**DATA TYPES**

Generally, while doing programming in any programming language, you need to use various variables to store various information. Variables are nothing but reserved memory locations to store values. This means that, when you create a variable you reserve some space in memory.

You may like to 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. There are many types of R-objects. The frequently used ones are −

* **Vectors:** A basic data structure of R containing the same type of data.
* **Matrices:** A matrix is a two-dimensional rectangular data set. It can be created using a vector input to the matrix function.
* **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 modelling.
* **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.
* **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.
* **Modes:** All objects have a certain *mode*. Some objects can only deal with one *mode* at a time, others can store elements of multiple *modes*. R distinguishes the following modes:

1. **integer**: integers (e.g. 1, 2 or -69)

2. **numeric**: real numbers (e.g 2.336, -0.35)

3. **complex**: complex or imaginary numbers

4. **character**: elements made up of text-strings (e.g. "text", "Hello World!", or

"123")

5. **logical**: data containing logical constants (i.e. TRUE and FALSE)

By *atomic*, we mean the vector only holds data of a single type.

* **character**: "a", "swc"
* **numeric**: 2, 15.5
* **integer**: 2L (the L tells R to store this as an integer)
* **logical**: TRUE, FALSE
* **complex**: 1+4i (complex numbers with real and imaginary parts) R provides many functions to examine features of vectors and other objects, for example
* 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?

|  |  |  |
| --- | --- | --- |
| **Data Type** | **Example** | **Code** |
| |  |  |  | | --- | --- | --- | | Logical | TRUE, FALSE | v <- TRUE  print(class(v))  it produces the following result −  [1] "logical" | | Numeric | 12.3, 5, 999 | v <- 23.5  print(class(v))  it produces the following result −  [1] "numeric" | | Integer | 2L, 34L, 0L | v <- 2L  print(class(v))  it produces the following result −  [1] "integer" | | | |
| |  |  |  | | --- | --- | --- | | Complex | 3 + 2i | v <- 2+5i  print(class(v))  it produces the following result −  [1] "complex" | | Character | 'a' , '"good", "TRUE", '23.4' | v <- "TRUE"  print(class(v))  it produces the following result −  [1] "character" | | Raw | "Hello" is stored as 48 65 6c 6c 6f | v <- charToRaw("Hello")  print(class(v))  it produces the following result −  [1] "raw" | | | |

**Variables and assignment**

A variable provides us with named storage that our programs can manipulate. A variable in R can store an atomic vector, group of atomic vectors or a combination of many Robjects. A valid variable name consists of letters, numbers and the dot or underline characters. The variable name starts with a letter or the dot not followed by a number.

The variables can be assigned values using leftward, rightward and equal to operator. The values of the variables can be printed using print() or cat() function. The cat() function combines multiple items into a continuous print output.

**Examples:**

# Assignment using equal operator.

var.1 = c(0,1,2,3)

# Assignment using leftward operator.

var.2 <- c("learn","R")

# Assignment using rightward operator.

c(TRUE,1) -> var.3

print(var.1)

cat ("var.1 is ", var.1 ,"\n")

cat ("var.2 is ", var.2 ,"\n")

cat ("var.3 is ", var.3 ,"\n")