Exp. 2: Demonstrate Vector operations in R

Aim: To experiment with vectors, the most basic data type / objects of R.

What is meant by a Data Type?

- A variable can be used for storing various kinds of data like character, integer, floating point, double floating point, Boolean etc.

Exp. 2: Demonstrate Vector operations in R

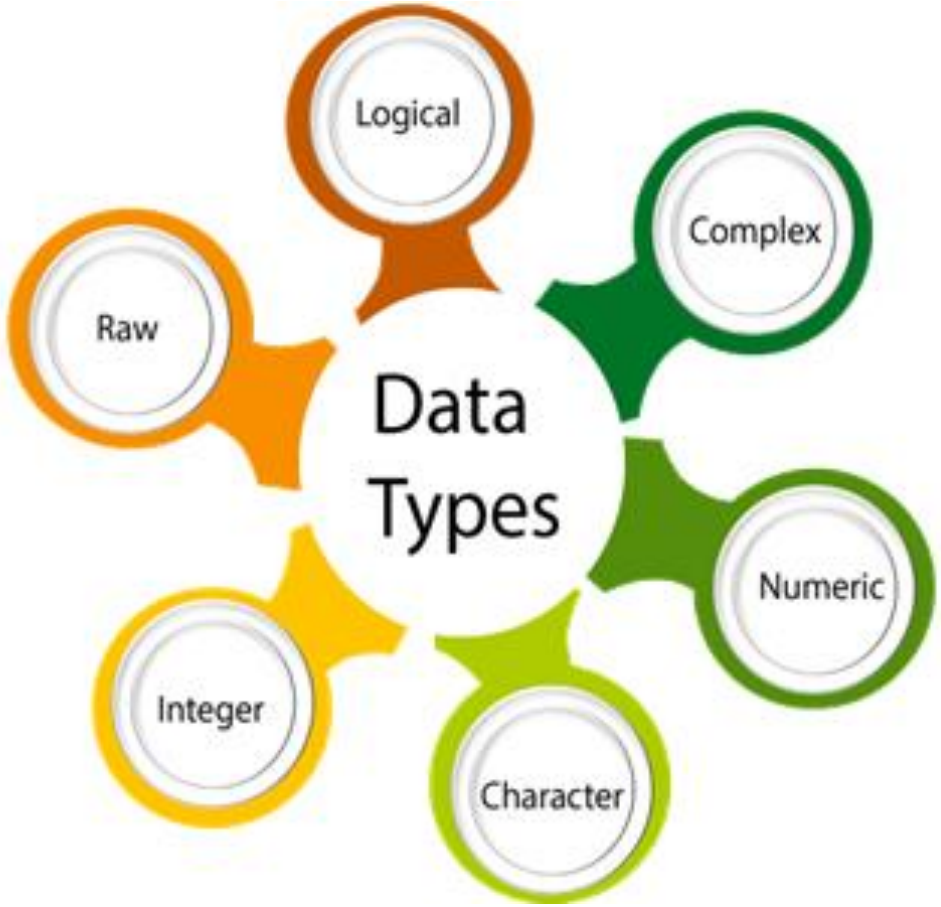
- Based on the data type of a variable, the operating system allocates memory and decides what can be stored in the reserved memory.
- In contrast to other programming languages like C and Java, in R, the variables are not declared as some data type.
- The variables are assigned with R Objects and the data type of the R object becomes the data type of the variable.

Exp. 2: Demonstrate Vector operations in R

- R is called a dynamically typed language, which means that we can change a variable's data type of the same variable again and again when using it in a program.
- For eg.

```
>a=4
>a=TRUE
```

Exp. 2: Demonstrate Vector operations in R



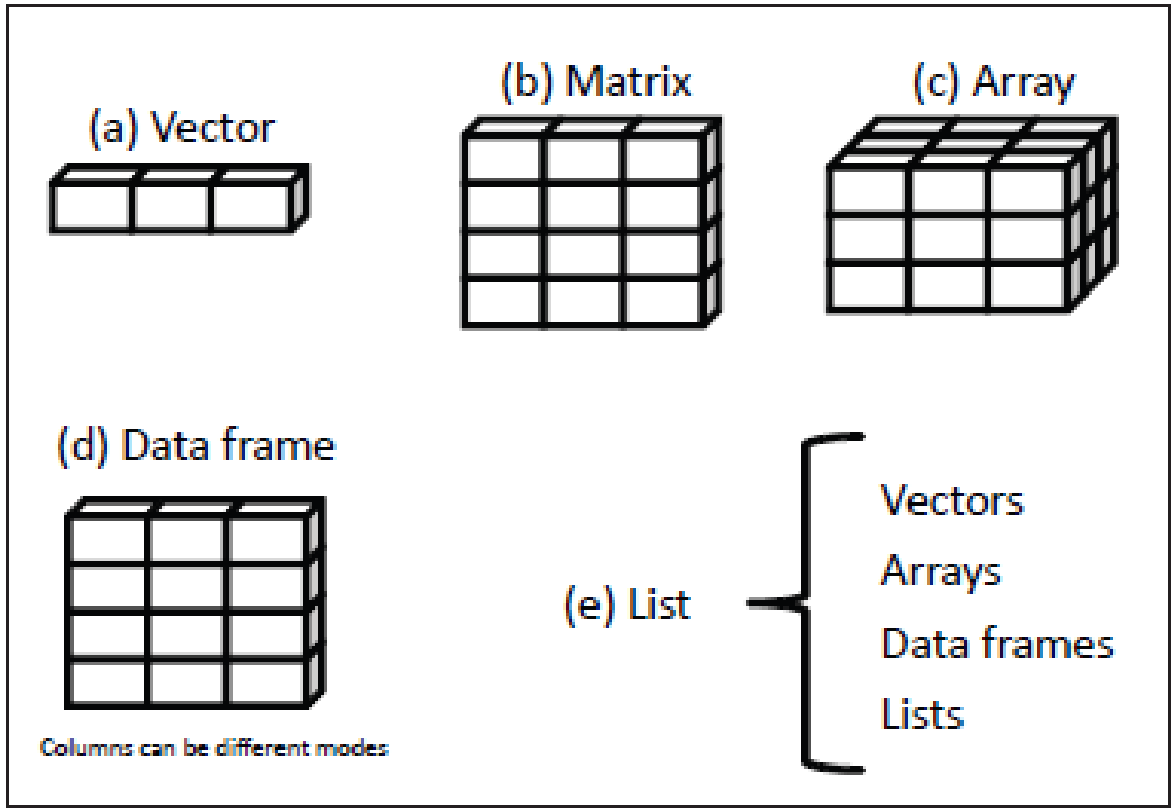
R Data Types

Exp. 2: Demonstrate Vector operations in R

- There are many types of R-objects. The basic data types are:
  - **Logical** : TRUE, FALSE
  - **Numeric** : 2, 5, 10.6
  - **Integer** : 2L, 5L, 10L
  - **Complex** : 2+5i, 3+4i
  - **Character** : "a", "u", "all", "SVEC", "TRUE", ‘10.6’, ‘AI’
  - **Raw** : v <- charToRaw("Hello")
    v is stored as 48 65 6c 6c 6f
  - **Date** : d<-as.Date("2022-1-4")

Exp. 2: Demonstrate Vector operations in R

- The advanced data types or data structures in R are:
  - Vectors
  - Matrices
  - Arrays
  - Data Frames
  - Lists

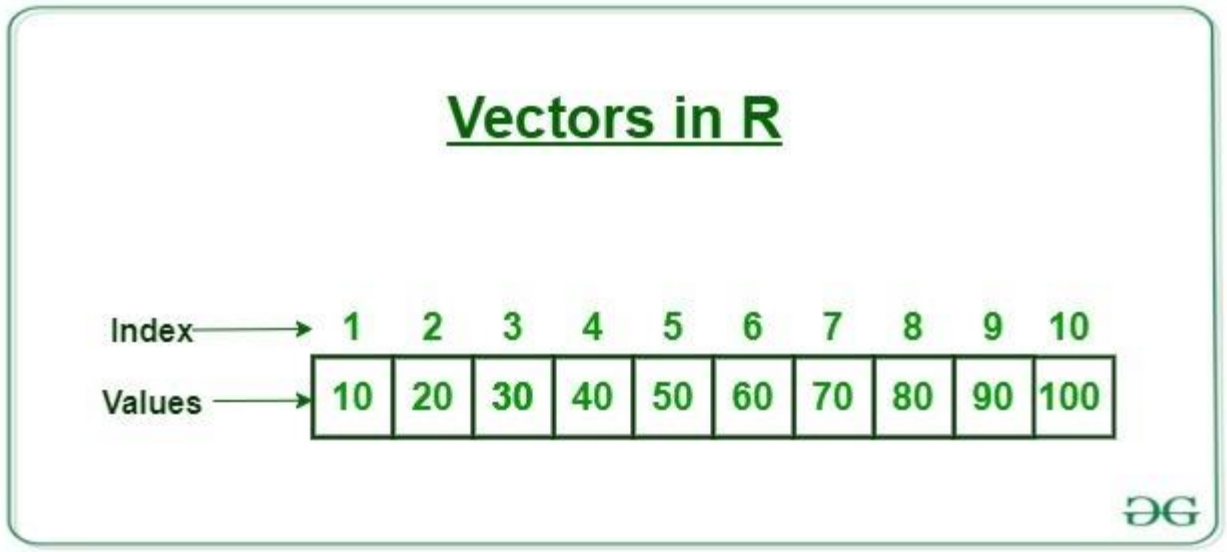


Exp. 2: Demonstrate Vector operations in R

Vectors

- The most basic R data objects
- Even when you write just one value in R, it becomes a vector of length 1.
- Vectors in R are used to hold multiple data values of the same type.

Exp. 2: Demonstrate Vector operations in R



Exp. 2: Demonstrate Vector operations in R

- Creating a vector:
  - There are different ways of creating vectors. Generally, we use ‘c’ to combine different elements together.
  - Ex:
    - X <- c(61, 4, 21, 67, 89, 2)
    - Y=c("S", "V", "E", ‘C’)
    - Z=c("SVEC", "AI")

Exp. 2: Demonstrate Vector operations in R

- We can create a numerical vector using colon operator.
- Eg:
  - v1 <- 1:15
  - v2 <- 6.6:12.6
  - v3 <- 11.4:3.8 ##
- Note: A colon operator always generate a sequence with step size 1 or -1.

Exp. 2: Demonstrate Vector operations in R

- We can create a numerical sequence vector with a specific step size using seq().
- Ex:
  - v3 <- seq(5, 9)
  - v4 <- seq(5, 9, 0.4)
  - v5<- seq(5, 9, by=0.4)

Exp. 2: Demonstrate Vector operations in R

- What happens if we try to create a vector of different data types?
- Ex:
  - v6 <- c('apple','red',5,TRUE)
  - v7 <- c(5,TRUE)
  - v8 <- c(2.6,5.7,87L,9)

Exp. 2: Demonstrate Vector operations in R

- Accessing a Vector: Elements of a Vector are accessed using indexing. The [ ] brackets are used for indexing. Indexing starts with position 1.
- Ex:
  - X <- c(61, 4, 21, 67, 89, 2)
  - X[1] #61
  - X[4] #67
  - X[c(1,3)] #61 21
  - X[c(3,1,4)] # 21 61 67

Exp. 2: Demonstrate Vector operations in R

- Giving a negative value in the index drops that element from result.
- Ex:
  - X[-2] # 61 21 67 89 2

Exp. 2: Demonstrate Vector operations in R

- TRUE, FALSE can be used for indexing.
- Ex:
  - X[c(TRUE,TRUE,FALSE,FALSE,TRUE,FALSE)] #61 4 89

Exp. 2: Demonstrate Vector operations in R

Manipulating a Vector

- An element of a vector can be altered using the indexing.
- Ex:
  - X <- c(61, 4, 21, 67, 89, 2)
  - X[2]=10
  - X[c(1,3)]=c(25,36)
  - X[1:2]=c(2,3)

Exp. 2: Demonstrate Vector operations in R

- Two vectors of same length can be added, subtracted, multiplied or divided giving the result as a vector output.
- Eg:
  - V1 <- c(61, 4, 21, 67, 89)
  - V2 <- c(6, 24, 43, 2, 8)
  - V3<-c(10,25)
  - V1+V2
  - V1*V3 #Here V3 will be recycled as 10 25 10 25 10 until it matches the size of V1
  - V1^V3

Exp. 2: Demonstrate Vector operations in R

Functions on Vectors:

- Eg.:
  - length(V1) #Length of the Vector
  - sort(V2) #Sorts elements of a Vector
  - rep(V1,2) #Creates a vector by Repeating given Vector
  - rep(1,8) # Creates a vector by repeating 1 eight times
  - is.vector(V1) #Tells whether the object is vector or not
  - as.vector(M) #Converts non-vector objects into Vector
  - mean(V1) # Computes Average of Vector

Exp. 2: Demonstrate Vector operations in R

- Functions on Vectors
- Ex:
  - any(V1>50) #Checks whether any element satisfy the given condition
  - all(V1>10) #Checks whether all elements satisfy the given condition

Exp. 2: Demonstrate Vector operations in R

Task -1

- Store the names of your 5 friends in a vector and display it
- Display the names of first, third and fifth friend
- Update the name of fourth friend
- Display the names of all friends except third friend

Exp. 2: Demonstrate Vector operations in R

Task -2 : print sum, prod and p

```
x=10
y=20
sum=x+y
prod=x*y
p=c(x,y,sum,prod,8,10)
```

Exp. 2: Demonstrate Vector operations in R

Task -3 : print mean of x and print y

```
x<-c(4,7,2,8,3,10,13,6)
y<-x[3:6]
```

Exp. 2: Demonstrate Vector operations in R

Task -4 : print length and value of x, x1, i

```
> x <- c(88,5,12,13)
> x1 <- c(x[1:3],168,x[4])
> i <- 12
> j=1:(i-1)
```