

Introduction to R

Alla Tambovtseva

R as a calculator

Basic calculations:

```
1 + 4 # addition
```

```
## [1] 5
```

```
4 - 9 # subtraction
```

```
## [1] -5
```

```
6 * 5 + 7 / 2 # multiplication and division
```

```
## [1] 33.5
```

```
sqrt(45) # taking square root
```

```
## [1] 6.708204
```

```
6 ^ 3 # raising to power
```

```
## [1] 216
```

```
6 ** 3 # the same
```

```
## [1] 216
```

Note: if you are planning to work both in R and Python, you had better memorize the latter variant of raising a number to some power (via `**`) since in Python the operator `^` corresponds to the bitwise addition that has nothing in common with powers.

In R we can calculate logarithms as well. By default the `log()` function returns the natural logarithm, the logarithm of the base e . In English books it is usually denoted as `log` as well, in Russian ones it is denoted as `ln`.

```
log(4)
```

```
## [1] 1.386294
```

We can also specify the base of a logarithm adding the option `base`:

```
log(4, base = 2)
```

```
## [1] 2
```

If we are interested in a decimal logarithm, a logarithm of a base 10, there is a special function in R (but, of course, we can simply add `base=10` as in the previous example):

```
log10(4)
```

```
## [1] 0.60206
```

In R we can easily perform operations with mathematical constants. Take e (`exp`), for example:

```
exp(1) # e = e^1
```

```
## [1] 2.718282
```

```
exp(2) # e^2
```

```
## [1] 7.389056
```

Variables in R

Names of the variables in R can contain letters, numbers, dots and underscores, but the name of a variable cannot start with a number (as in many programming languages). A name of a variable should not coincide with the reserved R words and operators (like if, else, for, while, etc).

Both operators `->` and `=` can be used for assigning values to variables, but `->` is a 'canonical' R operator that is usually applied in practice. In other words, writing code with `=` is technically correct, but not cool and has to be avoided :)

```
a <- 3  
a
```

```
## [1] 3
```

We can change the value of a variable and save it again with the same name:

```
a <- 3  
a <- a + 2  
a # now it is 3 + 2 = 5
```

```
## [1] 5
```

We can also assign text values to variables. A text is usually written in quotes:

```
s <- "hello"  
s
```

```
## [1] "hello"
```

It does not matter which quotes, single `'` or double `"` we will use. The only important thing is that the opening and the closing quote should be of the same type, so it is not allowed to write something like this: `"hello'`.

There are many functions that are aimed at working with text variables (in R they are called character variables), but now we will not concentrate on them. Just as an example, look at the function `upper()` that converts all letters into capital ones:

```
toupper(s)
```

```
## [1] "HELLO"
```