Exercises-1: Intro to R- types

Nathan Green, Imperial College London

07/09/2019

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

1. Type `x <- "dataset"` and `y <- 1:10`. What are the type and attributes of `x` and `y`?

R has many data structures. These include

- atomic vector
- list
- matrix
- data frame
- factors
- Vectors

A vector is the most common and basic data structure in R and is pretty much the workhorse of R. Technically, vectors can be one of two types:

- atomic vectors
- lists

although the term "vector" most commonly refers to the atomic types not to lists.

The Different Vector Modes

A vector is a collection of elements that are most commonly of mode character, logical, integer or numeric.

You can create an empty vector with `vector()`. (By default the mode is logical. You can be more explicit as shown in the examples below.) It is more common to use direct constructors such as `character()`, `numeric()`, etc.

2. Create a character vector of length 5.

3. Create character, numeric, logical vectors directly. What do you see?

You can also create vectors by directly specifying their content. R will then guess the appropriate mode of storage for the vector. For instance:

`x <- c(1, 2, 3)` will create a vector x of mode numeric. These are the most common kind, and are treated as double precision real numbers. If you wanted to explicitly create integers, you need to add an L to each element (or coerce to the integer type using as.integer()).

4. Create a vector of 3 integers.

Using TRUE and FALSE will create a vector of mode logical:

`y <- c(TRUE, TRUE, FALSE, FALSE)` While using quoted text will create a vector of mode character:

`z <- c("Sarah", "Tracy", "Jon")`

Examining Vectors

The functions `str()` provide useful information about your vectors and R objects in general.

5. View the structure of `z` above.

Adding Elements

The function `c()` (for combine) can also be used to add elements to a vector.

6. Attach `Annette` to the end of `z`. Attach `Greg` to the start.

Vectors from a Sequence of Numbers

You can create vectors as a sequence of numbers using `:`.

7. Create a series from 1 to 10 using `:`. Create the same output using `seq()`. Change the step size to 0.1.
8. Create the sequence above in reverse from 10 to 1.