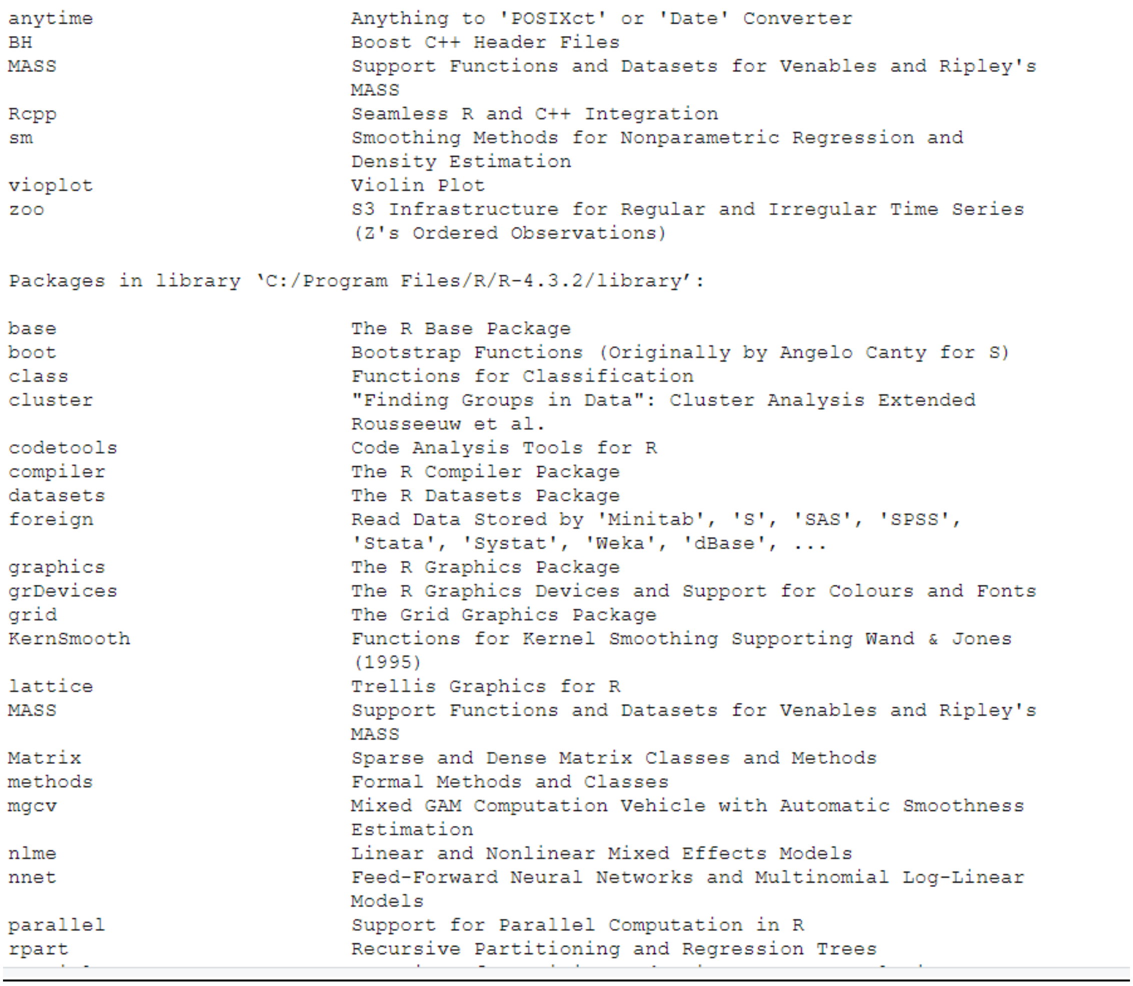
```
base          "base"          "C:/Program Files/R/R-4.3.2/library"             "4.3.2"
boot          "boot"          "C:/Program Files/R/R-4.3.2/library"             "1.3-28.1"
class         "class"         "C:/Program Files/R/R-4.3.2/library"             "7.3-22"
cluster       "cluster"       "C:/Program Files/R/R-4.3.2/library"             "2.1.4"
codetools     "codetools"     "C:/Program Files/R/R-4.3.2/library"             "0.2-19"
compiler      "compiler"      "C:/Program Files/R/R-4.3.2/library"             "4.3.2"
datasets      "datasets"      "C:/Program Files/R/R-4.3.2/library"             "4.3.2"
foreign       "foreign"       "C:/Program Files/R/R-4.3.2/library"             "0.8-85"
graphics      "graphics"      "C:/Program Files/R/R-4.3.2/library"             "4.3.2"
grDevices     "grDevices"     "C:/Program Files/R/R-4.3.2/library"             "4.3.2"
grid          "grid"          "C:/Program Files/R/R-4.3.2/library"             "4.3.2"
KernSmooth    "KernSmooth"    "C:/Program Files/R/R-4.3.2/library"             "2.23-22"
lattice       "lattice"       "C:/Program Files/R/R-4.3.2/library"             "0.21-9"
MASS          "MASS"          "C:/Program Files/R/R-4.3.2/library"             "7.3-60"
Matrix        "Matrix"        "C:/Program Files/R/R-4.3.2/library"             "1.6-1.1"
methods       "methods"       "C:/Program Files/R/R-4.3.2/library"             "4.3.2"
mgcv          "mgcv"          "C:/Program Files/R/R-4.3.2/library"             "1.9-0"
nlme          "nlme"          "C:/Program Files/R/R-4.3.2/library"             "3.1-163"
```

search()



```
> search()
 [1] ".GlobalEnv"        "tools:rstudio"     "package:stats"     "package:graphics"
 [5] "package:grDevices" "package:utils"     "package:datasets"  "package:methods"
 [9] "Autoloads"         "package:base"
> |
```

library()



```
anytime                    Anything to 'POSIXct' or 'Date' Converter
BH                         Boost C++ Header Files
MASS                       Support Functions and Datasets for Venables and Ripley's
                           MASS
Rcpp                       Seamless R and C++ Integration
sm                         Smoothing Methods for Nonparametric Regression and
                           Density Estimation
vioplot                    Violin Plot
zoo                        S3 Infrastructure for Regular and Irregular Time Series
                           (Z's Ordered Observations)

Packages in library 'C:/Program Files/R/R-4.3.2/library':

base                       The R Base Package
boot                       Bootstrap Functions (Originally by Angelo Canty for S)
class                      Functions for Classification
cluster                    "Finding Groups in Data": Cluster Analysis Extended
                           Rousseeuw et al.
codetools                  Code Analysis Tools for R
compiler                   The R Compiler Package
datasets                   The R Datasets Package
foreign                    Read Data Stored by 'Minitab', 'S', 'SAS', 'SPSS',
                           'Stata', 'Systat', 'Weka', 'dBase', ...
graphics                   The R Graphics Package
grDevices                  The R Graphics Devices and Support for Colours and Fonts
grid                       The Grid Graphics Package
KernSmooth                 Functions for Kernel Smoothing Supporting Wand & Jones
                           (1995)
lattice                    Trellis Graphics for R
MASS                       Support Functions and Datasets for Venables and Ripley's
                           MASS
Matrix                     Sparse and Dense Matrix Classes and Methods
methods                    Formal Methods and Classes
mgcv                       Mixed GAM Computation Vehicle with Automatic Smoothness
                           Estimation
nlme                       Linear and Nonlinear Mixed Effects Models
nnet                       Feed-Forward Neural Networks and Multinomial Log-Linear
                           Models
parallel                   Support for Parallel Computation in R
rpart                      Recursive Partitioning and Regression Trees
```

Conclusion:

By this experiment we understood various data frames and the application of vectors and matrices in R programming we also got and idea about the lists and various data types and various functions.