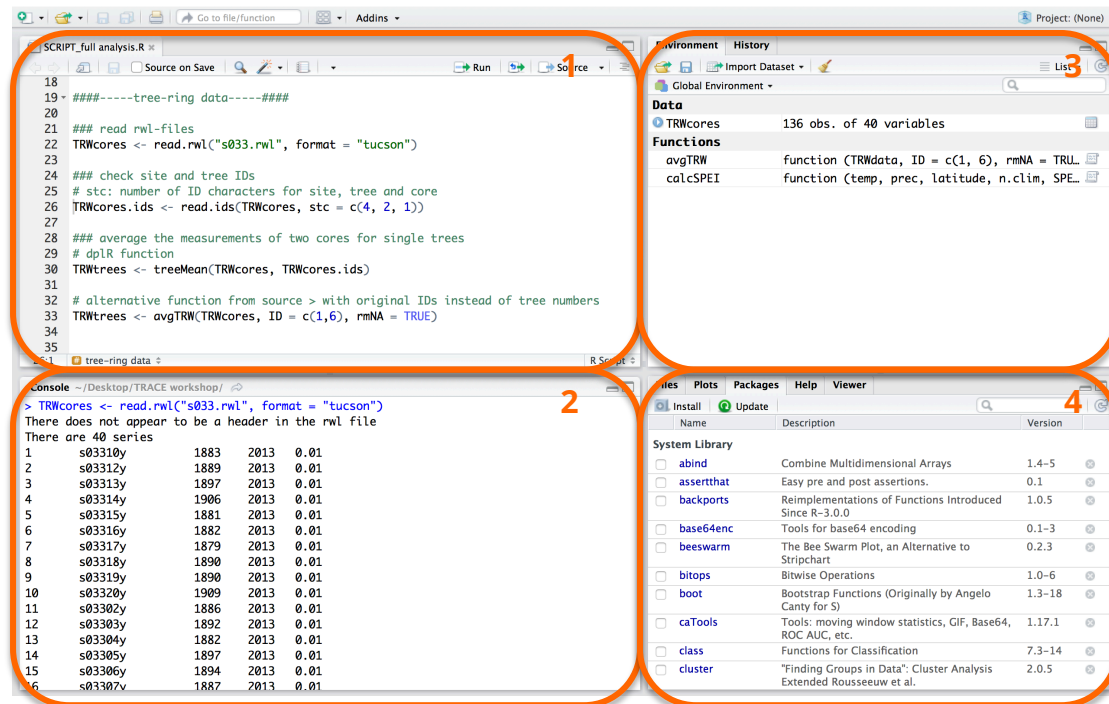


Introduction to R / RStudio

RStudio is a graphical user interface for R that offers convenient functionalities for writing code, plotting, tracking history, debugging and workspace management.

Panes

In RStudio there generally are four panes:



Pane 1: View files and data. In the file pane, scripts and data will be opened. Each file opens in its' own tab. A script is a text file with commands you want to execute in the console. Scripts are easy-to-use when analyzing data, for example in your thesis, as they offer the possibility to review and rerun your code.

Pane 2: Console. In the console, all calculations take place. In the command line interface, you type the commands you want to execute (by pressing return).

Pane 3: Environment and history. Under the tab Environment, a list of all the objects you have loaded into your workspace is shown. For example, all datasets you have loaded and objects you have created. Under History, the complete history of code you have typed is saved.

Pane 4: Files, plots, packages and help. This pane contains the following tabs:

- files: where you can see your files and folders
- plots: where your created plots will appear
- packages: a list of packages you have access to and are loaded
- help: where help files will open after a help-command in the console or clicking on a package under 'packages'.

Calculator

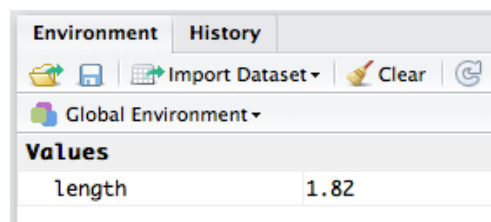
The console can be used as a simple **calculator** for addition, subtraction, multiplication, and division using the standard operators +, -, *, and /.

```
> 1+1
[1] 2
> (3-1)/2
[1] 1
> 2*3
[1] 6
```

In order to use the console effectively, one should know how to store data in so-called **variables**. In R, a variable is a name that is assigned to (a) particular value(s). Instead of numbers, variable names can be used in calculations.

Values can be assigned to variables using one of the following three operators <-, = and ->, for example to assign the number 1.82 to the variable 'length':

```
> length<-1.82
> length=1.82
> 1.82->length
> length
[1] 1.82
```



In each of the three cases, you can confirm the assigned value by typing 'length' and pressing return. The created variable is automatically saved in the Environment.

Similarly, you can assign the value of 65 to the variable 'weight' using the command `weight <- 65`, and issue a command such as `weight / length^2` to get a resulting body-mass index of 19.6.

Data formats

Data can appear as vector or matrix.

A **vector** has a one-dimensional structure (e.g. a list of numbers) and can be created as follows: `length <- c(1.55,1.62,1.57,1.83,1.97,1.75,1.89,1.56,1.90,1.67)`

You can confirm the values of this vector by typing 'length' in the console and pressing return.

You can call a specific value in the vector by entering its numerical position inside of square brackets:

```
> length <- c(1.55,1.62,1.57,1.83,1.97,1.75,1.89,1.56,1.90,1.67)
> length[2]
[1] 1.62
```

The mean of all these lengths can be calculated with `mean(length)`. Some other functions read `sum()`, `min()` and `max()`.

A **matrix** has a two-dimensional structure and can be created as follows:
`mat <- matrix(c(2, 4, 3, 1, 5, 7, 3, 2, 4), nrow = 3, ncol = 3, byrow = TRUE)`

In a matrix, all elements should be of the same type (e.g. numerical, logical, character).

You can call (a) specific value(s) in a matrix with its (their) coordinates:

- e.g. the value in row 2 and column 3: `mat[2,3]`
- only row 1 (all columns): `mat[1,]`
- only column 1 (all rows): `mat[,1]`
- all rows except row 1: `mat[-1,]`
- all columns except column 1: `mat[,-1]`
- to save it as new matrix: `mat2 <- mat[,-1]`

Besides vectors and matrices, another data structure exists in R: **data frame**. This is a list of vectors of equal length. Columns of a data frame may contain different data types (e.g. numerical, logical, character). Data frames are less memory efficient as compared to matrices, but they are easy if you want to refer to column names in subsequent coding.

To delete (a) certain row(s) or column in your data frame:

- `df[-1,]`
- `df[-(1:3),]`
- `df[-c(1,3,5),]`
- `df[, "columnname"] <- NULL` or `df$columnname <- NULL`
- `df[,c("column1", "column2")] <- list(NULL)`

For some functions in R, data should be structured as a matrix, whereas for others as a data frame. To check the format of your data, use the functions `is.data.frame()`, `is.matrix()`, `is.vector()`, `is.numeric()`, etc.

To remove objects from your Environment, use the remove function `rm()` as follows:

`rm(length)`

`rm(mat, weight)`

Parentheses in R

In R various types of parentheses are used:

- `()` for functions
- `[]` for indices, in matrix [rows,columns], for part of a list [[]]
- `{}` for loops (and functions you write yourself)

Packages and their functions

A great feature of the open-source software R is that all users can write and publish packages for specific analyses on the repository CRAN (according to well-defined standards). Such packages contain functions, data and (or) code. By default, R comes with a standard set of packages to perform basic statistical operations. Additional packages can be manually downloaded and installed. To be able to use a specific package in your R session, you should load the package by ticking the respective checkbox under 'Packages'. Alternatively, one may type `library("package name")` in the console.

Help pages for all functions can be loaded using `?` or `help()`. For example, if you are interested in knowing more about the function 'plot', type: `?plot` or `help(plot)`. Each help page contains a description on the function, how to use it, and which arguments can be specified.

Problem solutions

When you have sent a command to the console, look into the console, if there still is a '+' then you have not finished the command (often a bracket () is missing). If this was not your intention, press 'Esc', correct the command and execute it again.

If you call an object and it returns: 'Object not found' there are two possible reasons:

- spelling mistake
- you have not yet loaded the object in your Environment