Introduction to R

Antonia Santos

Program

PART I

Introduction to R and base R programming

PART II

Data manipulation

PART III

Data visualisation

PART IV

Introduction to modelling in R

Bibliography

- R Manual (`https://cran.r-project.org/doc/manuals/R-intro.html`)
- R for Data Science (2e) (`https://r4ds.hadley.nz`)

Part I

Introduction to R and Base R Programming

1. CODING
2. R AND RStudio
3. BASIC CONCEPTS

CODING

What is coding?

CODING

Programming languages.

R AND RStudio

What is R? And RStudio?

R AND RStudio

- R
  - Created by Ross Ihaka and Robert Gentleman (University of Auckland/R Development Core Team).
  - Open source.

R AND RStudio

- R
  - Created by Ross Ihaka and Robert Gentleman (University of Auckland/R Development Core Team).
  - Open source.
- RStudio is an Integrated Development Environment (IDE).

R AND RStudio

- R
  - Created by Ross Ihaka and Robert Gentleman (University of Auckland/R Development Core Team).
  - Open source.
- RStudio is an Integrated Development Environment (IDE).
- Download:
  - R: `https://www.r-project.org`.
  - RStudio: `https://www.rstudio.com`.

BASIC CONCEPTS

RStudio interface.

- Interface
  - Source pane
  - Console pane
  - Environment pane (Environment, History, Connections, Build, VCS, and Tutorial)
  - Output pane (Files/ Plots / Packages / Help)

BASIC CONCEPTS

RStudio interface.

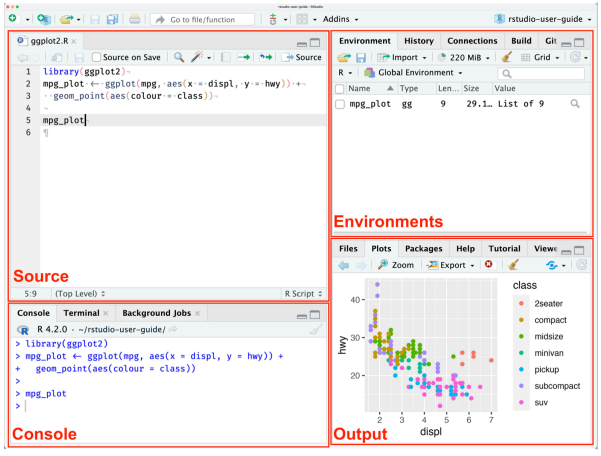


Figure: `https://docs.posit.co/ide/user/ide/guide/ui/ui-panes.html`

BASIC CONCEPTS

RStudio interface.

- Source pane
  - This is where you write and edit your R scripts (.R files).
  - You can write code, comments, and documentation in this area, and run selected lines by pressing Ctrl+Enter (Windows) or Cmd+Enter (macOS).
  - This area allows you to save your code, which is essential for keeping a record of your analysis.
- Console pane
  - The console is where you can execute commands interactively. It shows the output of the code you run and is useful for testing small code snippets or running analyses directly.
  - You can type commands directly into the console and see immediate feedback or results.
- Environment pane
  - Environment Tab: Displays all the objects (e.g., data frames, variables, functions) currently in your R workspace. You can inspect and manage your data and objects here.
  - History Tab: Shows a list of all the commands you have previously executed during the current R session. This is helpful for tracking your steps or re-running commands.

BASIC CONCEPTS

RStudio interface.

- Output pane
  - Files Tab: A file explorer that lets you browse files in your working directory and manage files (open, delete, move, etc.).
  - Plots Tab: Displays any plots or graphs that you generate with R (e.g., ggplot2 visualisations). You can export these plots as images or PDFs.
  - Packages Tab: Manage R packages installed in your system. You can install, load, or remove packages from here.
  - Help Tab: Provides access to R's help documentation. You can search for information about functions, packages, and datasets.
  - Viewer Tab: Used to display web-based content, such as HTML reports generated from R Markdown files.

Note 1: The RStudio layout may look slightly different based on your configuration, but the core functionality remains the same.

BASIC CONCEPTS

Files.

- Files
  - **.R** - To save codes and scripts.
  - **.RData** - To save workspace objects.
  - **.Rhistory** - To save the history of executed commands.

BASIC CONCEPTS

First steps

- Creating a script:
  - Click on “File” → “New File” → “R Script” to open a new script file.
- Writing and running code:
  - Type code in the script editor: “Hello World!”;
  - Highlight the lines you want to run, and press Ctrl+Enter (Windows) or Cmd+Enter (macOS) to execute the code in the console (or to run all the code, use the Run button at the top of the script editor).
- Saving your work:
  - You can save your R script by clicking “File” → “Save As” and giving your file a .R extension.

BASIC CONCEPTS

Some observations.

- Everything in R is an object.
  - Think of objects like containers that hold data or things you can work with. For example: numbers, words, list of things, group of numbers, functions...
- There are differences between uppercase characters and lowercase characters.
- Parentheses, square brackets and braces:
  - ( ): to group objects inside a function.
  - [ ]: to group functions inside other functions.
  - {}: to index objects inside other objects.
- Comments can be inserted after the # character.
- The dot (.) or underscore (_) symbols can be used, but not spaces.
- A cheatsheet providing a detailed explanation of some available functions can be found at `https://rstudio.github.io/cheatsheets/html/rstudio-ide.html`.

BASIC CONCEPTS

Calculator.

```
> 2+2 # sum
[1] 4
> 2-2 # subtraction
[1] 0
> 2*2 # multiplication
[1] 4
> 2/2 # division
[1] 1
> 2^2 # power
[1] 4
> (2+2^2)/2 # solution priority
[1] 3
```

BASIC CONCEPTS

Assigning.

To assign values to objects, just use the ← operator, which is the combination of the < operator with −. Alternatively, we can use the = operator.

```
> x <- 10 # the value 10 is saved in the object x
> y <- x + 10
```

The ← operator is preferred by many R users because it clearly distinguishes assignment from equality checking (==). The = operator can also be used for assignment, but it is more commonly used for setting function arguments. While it behaves similarly to ← when assigning variables, using ← is generally recommended for clarity and consistency in most R code.

BASIC CONCEPTS

Your turn.

Question 1: *Find the volume of a cylindrical water tank whose base radius is 25 inches and whose height is 120 inches. Use* $\pi = 3.14$.

Remember: $volume = \pi \times radius^2 \times height$.

BASIC CONCEPTS

Types of variables.

R has different classes to accommodate different types of data.

```
> x <- 4.5 # numeric
> x <- 4 # integer
> x <- "summer" # character
> x <- TRUE # logical
```

We can check the structure of any object by using the built-in `str()` function.

BASIC CONCEPTS

Logical operators.

Logical operators are binary operators for performing tests between two variables (objects). These operations return the value TRUE or FALSE.

```
# Logical operators

x <- 10 # assigning the value 10 to the object x
y <- 2 # assigning the value 2 to the object y

x < y # is x smaller than y?
x > y # is x greater than y?
x <= y # is x less than or equal to y?
x == y # is x equal to y?
x != y # is x different than y?
y == 2 | x == 2   # is x or y equal to 2?
x == 2 & y == 2  # are x and y equal to 2?
```

Basic Concepts

Your turn

Question 2: *You have three participants with scores of 85, 50, and 75. Use logical expressions (and, or, not) to answer the following:*

- *Did all participants score above 40?*
- *Did any participant score exactly 50?*
- *Is it true that none of the participants scored less than 30?*

BASIC CONCEPTS

Structures.

- Vectors: c( ).
  - Vectors are one-dimensional collections of data of the same type (e.g., all numbers or all characters).
  - You can create a vector using the c( ) function.
  - You can access elements of a vector using the square brackets [ ].

BASIC CONCEPTS

Structures.

- Vectors: c( ).
  - Vectors are one-dimensional collections of data of the same type (e.g., all numbers or all characters).
  - You can create a vector using the c( ) function.
  - You can access elements of a vector using the square brackets [ ].
- Matrix: matrix( )
  - Matrices are two-dimensional arrays that store elements of the same type (e.g., all numeric).
  - You can create a matrix using the matrix( ) function.
  - You can access elements of a matrix using the square brackets [ ].

BASIC CONCEPTS

Structures.

- Vectors: c( ).
  - Vectors are one-dimensional collections of data of the same type (e.g., all numbers or all characters).
  - You can create a vector using the c( ) function.
  - You can access elements of a vector using the square brackets [ ].
- Matrix: matrix( )
  - Matrices are two-dimensional arrays that store elements of the same type (e.g., all numeric).
  - You can create a matrix using the matrix( ) function.
  - You can access elements of a matrix using the square brackets [ ].
- Data frame: data.frame( )
  - Data frames are two-dimensional tables, similar to spreadsheets or SQL tables. Each column in a data frame can hold different data types (numeric, character, etc.).
  - You can create a data frame using the data.frame( ) function.

BASIC CONCEPTS

Structures.

- Vectors: c( ).
  - Vectors are one-dimensional collections of data of the same type (e.g., all numbers or all characters).
  - You can create a vector using the c( ) function.
  - You can access elements of a vector using the square brackets [ ].
- Matrix: matrix( )
  - Matrices are two-dimensional arrays that store elements of the same type (e.g., all numeric).
  - You can create a matrix using the matrix( ) function.
  - You can access elements of a matrix using the square brackets [ ].
- Data frame: data.frame( )
  - Data frames are two-dimensional tables, similar to spreadsheets or SQL tables. Each column in a data frame can hold different data types (numeric, character, etc.).
  - You can create a data frame using the data.frame( ) function.
- List: list( )
  - Lists can hold elements of different types, including numbers, strings, vectors, and even other lists.
  - You can create a list using the list( ) function.

BASIC CONCEPTS

Your turn.

Question 3: *You have three types of fertilisers: FertA, FertB, and FertC. Create a vector with these names. Then, check if the second element in the vector is "FertB".*

Question 4: *The crop yields (in tons per hectare) for wheat and corn are as follows:*

- *Wheat: 3.2 (Field A), 3.5 (Field B)*
- *Corn: 4.1 (Field A), 4.4 (Field B)*

Create a matrix with this data, and retrieve the yield of wheat in Field B.

Question 5: *Create a data frame with the following information about three fields. Then, retrieve the crop grown in Field C.*

- *Field: "Field A", "Field B", "Field C"*
- *Area (in hectares): 5, 10, 8*
- *Crop: "wheat", "corn", "soybean"*

BASIC CONCEPTS

Functions.

In R, a function is a block of code, which upon receiving data, called arguments, returns an object.

$$function(argument_1 = value_1, argument_2 = value_2, \ldots)$$

```
> sum(1,2,3,4) # sum of several values
[1] 10
> sqrt(36) # square root
[1] 6
> rep(x = 1, times = 10) # repeat 10 times the number 1
 [1] 1 1 1 1 1 1 1 1 1 1
> log(x = 10) # natural logarithm
[1] 2.302585
> log(x = 10, base = 2) # Logarithm of 10 to base 2
[1] 3.321928
> log(base = 2, x = 10) # changing the order of the arguments
[1] 3.321928
> log(2, 10) # the result is different if the arguments are not specif
[1] 0.30103
```

BASIC CONCEPTS

Your turn.

Question 6: *Create two numeric vectors in R. Using the length( ) function and the logical operators, answer if the length of the first vector is greater, smaller or equal to the second vector.*

BASIC CONCEPTS

Help function.

The help function (or ?) allows you to find the help file of the functions.

```
> help(sum) # Open the log function help
> ?sum
> help.search("sum") # Search for the term sum
> ??sum
```

BASIC CONCEPTS

Writing your own function.

In addition to using R's built-in functions, you can write your own custom functions to perform specific tasks. Writing your own function allows you to create reusable blocks of code for operations you might need frequently.

```
function_name <- function(arguments) {
  # Code to execute
  return(result)
}
```

BASIC CONCEPTS

Your turn.

Question 7: *In agronomy, crop yield is often measured in tons per hectare. However, some researchers need the yield in kilograms per hectare for specific analyses. Create a function in R called* `convert_to_kg` *that:*

- *Takes one argument:* `yield_tons` *(the yield in tons per hectare).*
- *Converts it to kilograms per hectare (1 ton = 1,000 kilograms).*
- *Returns the converted value.*

BASIC CONCEPTS

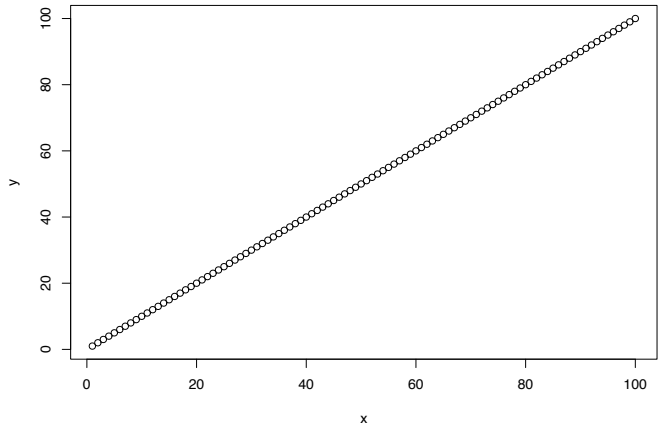
Packages.

- R base and packages.

  A collection of functions that can be written in different programming languages that are called directly from within R. A package contains code, data and documentation.

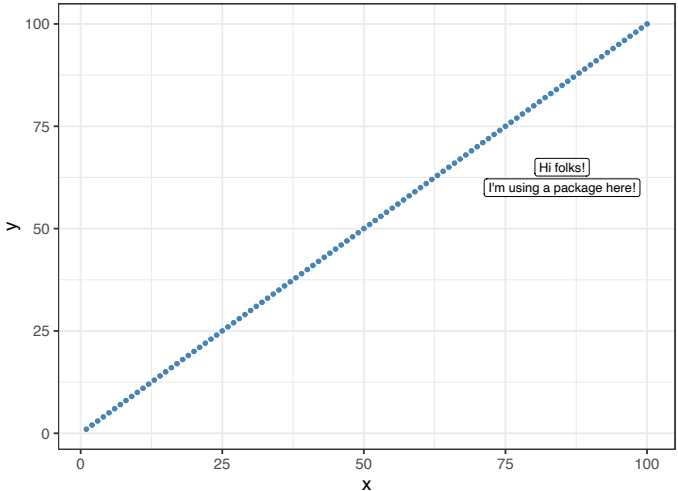
BASIC CONCEPTS

R base and packages.



BASIC CONCEPTS

R base and packages.



BASIC CONCEPTS

Your turn.

Question 8: *Install the following packages and look at the vignettes:*

- *dplyr*
- *tidyr*
- *lme4*

Thank you!