R + RStudio

R is a scripted language providing all functionalities of a programming language by attaching libraries to C/C++/Fortran compiled code.

The scientific community provides many added functionalities that are available on packages that can be installed (see later)

RStudio is an IDE (development environment). Also available in the cloud (RStudio-Server) if you like to have all your data on the cloud.

BELOW SOME BASICS R

STRUCTURES

Try to run the following lines

```
character
integer
double
numeric
list
vector
array
matrix
data.frame
```

Basic math

```
4+4
```

```
## [1] 8
```

```
2**3 ## this is javascript-style
```

```
## [1] 8
```

```
2^3
```

```
## [1] 8
```

```
## remember ANGLES are in RADIANS
sin(90/180*pi)
```

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

Assigning values to variables

Use the `<-` operator or `<<-` if you want to go up of scope (see later)

```
greeting <- "Hello class!"
print(greeting)
```

```
## [1] "Hello class!"
```

```
print(greeting[[1]])
```

```
## [1] "Hello class!"
```

```
age <- 28 # Numeric variable
name <- "Frank" # Character variable
# logical values TRUE or FALSE
do.I.like.this.course <- TRUE # logical variable
#
do.I.like.this.course == F
```

```
## [1] FALSE
```

FACTORS!

Factors are a smart way that R uses to create classes:

```
countries <- c( rep("Italy", 40), rep("Germany", 20), rep("China", 10), rep("India", 10)  )
table(countries)
```

```
## countries
##   China Germany   India   Italy 
##      10      20      10      40
```

```
save(countries, file="countries.rda")

countries.factors<-as.factor(countries)
levels(countries.factors)
```

```
## [1] "China"   "Germany" "India"   "Italy"
```

```
save(countries.factors, file="countries.factors.rda")
#' Check SIZE difference of file!
file.size.difference <- file.size("countries.rda") / file.size("countries.factors.rda")
message("File with factors is " , round(file.size.difference, 2), "X smalle than file with full characte
```

```
## File with factors is 0.72X smalle than file with full characters
```

More comple objects

```
#raster # not found?? Additional functionalities require additional libraries/packages
library(raster)
```

```
## Loading required package: sp
```

```
raster
```

```
## standardGeneric for "raster" defined from package "raster"
##
## function (x, ...)
## standardGeneric("raster")
## <bytecode: 0x55e2067c0978>
## <environment: 0x55e20666b5f0>
## Methods may be defined for arguments: x
## Use  showMethods("raster")  for currently available ones.
```

```
#' but we use a better library for raster... terra
library(terra)
```

```
## terra version 1.0.10
```

HELP!

Let's get some help about a function returning an object

```
?terra::rast
??terra::vect
```

Final exc