

R Data Types and Data Structures

Everything in R is an object.

R has 6 basic data types. (In addition to the five listed below, there is also raw which will not be discussed here.)

- character
- numeric (real or decimal)
- integer
- logical
- complex

Elements of these data types may be combined to form data structures, such as atomic vectors. When we call a vector atomic, we mean that the vector only holds data of a single data type. Below are examples of atomic character vectors, numeric vectors, integer vectors, etc.

- 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)

file:///C:/Users/pmuhuri/Downloads/An_Introduction_to_Data_Analysis_and_Graphics_with.pdf
(page 11)

The term “data type” in R refers to the type of data that is present in a data structure, and does not describe the data structure itself. There are four common types of data in R: numerical, character, logical, and complex numbers. These are referred to as modes in R and are shown below.

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?
- attributes() - does it have any metadata?

```
df1 <- data.frame(tf= c(TRUE, FALSE),
                  nume = c(45, 37.49),
                  integ = c(35L, 20L),
                  complx = c(1+2i, 0+1i),
                  char = c("CANCER", "Diarrhea"))
df1
```

```
##      tf  nume integ complx    char
## 1 TRUE 45.00   35   1+2i  CANCER
## 2 FALSE 37.49   20   0+1i Diarrhea
```

```
paste("Class of various data types")
```

```
## [1] "Class of various data types"
```

```
sapply(df1, class)
```

```
##          tf          nume          integ          compl          char
## "logical" "numeric"  "integer"  "complex" "character"
```

```
paste("mode of various data types")
```

```
## [1] "mode of various data types"
```

```
sapply(df1, mode)
```

```
##          tf          nume          integ          compl          char
## "logical" "numeric"  "numeric"  "complex" "character"
```

```
paste("typeof of various data types")
```

```
## [1] "typeof of various data types"
```

```
sapply (df1, typeof)
```

```
##          tf          nume          integ          compl          char
## "logical" "double"  "integer"  "complex" "character"
```

From: <https://www.mail-archive.com/r-help@r-project.org/msg17169.html> :

‘mode’ is a mutually exclusive classification of objects according to their basic structure. The ‘atomic’ modes are numeric, complex, character and logical. Recursive objects have modes such as ‘list’ or ‘function’ or a few others. An object has one and only one mode.

‘class’ is a property assigned to an object that determines how generic functions operate with it. It is not a mutually exclusive classification. If an object has no specific class assigned to it, such as a simple numeric vector, it’s class is usually the same as its mode, by convention.

Changing the mode of an object is often called ‘coercion’. The mode of an object can change without necessarily changing the class.

You can use the mode or class function to see what type of data is stored in any symbolic variable.

Mode and class are not identical—as Bill Venables writes: — mode‘ is a mutually exclusive classification of objects according to their basic structure. . . class‘ is a property assigned to an object that determines how generic functions operate with it. (<http://tolstoy.newcastle.edu.au/R/e4/help/08/04/8330.html>).

The main difference between class and typeof is that the first can be defined by the user, but the type cannot.