While doing programming variable is use to store information.The store information of various data types like character, wide character, integer, floating point, double floating point, Boolean etc. Based on the data type of a variable, the operating system allocates memory and decides what can be stored in the reserved memory.The variables are assigned with R-Objects and the data type of the R-object becomes the data type of the variable.

Types of R objects

1. Vectors

A sequence of elements which share the same data type is known as vector. A vector supports logical, integer, double, character, complex, or raw data type.

```
#example: vector
apple <- c('red','green','yellow')
print(apple)
```

```
## [1] "red"    "green"  "yellow"
```

```
print(class(apple))
```

```
## [1] "character"
```

2. Lists

A list is an R-object which can contain many different types of elements inside it like vectors, functions and even another list inside it.

```
# example list
listitems <- list(c(4,3,2), 20.2,'hello')
print(listitems)
```

```
## [[1]]
## [1] 4 3 2
##
## [[2]]
## [1] 20.2
##
## [[3]]
## [1] "hello"
```

```
print(class(listitems))
```

```
## [1] "list"
```

3. Matrices

A matrix is a two-dimensional rectangular data set. It can be created using a vector input to the matrix function.

```
# example matrices
matr = matrix( c('a','b','c','d','e','f'), nrow = 2, ncol = 3, byrow = TRUE)
print(matr)
```

```
##      [,1] [,2] [,3]
## [1,] "a"  "b"  "c"
## [2,] "d"  "e"  "f"
```

```
class(matr)
```

```
## [1] "matrix" "array"
```

4. Arrays

While matrices are confined to two dimensions, arrays can be of any number of dimensions. The array function takes a dim attribute which creates the required number of dimension

```
# example array
a <- array(c('apple','mango','orange'), dim=c(3,3,2))
print(a)
```

```
## , , 1
##
##      [,1]     [,2]     [,3]
## [1,] "apple"  "apple"  "apple"
## [2,] "mango"  "mango"  "mango"
## [3,] "orange" "orange" "orange"
##
## , , 2
##
##      [,1]     [,2]     [,3]
## [1,] "apple"  "apple"  "apple"
## [2,] "mango"  "mango"  "mango"
## [3,] "orange" "orange" "orange"
```

```
print(class(a))
```

```
## [1] "array"
```

5. Factors

Factors are the r-objects which are created using a vector. It stores the vector along with the distinct values of the elements in the vector as labels. The labels are always character irrespective of whether it is numeric or character or Boolean etc. in the input vector. They are useful in statistical modeling.

```
# Create a vector.
apple_colors <- c('green','green','yellow','red','red','red','green')

# Create a factor object.
factor_apple <- factor(apple_colors)

# Print the factor.
print(factor_apple)
```

```
## [1] green  green  yellow red    red    red    green
## Levels: green red yellow
```

```
print(nlevels(factor_apple))
```

```
## [1] 3
```

6. Data Frames

Data frames in R language are the type of data structure that is used to store data in a tabular form which is of two dimensional. The data frames are special categories of list data structure in which the components are of equal length. R languages support the built-in function i.e. data.frame() to create the data frames

and assign the data elements. Data frame name is used to modify and retrieve data elements from the data frames. It is structured as column name by the component name also, structured as rows by the component values. It is widely used data structure while developing the machine learning models in data science projects.

```
# dataframe example
df <- data.frame(x=c(1,2,3),y=c(2,3,4),z=c(3,4,5))
print(df)
```

```
##   x y z
## 1 1 2 3
## 2 2 3 4
## 3 3 4 5
```

```
print(class(df))
```

```
## [1] "data.frame"
```

rbind() - row bind

The rbind function is is used to combine several vectors, matrices or dataframe by rows.

```
# rbind example
x1 <- c(7, 4, 4, 9)                  # Column 1 of data frame
x2 <- c(5, 2, 8, 9)                  # Column 2 of data frame
x3 <- c(1, 2, 3, 4)                  # Column 3 of data frame
data_1 <- data.frame(x1, x2, x3)     # Create example data frame

vector_1 <- c(9, 8, 7)               # Create example vector

rbind(data_1, vector_1)              # rbind vector to data frame
```

```
##   x1 x2 x3
## 1  7  5  1
## 2  4  2  2
## 3  4  8  3
## 4  9  9  4
## 5  9  8  7
```

cbind() - column bind

It is used to combine two data frames with the same number of multiple rows into a single data frame.

```
# cbind example
data_1 <- data.frame(x1 = c(7, 3, 2, 9, 0),          # Column 1 of data frame
                     x2 = c(4, 4, 1, 1, 8),          # Column 2 of data frame
                     x3 = c(5, 3, 9, 2, 4))          # Column 3 of data frame
y1 <- c(9, 8, 7, 6, 5)                               # Create vector
data_new1 <- cbind(data_1, y1)                       # cbind vector to data frame
print(data_new1)
```

```
##   x1 x2 x3 y1
## 1  7  4  5  9
## 2  3  4  3  8
## 3  2  1  9  7
## 4  9  1  2  6
## 5  0  8  4  5
```