

```

# R is case Sensitive Language

# ??arules (search for the help in additional packages)
# ?mean (search for the help in base R docs)

#How to assign a value in a variable

Var1<-25
Var1
Var1=10
Var1

# Can use gets operator(<-) or equal to operator(=) for assignment of variable
# No need of declaring a variable like in programming language
# Single = is for assignment and double = is to parse a statement
#Variable values can be changed dynamically

sessionInfo()

#### Datastructures in R Programming Language

# 1. Vector
# 2. Matrix
# 3. Arrays
# 4. List
# 5. Dataframes

### 1. Vector

# One dimension
# Same type of elements

Vec1 <- 1:10
Vec1
class(Vec1)
typeof(Vec1)

#Colon operator will give sequence of numbers
# class = "integer" and Type = " integer"
# Type represent the data type i.e character , integer , numeric , logical ,
boolean etc...

Vec2 <- c("POCO","RHEA","RAM","RAJ")
Vec2
class(Vec2)
typeof(Vec2)

# class = "character" and Type = " character"

```

```

Vec3 <- c(101,95,10,3,85)
Vec3
class(Vec3)

# class = "numeric"
# When we specify with colon operator(:) it is definitely a whole number
#when we specify with "c" oprator we can put decimal value as well
#Hence, class of Vec 1 is "integer" and class of Vec3 is "numeric"
# integer = whole numbers & numeric = decimal

Vec4 <- c("101","95","10","3","85")
Vec4
class(Vec4)

# class = character
# When we specify numbers inside double codes its class will always be Character

Vec6 <- c("Raj","Shyam",1,2,10,21,35)
class(Vec6)

# class = character
# In the mix class type of data (character and interger) , class of Vector will
always be character

Vec7 <- c(Raj,Shyam,1,2,10,21,35)
class(Vec7)

# Character values should always be specified in "Double codes" else it will give
error

# To access data from vector we have to use Square bracket ( [] )
# Index start from 1 and goes on like 1,2,3,.....

## I want to access 1st element of Vec6

Vec6[1]

# will give output as "Raj"

## I want to access 2nd element of Vec6

Vec6[2]

# will give output as "Shyam"

## I want to access first 3 elements of Vec6

Vec6[1:3]

```

```
Vec6[c(1,2,3)]
```

```
## I want to access 1st , 2nd and 7th element of Vec6
```

```
Vec6[c(1,2,7)]
```

```
## I want to access 1st to 4th and 7th element of Vec6
```

```
Vec6[c(1:4,7)]
```

```
Vec6[c(1,2,3,4,7)]
```

```
Vec5 <- c(TRUE,FALSE,FALSE,TRUE,T,F)
```

```
class(Vec5)
```

```
# class = logical
```

```
Vec8 <- c("TRUE","FALSE","FALSE","TRUE","T","F")
```

```
class(Vec8)
```

```
# If we specify logical values inside double codes its class will always be  
character
```

```
# Any value inside double codes will be treated as "character value"
```

```
Vec9 <- c("TRUE","FALSE",FALSE,TRUE,"T","F")
```

```
class(Vec9)
```

```
# In the mix class type of data (character and logical) , class of Vector will  
always be character
```

```
Vec10 <- c(TRUE,FALSE,1,2)
```

```
class(Vec10)
```

```
# In the mix class type of data (numeric and logical) , class of Vector will always  
be numeric
```

```
## How to find out no of elements in a vec ??
```

```
length(Vec6)
```

```
# length is 7 that means 7 elements are present in Vec6
```

```
### 2. Matrix
```

```

# Two dimension
# Same type of elements

Mat1 <- matrix(1:20,4,5)
Mat1
class(Mat1)
typeof(Mat1)

# Syntax
# 1: 20 represent the "data"
# 4 represent "no of rows"
# 5 represent "no of columns"
# class = matrix and type = integer

## How to access data from a Matrix

# Square bracket is used to access the data
# MATRIXNAME[ROW,column] => matrixname [row no,column no]

Mat1[2,]
class(Mat1[2,])
# gives 2nd row from Matrix
# Each row of a matrix is a vector

Mat1[,3]
class(Mat1[,3])
# gives 3rd column from Matrix

Mat1[2,3]
class(Mat1[2,3])

# gives the element from 2nd row of 3rd column

## How to change a value in a Matrix ??

Mat1[2,3] <- "ketan"
Mat1
class(Mat1)
typeof(Mat1)

# class = matrix and type = character
# Type of matrix is changed to "character" as it can store elements of 1 type only

## How to access col 2,4 and 5 of Matrix 1

Mat1[,c(2,4,5)]
class(Mat1[,c(2,4,5)])

```

```
## How to access row 1,2 and 4 of Matrix 1
```

```
Mat1[c(1,2,4),]  
class(Mat1[c(1,2,4),])
```

```
# Whenever we access a multiple rows or columns from a Matrix its class will always  
be matrix
```

### ### 3. Array

```
# More than one dimension  
# Elements of same type  
# Array is collection of matrices
```

```
Array1 <- array(1:100,dim = c(3,4,3))  
Array1  
class(Array1)  
typeof(Array1)
```

```
Array2 <- array(1:100,c(3,4,3))  
Array2  
class(Array2)  
typeof(Array2)
```

```
# Array1 and Array2 represent same data ( "dim=" is an optional command )  
# class = array and type = interger  
# "dim=" => to set the dimensions  
# first 3 represent => no of rows  
# 4 reprsrent => no of columns  
# second 3 represent => no of matrices is an array
```

```
## How to access data from an Array ??
```

```
#ARRAYNAME [ROW,COLUMN,MATRIX]
```

```
Array1[, ,2]  
class(Array1[, ,2])  
# to access 2nd Matrix from an array  
#class = matrix
```

```
Array1[3, ,]  
class(Array1[3, ,])  
# to access 3rd row of each matrix  
# class = matrix  
# each row can be seen as column in new matrix
```

```
Array1[,2,]  
class(Array1[,2,])  
# to access 2nd column of each matrix  
# class = matrix
```

```
dim(Array1)  
dim(Array1[,2,])  
dim(Array1[3,,])  
dim(Array1[,2,])
```

```
Array3 <- array(1:100,dim = c(5,6,3))  
Array3  
dim(Array3)
```

```
Array3[3,,]  
dim(Array3[3,,])
```

```
Array3[,2,]  
dim(Array3[,2,])
```

```
# Whenever we extract only a particular row from an array its dimension will be  
(column nos,matrix nos)  
# Whenever we extract only a particular column from an array its dimension will be  
(row nos,matrix nos)
```

```
Array3[,2,3]  
class(Array3[,2,3])  
# it will be a vector
```

```
Array3[2,,3]  
class(Array3[2,,3])  
# it will be a vector
```

```
Array3[2,3,]  
class(Array3[2,3,])  
# it will be a vector
```

```
Array3[2,3,1]  
class(Array3[2,3,1])  
# gives the element present at 2nd row & 3rd col of matrix 1 from an array
```

```
## How to change a value in an array ??
```

```
Array1  
Array1[1,2,3] <- "Nikhil"  
Array1
```

```
class(Array1)
typeof(Array1)

# class = Array
# type = character

# datatypes ( int , num , char , logical , complex )
# datastructures ( vector , matrix , array , list )
```

#### ### 4. List

```
# List can have many datatypes together
# list can have many datastructures together
# slightly complicated but very powerful
```

##### ## List of vectors

```
List_of_vectors <- list(Vec1,Vec2,Vec3,Vec5)
List_of_vectors
class(List_of_vectors)
```

```
# class = list
```

```
## How many items are in the list ??
```

```
length(List_of_vectors)
```

##### ## Access items from the list

```
List_of_vectors[[1]]
class(List_of_vectors[[1]])
```

```
# gives 1st vector(datastructure) from the list
```

```
List_of_vectors[[4]]
class(List_of_vectors[[4]])
```

```
# gives 4th vector(datastructure) from the list
```

```
List_of_vectors[[2]][3]
class(List_of_vectors[[2]][3])
```

```
# gives 3rd element of 2nd vector in the list
```

```
## List of Vector and Matrix
```

```
List_of_VecMat <- list(Vec1,Vec2,Vec3,Mat1)
List_of_VecMat
class(List_of_VecMat)
```

```
# class = list
```

```
List_of_VecMat[[4]]
class(List_of_VecMat[[4]])
```

```
List_of_VecMat[[4]][2,3]
class(List_of_VecMat[[4]][2,3])
```

```
List_of_VecMat[[4]][2,]
class(List_of_VecMat[[4]][2,])
```

```
List_of_VecMat[[4]][,3]
class(List_of_VecMat[[4]][,3])
```

```
class(List_of_VecMat[[1]])
List_of_VecMat[[1]][1]
```

```
# Double Bracket is used for accessing element of the list
# Each element of the list is a datastructure
# Single Bracket "["]" is used for accessing element of a datastructure
# Always check for the class of the datastructure of the list while accessing its
element
# If is a vector then Syntax : [number]
# If is a matrix then Syntax : [row no,col no]
# If is an array then Syntax : [row no,col no,matrix no]
```

```
## List of Vector , Matrix and List
```

```
List_of_VecMatList <- list(Vec1,Mat1,List_of_VecMat)
List_of_VecMatList
class(List_of_VecMatList)
length(List_of_VecMatList)
```

```
## I want to access 3rd datastructurs of List_of_VecMatList
```

```
List_of_VecMatList[[3]]
class(List_of_VecMatList[[3]])
```

```
# class = list
# 3rd datastructure of the list "List_of_VecMatList" is also a "list"
```



```
## I want to access 2nd element form the 3rd datastructure of "List_of_VecMatList"
```

```
List_of_VecMatList[[3]][[2]]  
class(List_of_VecMatList[[3]][[2]])
```

```
# it is a vector
```

```
## Now , i want to access 1st element of this vector
```

```
List_of_VecMatList[[3]][[2]][1]
```

```
# Double Bracket is used for accessing element of the list  
# Each element of the list is a datastructure  
# Single Bracket "[" is used for accessing element of a datastructure  
# Always check for the class of the datastructure of the list while accessing its  
element  
# If is a vector then to access it Syntax : [number]  
# If is a matrix then to access it Syntax : [row no,col no]  
# If is an array then to access it Syntax : [row no,col no,matrix no]  
# If it is a list then we will use Double Brcket to access it  
# Further to access the datastructures from that list we will use the syntax of  
vector,matrix or array as per the class of datastructure
```

```
## List of Vector , Matrix and Array
```

```
List_of_VecMatArray <- list(Vec1,Mat1,Array1)  
List_of_VecMatArray
```

```
class(List_of_VecMatArray)  
# class = list
```

```
class(List_of_VecMatArray[[1]])  
# class = interger (means it is a vector)
```

```
List_of_VecMatArray[[1]][1]  
# i am accessing 1st element of vector
```

```
class(List_of_VecMatArray[[2]])  
# class = matrix
```

```
List_of_VecMatArray[[2]][1,2]  
# i am accessing element from a matrix
```

```
class(List_of_VecMatArray[[3]])  
# class = array
```

```
dim(List_of_VecMatArray[[3]])
List_of_VecMatArray[[3]][1,4,3]
# i am accessing element from an array
```

```
## List of Vector , Matrix , Array and List
```

```
List_of_VecMatArrayList <- list(Vec1,Mat1,Array1,List_of_VecMatList)
List_of_VecMatArrayList
class(List_of_VecMatArrayList)
```

```
# class = list
```

```
class(List_of_VecMatArrayList[[1]])
# i am checking class of 1st datastructure of list "List_of_VecMatArrayList"
# class = interger ( means it is a vector)
```

```
List_of_VecMatArrayList[[1]][2]
# i am accessing element of a vector
```

```
class(List_of_VecMatArrayList[[2]])
# i am checking class of 2nd datastructure of list "List_of_VecMatArrayList"
# class = matrix
```

```
dim(List_of_VecMatArrayList[[2]])
# i am checking the dimension of a matrix
```

```
List_of_VecMatArrayList[[2]][3,2]
# i am accessing data from a matrix
```

```
class(List_of_VecMatArrayList[[3]])
# i am checking class of 3rd datastructure of list "List_of_VecMatArrayList"
# class = array
```

```
dim(List_of_VecMatArrayList[[3]])
# i am checking the dimension of an array
```

```
List_of_VecMatArrayList[[3]][2,4,3]
# i am accessing data from an array
```

```
class(List_of_VecMatArrayList[[4]])
# i am checking class of 4th datastructure of list "List_of_VecMatArrayList"
# class = list
# means further it will have datastructures
```

```

class(List_of_VecMatArrayList[[4]][[1]])
# class = integer (means it is a vector)

length(List_of_VecMatArrayList[[4]][[1]])
# if i am trying to find dimension of a vector then it will give output as "NULL"
# However , we can find length of vector

List_of_VecMatArrayList[[4]][[1]][7]

class(List_of_VecMatArrayList[[4]][[2]])
# class = matrix

dim(List_of_VecMatArrayList[[4]][[2]])
# i am checking the dimension of matrix

List_of_VecMatArrayList[[4]][[2]][3,4]
# i am accessing element from a matrix

class(List_of_VecMatArrayList[[4]][[3]])
# class = list
# means further it will have datastructures

length(List_of_VecMatArrayList[[4]][[3]])

dim(List_of_VecMatArrayList[[4]][[3]])

class(List_of_VecMatArrayList[[4]][[3]][[1]])
# class = integer ( means it is a vector)

length(List_of_VecMatArrayList[[4]][[3]][[1]])

List_of_VecMatArrayList[[4]][[3]][[1]][10]
# i am accessing element from a vector

class(List_of_VecMatArrayList[[4]][[3]][[2]])
# class = character ( means it is a vector)

length(List_of_VecMatArrayList[[4]][[3]][[2]])

List_of_VecMatArrayList[[4]][[3]][[2]][3]
# i am accessing element from a vector

class(List_of_VecMatArrayList[[4]][[3]][[3]])
# class = numeric ( means it is a vector)

length(List_of_VecMatArrayList[[4]][[3]][[3]])

List_of_VecMatArrayList[[4]][[3]][[3]][5]

```

```
# i am accessing element from a vector
```

```
class(List_of_VecMatArrayList[[4]][[3]][[4]])
```

```
# class = matrix
```

```
dim(List_of_VecMatArrayList[[4]][[3]][[4]])
```

```
# since it is a matrix , iam checking the dimension instead of length
```

```
List_of_VecMatArrayList[[4]][[3]][[4]][4,2]
```

```
# i am accessing element from a matrix
```

### ### 5. Dataframe

```
# The most used datastructure in data science
```

```
# It is a spreadsheet
```

```
# Can have many datatypes
```

```
View(iris)
```

```
# iris is a dataset that comes with base R install ( for learning purpose)
```

```
class(iris)
```

```
# class = data.frame
```

```
View(mtcars)
```

```
class(mtcars)
```

```
# class = data.frame
```

```
dim(iris)
```

```
# 150 5 : 150 rows and 5 columns
```

```
dim(mtcars)
```

```
# 32 11 : 32 rows and 11 columns
```

### ## How to save the workspace

```
# .R File
```

```
# can be saved from file menu. It will save only the code in source panel not the data
```

```
# .Rdata file
# To save the workspace
# Go to session menu and click on "save workspace as"
# It will save the data ( datasets that we have created)
```

```
## OPen the saved workspace
```

```
# Go to session menu and click on load workspace
```

```
# Code and workspace needs to be saved differently
```

```
save.image("location/filename.Rdata")
```

```
# It will also save the workspace
# Need to mention ".Rdata" extension in the syntax
```

```
load("workspace name")
```

```
# to load the workspace
```