Introduction to R: Part I

Luz Sanchez Steinhagen

20/08/2025

Credit statement and licence

Possible roles using the CRediT contribition system:

- **Conceptualization**: Ideas; formulation or evolution of overarching research goals and aims
- **Methodology**: Development or design of methodology; creation of models
- **Software** : Programming, software development; designing computer programs; implementation of the computer code and supporting algorithms; testing of existing code components
- **Validation**: Verification, whether as a part of the activity or separate, of the overall replication/ reproducibility of results/experiments and other research outputs
- **Formal analysis**: Application of statistical, mathematical, computational, or other formal techniques to analyze or synthesize study data
- **Investigation**: Conducting a research and investigation process, specifically performing the experiments, or data/evidence collection
- **Resources**: Provision of study materials, reagents, materials, patients, laboratory samples, animals, instrumentation, computing resources, or other analysis tools
- **Data Curation**: Management activities to annotate (produce metadata), scrub data and maintain research data (including software code, where it is necessary for interpreting the data itself) for initial use and later reuse
- **Writing - Original Draft**: Preparation, creation and/or presentation of the published work, specifically writing the initial draft (including substantive translation)
- **Writing - Review & Editing**: Preparation, creation and/or presentation of the published work by those from the original research group, specifically critical review, commentary or revision – including pre-or postpublication stages
- **Visualization**: Preparation, creation and/or presentation of the published work, specifically visualization/ data presentation

- **Supervision**: Oversight and leadership responsibility for the research activity planning and execution, including mentorship external to the core team
- **Project administration**: Management and coordination responsibility for the research activity planning and execution
- **Funding acquisition**: Acquisition of the financial support for the project leading to this publication

---

Prerequisites TO DO

> ! Prerequisites
>
> Before completing this submodule, please carefully read about the necessary prerequisites.

Prerequisite	Description	Link/Where to find it
Topic Name	Basic intro to X	Module + Submodule
Software Name	Configuring the environment	Download Link

These are the **speaker notes**. You will a script for the presenter for every slide. In presentation mode, your audience will not be able to see these speaker notes, they are only visible to the presenter.

There are also **instructor notes**. For some slides, there will be pedagogical tips, suggestons for acitivities and troubleshooting tips for issues your audience might run into. You can find these notes underneath the speaker notes.

Questions from previous submodule

- **Aim**: This first slide is dedicated to clarifying questions from the previous submodule and/or to discuss assignments.
- Additional slides may need to be added depending on the nature of the homework assignments.
- Critical for the learning process to ensure that students are on the same page and have been able to achieve the learning goals of the previous workshop.
- Not applicable if this set of slides corresponds to the first submodule of a new module.

Welcome/What to Expect

- Overview of what will happen (**make this slide in the end**)
- mention they should already have R and RStudio installed
- outline the structure and general idea (hands-on, follow-along etc…)

What is R?

- Knowledge of statistical software like R is crucial when conducting quantitative research
- R is free and open source
  - large online community
  - many free to use resources

Let's start with the basics: What exactly is R?

R is a programming language specifically designed for data analysis and statistics. That means it's not just a tool — it's a full language, built with researchers and data in mind.

It's also free and open source, which means anyone can download and use it, and developers from all over the world contribute to improving it. This has resulted in a large online community — so if you ever run into a problem, chances are someone else has had the same issue and posted a solution online.

And because of that community, there are also countless free learning resources: tutorials, blog posts, videos, books, and packages that extend R's functionality.

So in short: R is powerful, flexible, and supported by a strong community — which is why it's become such a widely used tool in quantitative research.

The RStudio Environment

RStudio is the main program we'll be using to interact with R throughout this course. It's a free and open-source editor specifically designed for R, and it makes things much easier — both for writing code and for organizing your work.

On this slide, you can see the four main panes of the RStudio interface:

The top left pane is where your scripts or other files appear. This is where you'll usually write and edit your code.

The top right is the Environment pane, which shows the objects you've created — like datasets or functions — so you can always check what's currently in your workspace.

The bottom left is the Console, where R actually runs your commands. If you type directly into R instead of using a script, this is where it happens.

And finally, the bottom right is for Help, Plots, and Packages. So if you ask for help on a function, create a plot, or install a new package, it'll show up here.

So, RStudio combines everything in one interface to help you keep track of your code, your results, and your progress as you go.

Exercise 1

Open RStudio and check if you can see everything that's on the slide (might be only three panes)

Instructor notes: add potential troubleshooting tips for instructors to efficiently solve the problems at this stage.

Mathematical Operations

Symbol	Operation
`+`	Addition
`-`	Subtraction
`*`	Multiplication
`/`	Division
`^`	Exponentiation
`sqrt()`	Square Root
`log()`	(Natural) Logarithm

In the bottom left pane of RStudio - the console - we can run commands. For example, we can use R to execute calculations. To do this, we use operators that correspond to mathematical symbols, such as the + and - sign.

Here, you can see a table with the most important mathematical operators in R.

Exercise 2 - Mathematical Operations

Use the table on the previous slide to run the following math operations directly in the console (bottom left).

- $12 + 17$
- $\frac{218}{3}$
- 357^2

Hint: You can run code from the console by pressing Enter.

We've already arrived at our second exercise. Use the console in RStudio to execute the mathematical operations you see on this slide.

optional: live demonstration of what it looks like to run code from the console

---

Exercise 2 - Solution

```
12+17
218/3
357^2
```

Great, now you have a first impression of what it's like to use R!

Logical Comparisons

- Check if two numbers are equal or unequal
- Check if a number is larger or smaller than another number

Example:

```
49 == 50
```

```
[1] FALSE
```

```
49 < 50
```

```
[1] TRUE
```

- R returns a **logical value**:
  - `TRUE` or
  - `FALSE`

However, R can of course do more than just calculate. One of its basic features, that is needed for many of its more complex functions are logical comparisons.

We can compare two numbers to determine if they are equal or unequal and check if a number is larger or smaller than another number. Comparisons like that are answered by R with either `TRUE`(yes) or `FALSE` (no). These comparisons are called **logical comparisons**. We can also combine logical comparisons using the logical AND, OR, and NOT to formulate longer statements.

Logical Comparisons - Operators

Symbol	Operation
`==`	Equals
`!=`	Not Equal
`>` and `<`	Greater and Lesser Than
`>=` and `<=`	Greater Than or Equal / Less Than or Equal
`&`	logical AND
`|`	logical OR
`!`	logical NOT

Here you can see an overview of logical operators. You can combine several single statements to longer statements using the logical AND, OR, and NOT.

Exercise 3: Logical Comparisons

`TRUE` or `FALSE`?

Express the following statements in words and evaluate whether they are correct or not. Check your answers by running the statements in the console.

```r
(36 + 5)> 40
```

```r
(4 == 5) | (25 > 17)
```

```r
(5 == (2 + 3)) & (7 >= 8)
```

TODO: decide whether there should be another exercise that involves writing code to express a written statement

---

Exercise 3: Solutions

```r
(36 + 5)> 40
```

```
[1] TRUE
```

```r
(4 == 5) | (25 > 17)
```

```
[1] TRUE
```

```r
(5 == (2 + 3)) & (7 >= 8)
```

```
[1] FALSE
```

Important Data Types

In R, different data types exist. We have already got to know the logical values `TRUE`and `FALSE`. Data that are numbers can be further differentiated into numeric values, also known as `double`, and integers. Numeric variables have decimals, integers do not. Values that are words are normally either characters or factors. Characters have to be written in quotation marks. Factors have prespecified levels, such as brown, blonde, and black for a variable "hair color".

Depending on which type a variable has, different operations can be performed with it.

Data Types - Operations

Two numeric variables can be summed up:

```
4 + 5
```

```
[1] 9
```

A numeric and a character variable cannot be summed up:

```
4 + "five"
```

```
Error in 4 + "five": non-numeric argument to binary operator
```

R Scripts

- Easy way to save and share code you wrote
- You can run code from an R script by pressing Ctrl + Enter (Windows) or
- To comment your code use `#`, for example:

```
# sum up two numbers:
4+5
```

```
[1] 9
```

Until now, we've only worked in the console. However, once we close R, everything typed into the console will be lost. To make our work traceable and reproducible, it makes sense to save the code we used. One way of saving and sharing code are R scripts. To run code from an R script, put the cursor anywhere in the respective line of code or mark everything you want to execute. Then press Ctrl+Enter (Windows) or + Enter (Mac).

If you want to take notes or write anything that is not code into an R script, type # and then type your text.

Creating a New R Script

This is how you create a new R script. Navigate to the File tab –> New File –> R Script or use the shortcut Ctrl + Shift + N. Then you can save it by navigating to File –> Save or use the shortcut Ctrl + S.

Exercise 4

Create a new R Script with only a comment in the first line and save it in a fresh folder.

Objects and Assignments

- Everything in R is an object
- Oftentimes, we want to work with an object repeatedly
- Assign a name to an object using the assignment arrow `<-`

For Example:

```
my_number <- 4
```

- This saves the value 4 under the name `my_number`in the environment of R Studio

Everything in R is referred to as an object. For example, the result of a logical comparison, is an object of type logical. Oftentimes, we want to work with an object repeatedly. To do this, we can create objects and give them names. This is done via assignments.

TODO:

maybe include instructions on a live demonstration? so that participants can see the object "my_number" appear in the environment pane.

Assignments and Objects

```
my_number <- 5
```

Vectors, Lists, and Data Frame

Indexing

Relevance and implications

- **Aim**: To work out the relevance of the topic to your students.
- In an interactive setting, discuss how the new skills could be applied in practise with specific examples.
- Examine downfalls and practical obstacles.

Take-home message

Aim: End lesson on clear take-home message that are interactively compiled by students.

> **Tip with Title**
>
> Add one practical tips or take-home message.

Assignment

- **Aim**: Explain the homework assignment and the rationale behind the homework.
- Examine whether/how it will be assessed
- Mention scoring rubrics, if applicable
- Design a peer-review system for assignments to place students in role of reviewer and author

---

To conclude: Survey time!

- **Aim**: This post-submodule survey serves to examine students' current knowledge about the sumodule's topic.
- Use free survey software such as or other survey software (particify, formR) to establish the following questions (shown on separate slides):

What is your level of familiarity with [Topic] (e.g., basic concepts, terminology, or tools)?

a. I have never heard of it before.

b. I have heard of it but have never worked with it.

c. I have basic understanding and experience with it.

d. I am very familiar and have worked with it extensively.

Which of the following concepts or skills do you feel most confident about in relation to [Topic]? (Select all that apply)

a. Concept 1

b. Concept 2

c. Concept 3

d. Concept 4

e. I am not sure about any of these concepts.

On a scale of 1 to 5, how comfortable are you with using [specific tool/technology] related to [Topic]? (1 = Not comfortable at all, 5 = Very comfortable)

a. 1

b. 2

c. 3

d. 4

e. 5

Discussion of survey results

- **Aim**: Briefly examine the answers given to each question interactively with the group.
- Compare and highlight specific differences in answers between pre- and post-survey answers

---

References

- Provide literature you refer to throughout this lesson.

Thanks!

See you next class :)
