Data Types

Author: Cole Brookson **Date:** 13 June 2022

Objects and OOP

R is an Object-Oriented Programming (OOP) language which means it functions best by storing things as "objects". An object is a data structure that has some attributes and a set of methods that act on those attributes.

Object-oriented languages (often called "high-level" languages) are typically more intuitive, as whenever you want to assign some value(s) to a variable and save those values for later, you can make the variable to an object which will be saved by R in the environment for future use.

Data Types

There are six main types in R, but we will discuss only the first four.

1. **Characters**
2. **Numerics (real or decial)**
3. **Integers**
4. **Logicals**
5. Complex
6. Raw

When we assign variables, it is important to know what type we are using. We can check what the `type` of an object, and get other useful information about it, with a number of useful commands:

```
x <- "abc"
typeof(x)
```

```
## [1] "character"
```

If we want a more detailed answer we can ask what the structure (`str()`) of an object is

```
str(x)
```

```
##  chr "abc"
```

We see the output is slightly different here, with the function telling us the type of the object (`chr`) and also the content of the object (`"abc"`).

Characters A character is any alphanumeric string that begins with a letter.

```
x <- "abc"
y <- "abc123"
typeof(x)
```

```
## [1] "character"
```

```
typeof(y)
```

```
## [1] "character"
```

Numerics A numeric type is any non-integer number.

```
z <- 3.14
typeof(z)
```

```
## [1] "double"
```

The output `"double"` here refers to the fact that R automatically stores numeric data types with "double" precision.

Integers An integer is a non-decimal whole number. **Note:** In R, the default is to store values as numeric unless explicitly told otherwise. We can see that if we make a new variable with only an integer value R will store it as a numeric type.

```
x1 <- 2
typeof(x1)
```

```
## [1] "double"
```

To force R to store it as an integer we can simply add an `L` after the value

```
x2 <- 2L
typeof(x2)
```

```
## [1] "integer"
```

Logicals Logical types are simply TRUE or FALSE.

```
x3 <- TRUE
typeof(x3)
```

```
## [1] "logical"
```