

Objects and operations in R

Or one of the most common sources of **frustration** in R

An Introduction to R

Notes on R: A Programming Environment for Data Analysis and Graphics
Version 3.6.1 (2019-07-05)

By W. N. Venables, D. M. Smith and the R Core Team

O'REILLY



Hands-On
Programming
with R

WRITE YOUR OWN FUNCTIONS AND SIMULATIONS

Garrett Golemund
Foreword by Hadley Wickham

Download the script and files

https://github.com/Batis007/R_course-UA

Data types vs Data structures

R has 6 basic data types.

Character

Numeric (real or decimal)

Integer

Logical

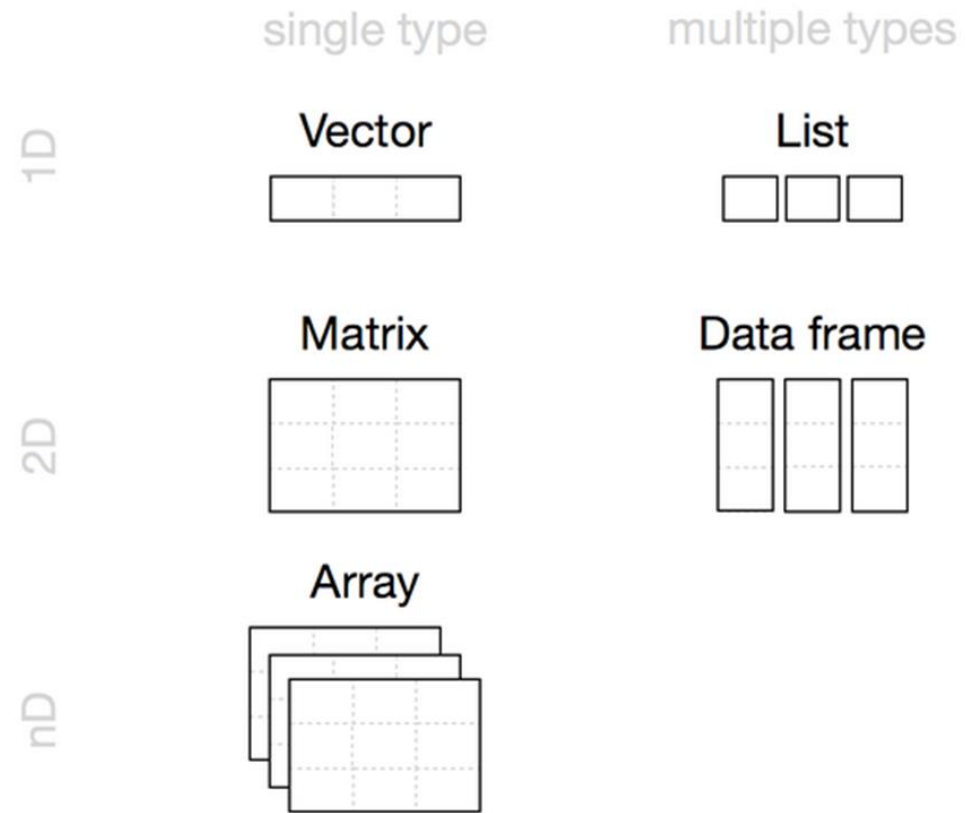
Complex

Raw

**And what about data
structures in R?**

Data structures

- Vector
- Matrix
- Array
- Lists
- Data Frames



Let's create
some objects!

Good names	Names that cause errors
a	1trial
b	\$
F00	^mean
my_var	2nd
.day	!bad

Vector

Each atomic vector stores its values as a one-dimensional vector, and each atomic vector can only store one type of data.

There are six common types of R Atomic Vectors:

Doubles

Integers

Characters

Logicals

Complex (Bonus lecture – extra fee)

Raw (Bonus lecture – extra fee)

Operator	Description
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
==	Equal to
!=	Not equal to

Bonus

Complex and Raw

Doubles, integers, characters, and logicals are the most common types of atomic vectors in R, but R also recognizes two more types: complex and raw. It is doubtful that you will ever use these to analyze data, but here they are for the sake of thoroughness.

Complex vectors store complex numbers. To create a complex vector, add an imaginary term to a number with `i`:

```
comp <- c(1 + 1i, 1 + 2i, 1 + 3i)
comp
## 1+1i 1+2i 1+3i

typeof(comp)
## "complex"
```

Raw vectors store raw bytes of data. Making raw vectors gets complicated, but you can make an empty raw vector of length n with `raw(n)`. See the help page of `raw` for more options when working with this type of data:

```
raw(3)
## 00 00 00

typeof(raw(3))
## "raw"
```

Matrix

Matrices store values in a two-dimensional array, just like a matrix from linear algebra.

Array

The array function creates an n-dimensional array. For example, you could use array to sort values into a cube of three dimensions or a hypercube in 4, 5, or n dimensions

Lists

Lists are like atomic vectors because they group data into a one-dimensional set. However, lists do not group together individual values; lists group together R objects, such as atomic vectors and other lists.

Data frames

Data frames are the two-dimensional version of a list. They are far and away the most useful storage structure for data analysis, and they provide an ideal way to store an entire deck of cards. You can think of a data frame as R's equivalent to the Excel spreadsheet because it stores data in a similar format.