R Data Types

**Everything** in R is an object.

* Vectors
* Lists
* Matrices
* Arrays
* Factors
* Data Frames

class() - what kind of object is it (high-level)?

typeof() - what is the object’s data type (low-level)?

length() - how long is it? What about two dimensional objects?

attributes() - does it have any metadata?

Vectors

A vector is a collection of elements.

There are six data types of these atomic vectors, also termed as six classes of vectors.

* Character🡪 'a' , '"good", "TRUE", '23.4'
* numeric (real or decimal)🡪 12.3, 5, 999
* integer🡪 2L, 34L, 0L
* logical🡪 TRUE, FALSE
* complex🡪 3 + 2i
* raw🡪 "Hello" is stored as 48 65 6c 6c 6f---🡪v🡨charToRaw(“hello”)

a<-vector()

print(typeof(a))

b<-vector("character", length = 5)

print(typeof(b))

print(length(b))

c<-vector("integer", length = 5)

print(typeof(c))

print(length(d))

d<-vector("numeric", length = 5)

|  |
| --- |
| > x<-character(5)  > typeof(x)  > length(x)  > print(x) |
|  |
| |  | | --- | | > x<-logical(5)  > typeof(x)  > length(x)  > print(x) | |

vectors can create directly specifying their content. R will then guess the appropriate mode of storage for the vector

x<-1000

print(x)

print(class(x))

print(length(x))

---------------------------------------

p <- c(1, 2, 3)

q <- c(1L, 2L, 3L)

x <- c(**TRUE**, **TRUE**, **FALSE**, **FALSE**)

y <- c("lokesh", "python", "r")

typeof(y)

length(y)

class(y)

y <- c(y, "datascience")

y <- c("django", y)

z<- c("lokesh", 1000, 123L, 3+4i)

print(typeof(z))

print(length(z))

print(z)

Missing Data

R supports missing data in vectors. They are represented as NA (Not Available) and can be used for all the vector types covered in this lesson:

x <- c(0.5, **NA**, 0.7)

x <- c(**TRUE**, **FALSE**, **NA**)

x <- c("a", **NA**, "c", "d", "e")

x <- c(1+5i, 2-3i, **NA**)

The function is.na() indicates the elements of the vectors that represent missing data, and the function anyNA()returns TRUE if the vector contains any missing values:

x <- c("a", **NA**, "c", "d", **NA**)

y <- c("a", "b", "c", "d", "e")

is.na(x)

is.na(y)

anyNA(x)

anyNA(y)

### Other Special Values

Inf is infinity. You can have either positive or negative infinity.

1/0

[1] Inf

NaN means Not a Number. It’s an undefined value.

0/0

[1] NaN

## Matrix

All columns in a matrix must have the same mode(numeric, character, etc.) and the same length. The general format is

mymatrix <- **matrix(**vector, **nrow=**r, **ncol=**c, **byrow=**FALSE,  **dimnames=list(**char\_vector\_rownames, char\_vector\_colnames**))**

**byrow=TRUE** indicates that the matrix should be filled by rows. **byrow=FALSE** indicates that the matrix should be filled by columns (the default). **dimnames** provides optional labels for the columns and rows.

x = matrix( c('a','a','b','c','b','a'), nrow = 2, ncol = 3, byrow = TRUE)

print(x)

x = matrix( c('a','a','b','c','b','a'), nrow = 2, ncol = 3, byrow = FALSE) print(x)

y<-matrix(1:20, nrow=5,ncol=4)

y[,4] # 4th column of matrix  
y[3,] # 3rd row of matrix   
y[2:4,1:3] # rows 2,3,4 of columns 1,2,3

z= matrix( c('a','a','b','c',TRUE,100), nrow = 2, ncol = 3, byrow = FALSE) prin)

cells <- c(1,26,24,68)  
rnames <- c("R1", "R2")  
cnames <- c("C1", "C2")   
mymatrix <- matrix(cells, nrow=2, ncol=2, byrow=TRUE,

dimnames=list(rnames, cnames))

print(mymatrix)

cells <- c(1,26,NA,68)

mymatrix <- matrix(cells, nrow=2, ncol=2, byrow=FALSE)

print(matrix)

## 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.

a <- array(c('green','yellow'),dim = c(3,3,2))

print(a)

List

-----

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.

list1 <- list(c(2,5,3),21.3,sin)

print(list1)

v=c(10,20,30)

m<-matrix(c(50,60,70,80),nrow=2,ncol=2)

a<-array(c(91,92,93), dim = c(3,3,2))

mylist<-list(v,m,a)

mylist[1]

mylist[2]

## 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.

Factors are created using the **factor()** function. The **nlevels**functions gives the count of levels.

apple\_colors <- c('green','green','yellow','red','red','red','green')

factor\_apple <- factor(apple\_colors)

print(factor\_apple)

print(nlevels(factor\_apple))

gender <- c(rep("male",20), rep("female", 30))

## Data Frames

------------------------

Data frames are tabular data objects. Unlike a matrix in data frame each column can contain different modes of data. The first column can be numeric while the second column can be character and third column can be logical. It is a list of vectors of equal length.

d <- c(1,2,3,4)  
e <- c("red", "white", "red", NA)  
f <- c(TRUE,TRUE,TRUE,FALSE)  
mydata <- data.frame(d,e,f)  
names(mydata)

print(mydata)

BMI <- data.frame(

gender = c("Male", "Male","Female"),

height = c(152, 171.5, 165),

weight = c(81,93, 78),

Age = c(42,38,26)

)

print(BMI)

BMI[3:4]

BMI[c("height","Age")]