Introduction to Numerical Analysis

Day 1: Introduction to R (2)

Joaquin Cavieres

Geoinformatics, Bayreuth University

Outline

Simple commands

- The instruction “#” allow us “comment” lines and in this way it does not read.

- If we want to write an algorithm, then:
  - Use correct names
  - Don’t use repeated lines in the code
  - Don’t assume that the names should be remembered in the next lines of code

Simple commands

$> 2 + 2$

```
[1] 4
```

$> 2^2$

```
[1] 4
```

$> 2 * (1 + 1)$

```
[1] 4
```

Examples

Write in R and calculates:

$> 1 + 2(3 + 4)$

Examples

Write in R and calculates:

$> 1 + 2(3 + 4)$

$> log(4^3 + 3^{2+1})$

Examples

Write in R and calculates:

$> 1 + 2(3 + 4)$

$> log(4^3 + 3^{2+1})$

$> \sqrt{(4 + 3)(2 + 1)}$

Examples

Write in R and calculates:

$> 1 + 2(3 + 4)$

$> log(4^3 + 3^{2+1})$

$> \sqrt{(4 + 3)(2 + 1)}$

$> (\frac{1+2}{2+4})^2$

Creating objects

- Each result of any mathematical operation can be saved in a "object".
    - Numbers
    - Characters
    - Tables
    - Vectors/Matrices
    - Graphs
    - Statistical model
    - Etc...

Assign

We can assign a specific value for a variable, for example:

```
variable <- x
```

The “variable” have now the value of x.

but also we can assign with the sign “=”, for example:

```
variable = x
```

Assign

Also we can assign characters to a variable:

```
name <- ‘‘John’’
```

To “name" we assigned the character “John”.

If we want to assign a name with a space, we have to do:

```
name <- ‘‘John Cavieres’’
```

Seeing the created objects

There are different ways to visualise the created objects:

```
print(name)
```

```
[1] John Cavieres.
```

```
name
```

```
[1] John Cavieres.
```

If we want to manipulate an object we can do: `x <- 2`

```
x*2
```

```
[1] 4.
```

Remove created objects

We can write `list()` to see the created objects

```
ls()
```

Remove created objects

We can write `list()` to see the created objects

```
ls()
```

```
[1] "name" "variable"
```

Remove created objects

We can write `list()` to see the created objects

```
ls()
```

```
[1] "name" "variable"
```

To remove a particular object we write `rm()`, for example, if we want to remove the object "name", we write:

```
rm(name)
```

Remove created objects

We can write `list()` to see the created objects

```
ls()
```

```
[1] ‘‘name’’ ‘‘variable’’
```

To remove a particular object we write `rm()`, for example, if we want to remove the object “name”, we write:

```
rm(name)
```

```
ls()
```

Remove created objects

We can write `list()` to see the created objects

```
ls()
```

```
[1] ‘‘name’’ ‘‘variable’’
```

To remove a particular object we write `rm()`, for example, if we want to remove the object “name”, we write:

```
rm(name)
```

```
ls()
```

```
[1] ‘‘variable’’
```

Remove created objects

If we want to remove all the created objects `rm(list=ls())`

Remove created objects

If we want to remove all the created objects `rm(list=ls())`

Review if there are more objects with `list()`...

Types of data

There are different types of data to be used in R and they differ in the way that are saved in the computer.

- Numeric (integer, floating point, etc)
- Logic (boolean, true/false)
- Characters(text)

The type of data is not obvious, furthermore when they are read from external sources, for the same is necessary to know what type the data are.

Type of data

```
>variable2 <- 2

>variable2

[1] 2.

>mode(variable2)

[1] ‘‘numeric’’.

>is.numeric(variable2)

[1] ‘‘TRUE’’.
```

Type of data

Similar functions can be applied to character objects:

```
>is.character(variable2)
```

```
[1] ‘‘FALSE’’.
```

```
>is.character(name)
```

```
[1] ‘‘TRUE’’.
```

The class “numeric” and “character” are the mos commonly found in the declaration of objects.

See you next class!...


Howard, J. P. (2017). Computational Methods for Numerical Analysis with R. CRC Press.