Neuro-SysMed

8 Dec 2025

Introduction to R programming

NEUROSYSM930

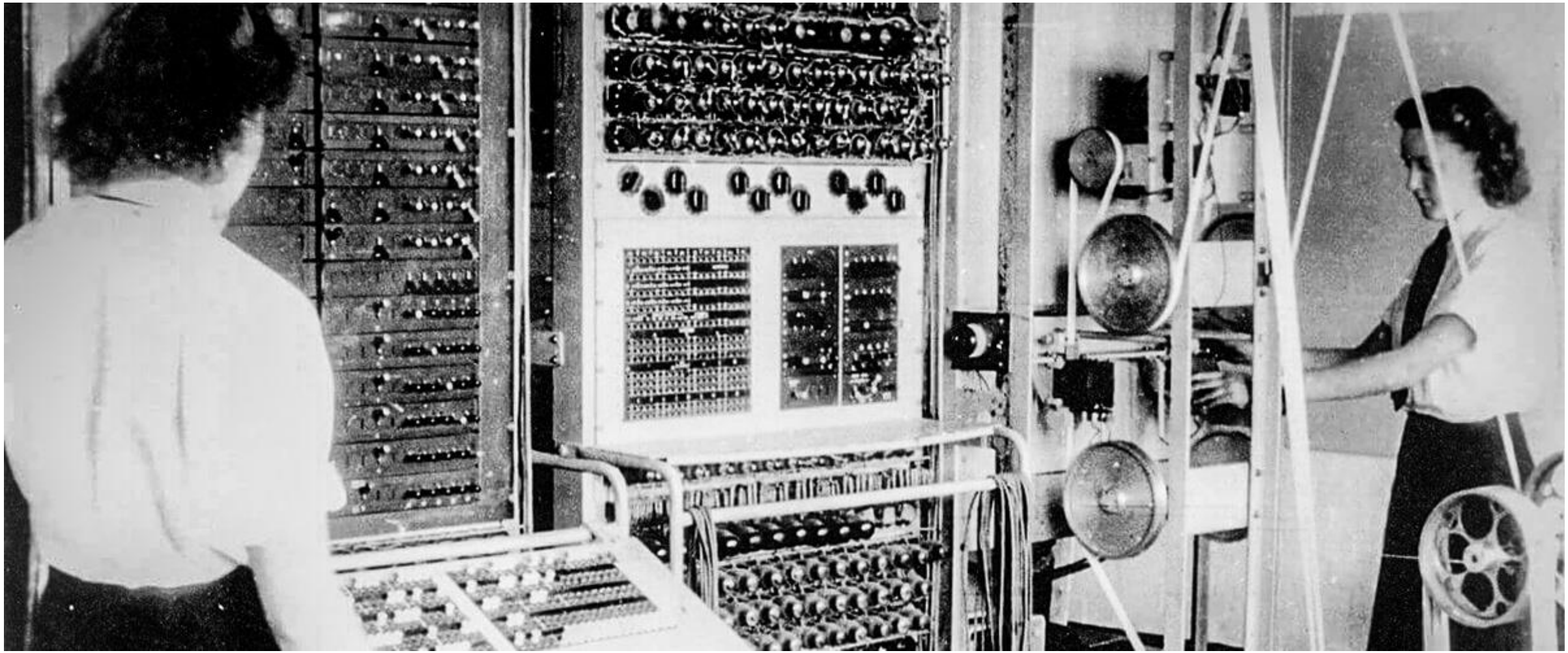
Dimitrios Kleftogiannis

dimitrios.kleftogiannis@uib.no

Centres for Clinical Treatment Research

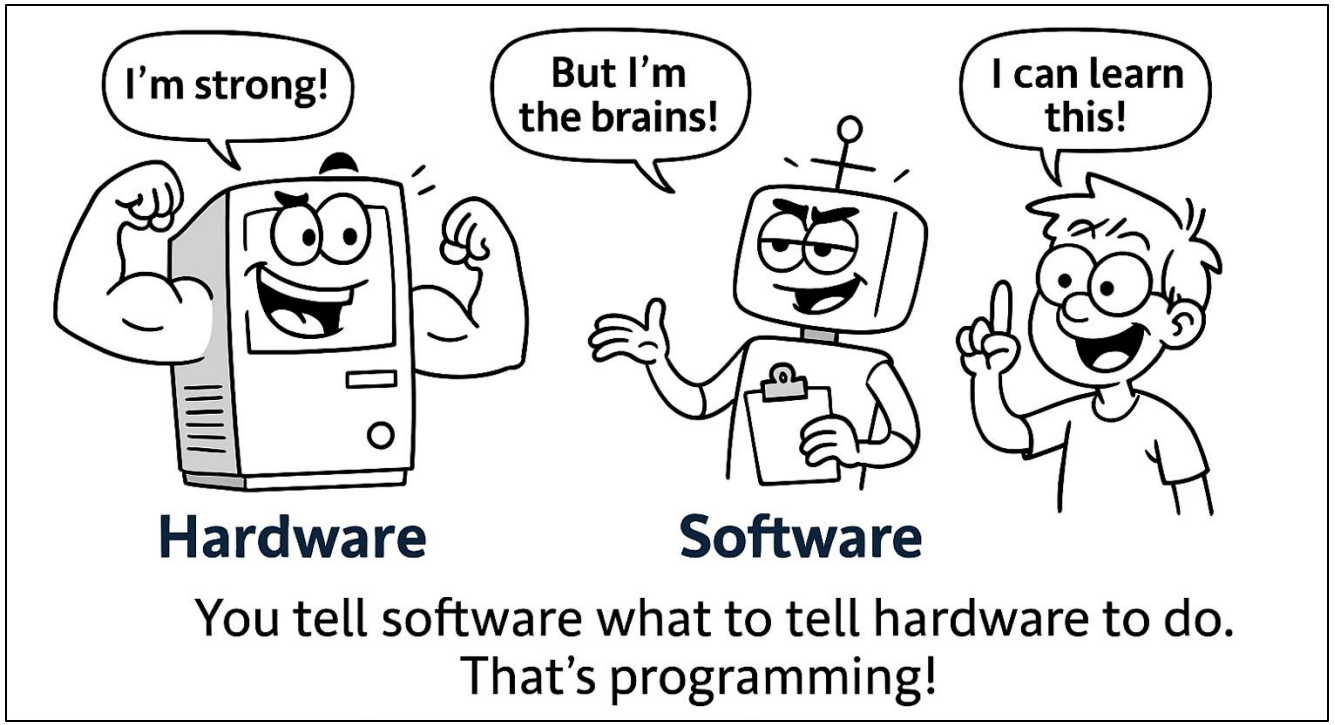
The Research Council of Norway

Computers and Programming

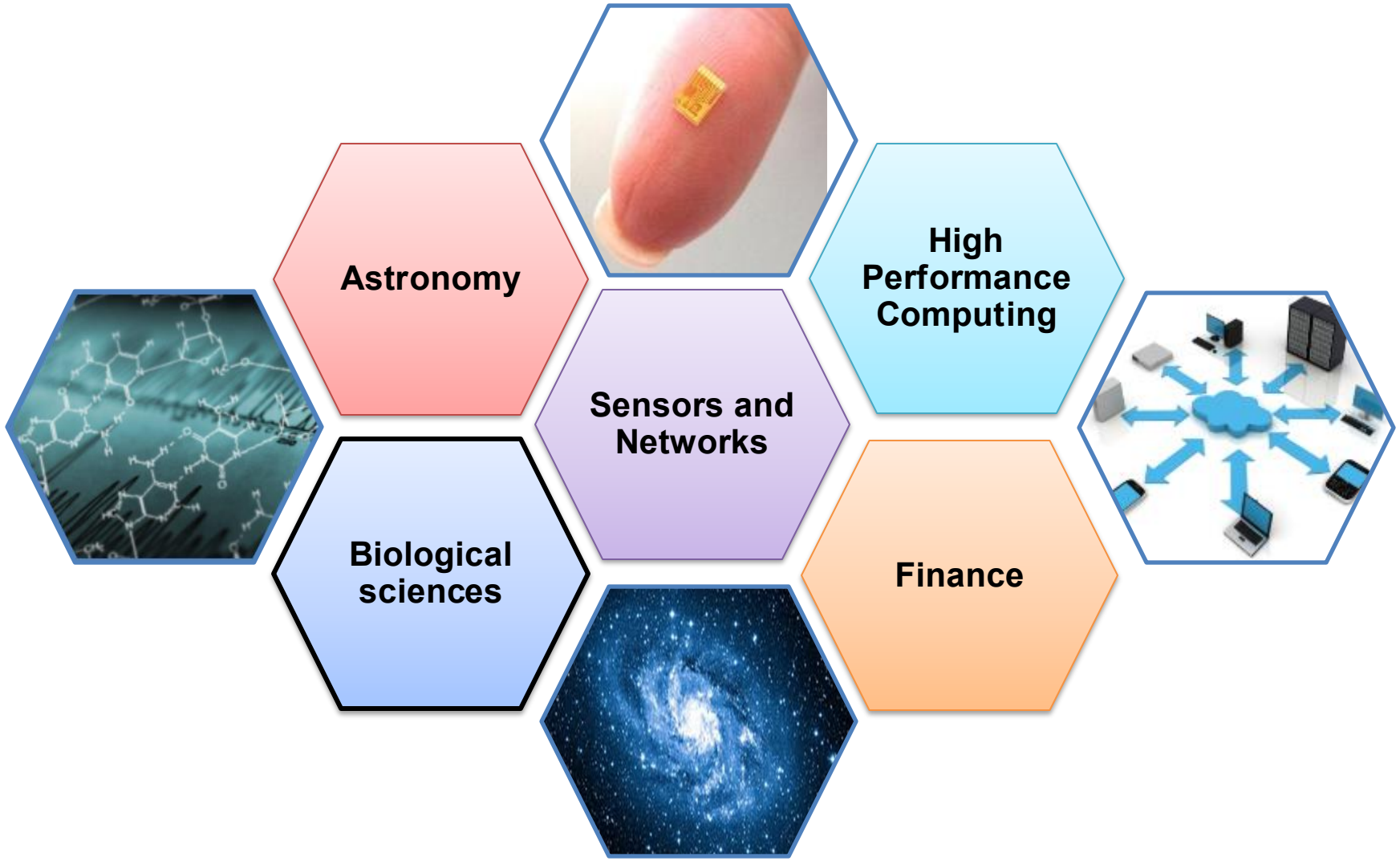


Computers and Programming

- A computer is just a machine (***hardware***) for executing programs (***software***)
- The process of creating a program is called ***programming***, and it is the focus of this course
- **Anyone can learn programming**



Computers and programming are part of our life



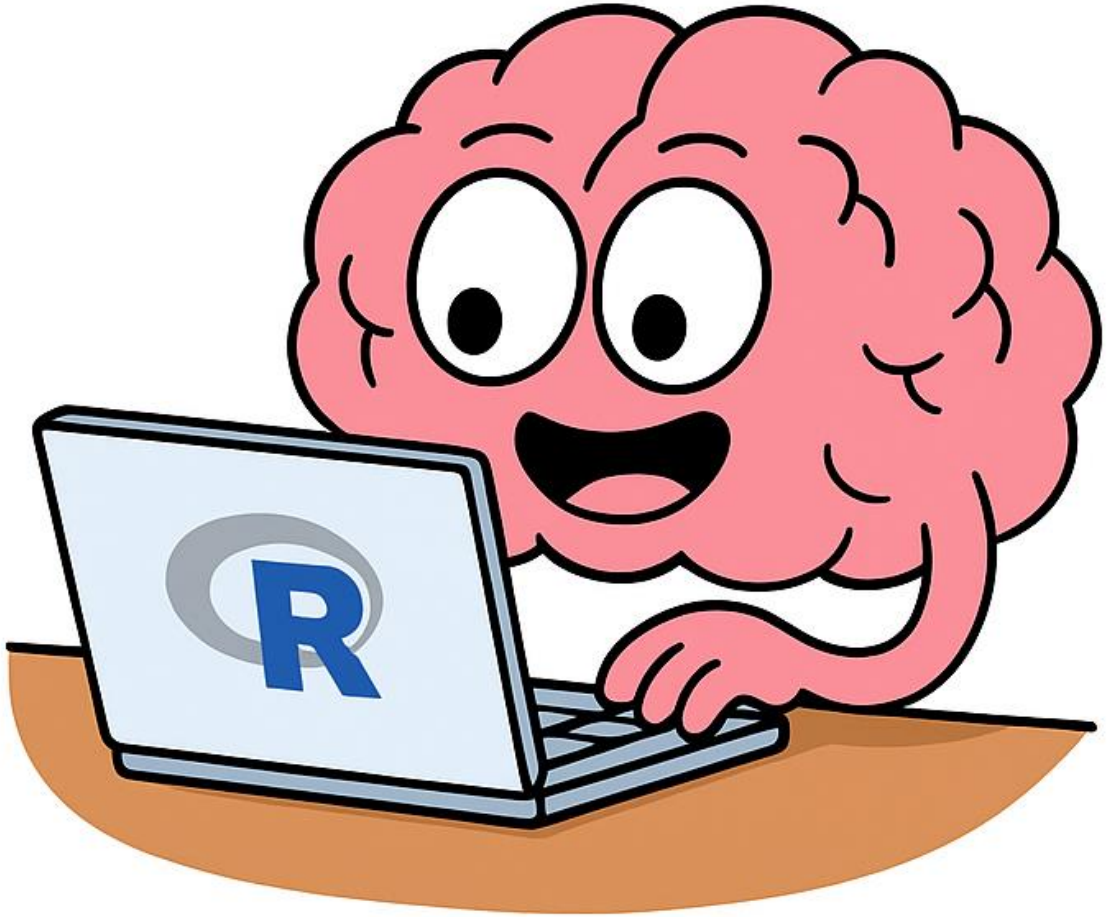
Why to learn programming in clinical research ?

Perform patient cohort/
data exploration

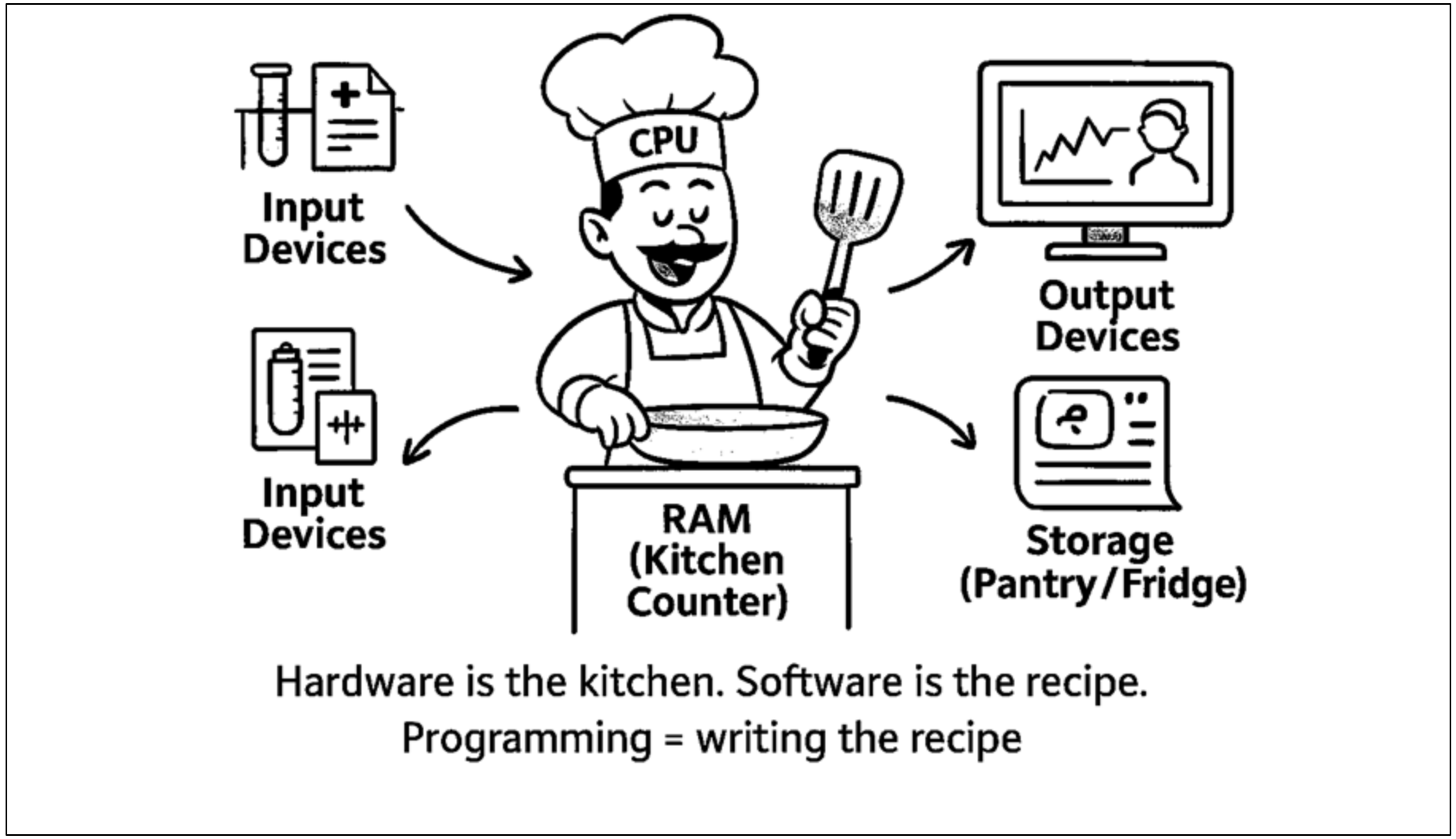
Compute p-values and
other statistics

High-throughput
data analysis

Reproducible research



Computers explained simply

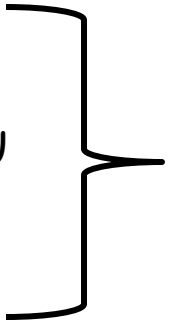


Programming languages

- The hardware understands a **very low-level language** known as machine language which is translated to binary….

For example:

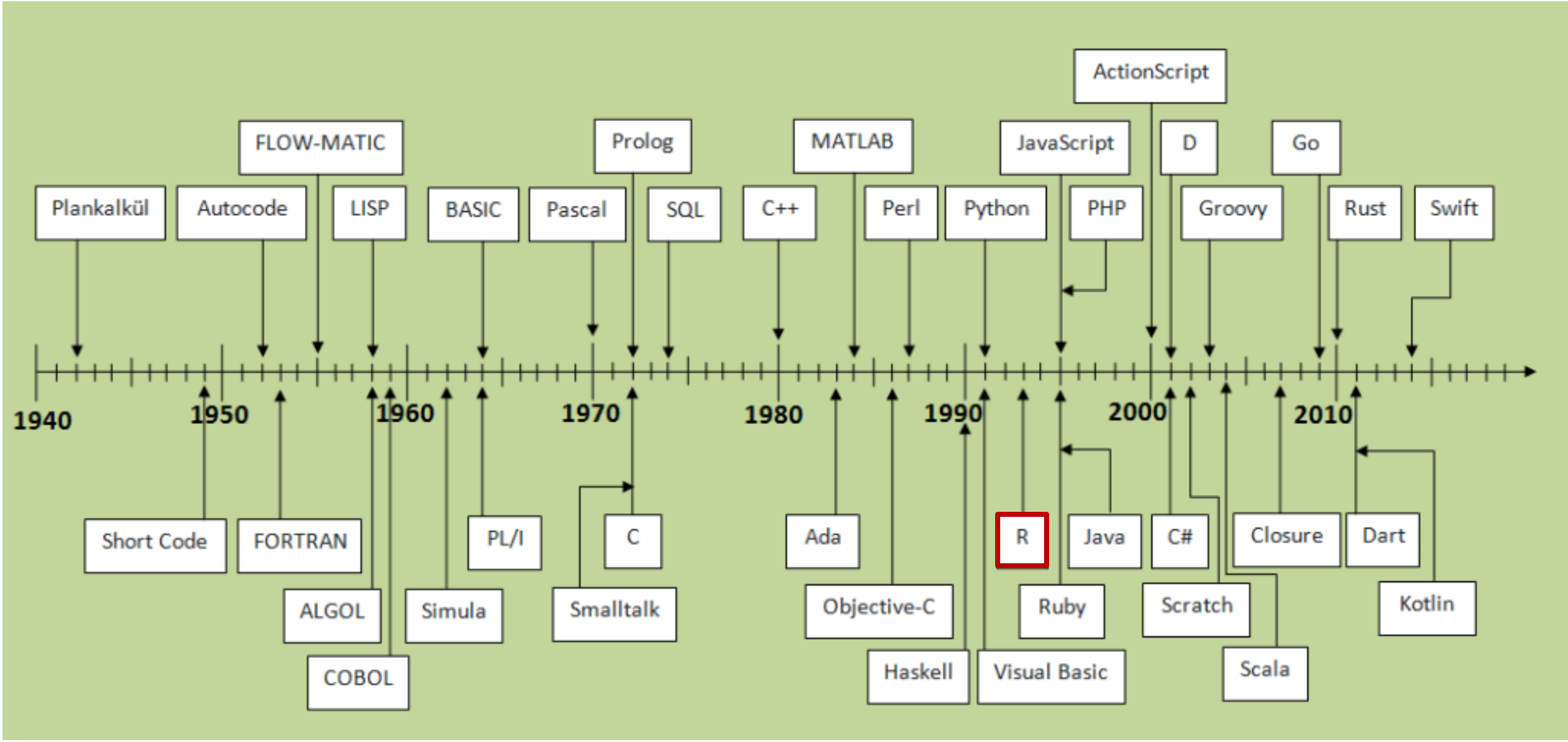
1. Load the number from memory location 2001 into the CPU
2. Load the number from memory location 2002 into the CPU
3. Add the two numbers in the CPU
4. Store the result into location 2003



A + B = C

- We need to use a **higher-level language and convert it** into a machine language that the hardware can execute…
  - **Compiled languages** → use a complex program called compiler
  - **Interpreted languages** → executes the code on the fly

Timeline of Programming Languages



The brief history of R

"R is a programming language for statistical computing and data visualization. It has been adopted in the fields of data mining, bioinformatics, and data analysis" (from wiki)

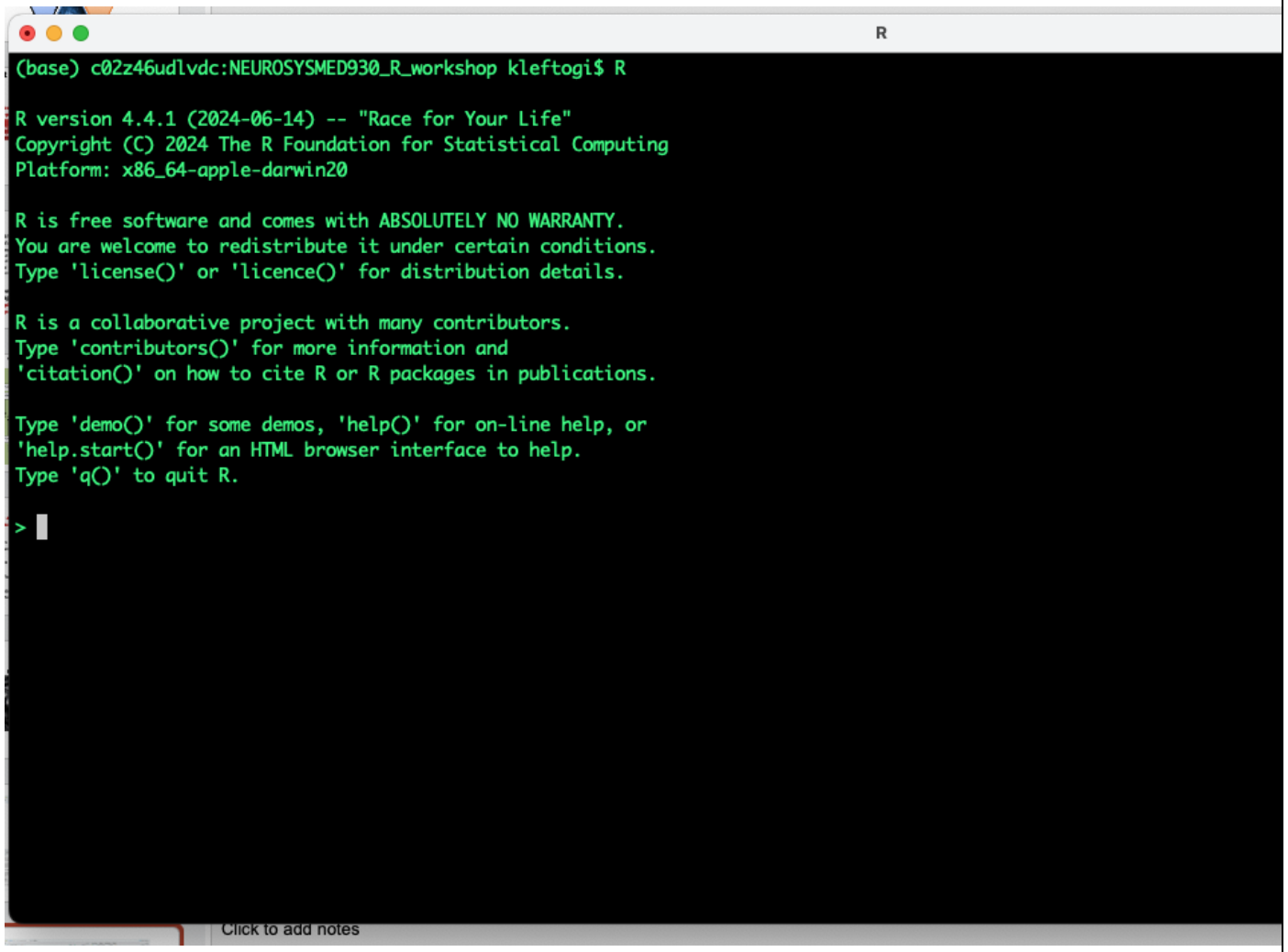
- R was developed in 1993 by **R**oss Ihaka and **R**obert Gentleman at the University of Auckland
- It was based on the S statistical programming language
- R software is **open-source and free** software licensed by the GNU Project
- In 1998, the Comprehensive R Archive Network **CRAN** was established
  - ✓ Now CRAN has more than 22,000 packages

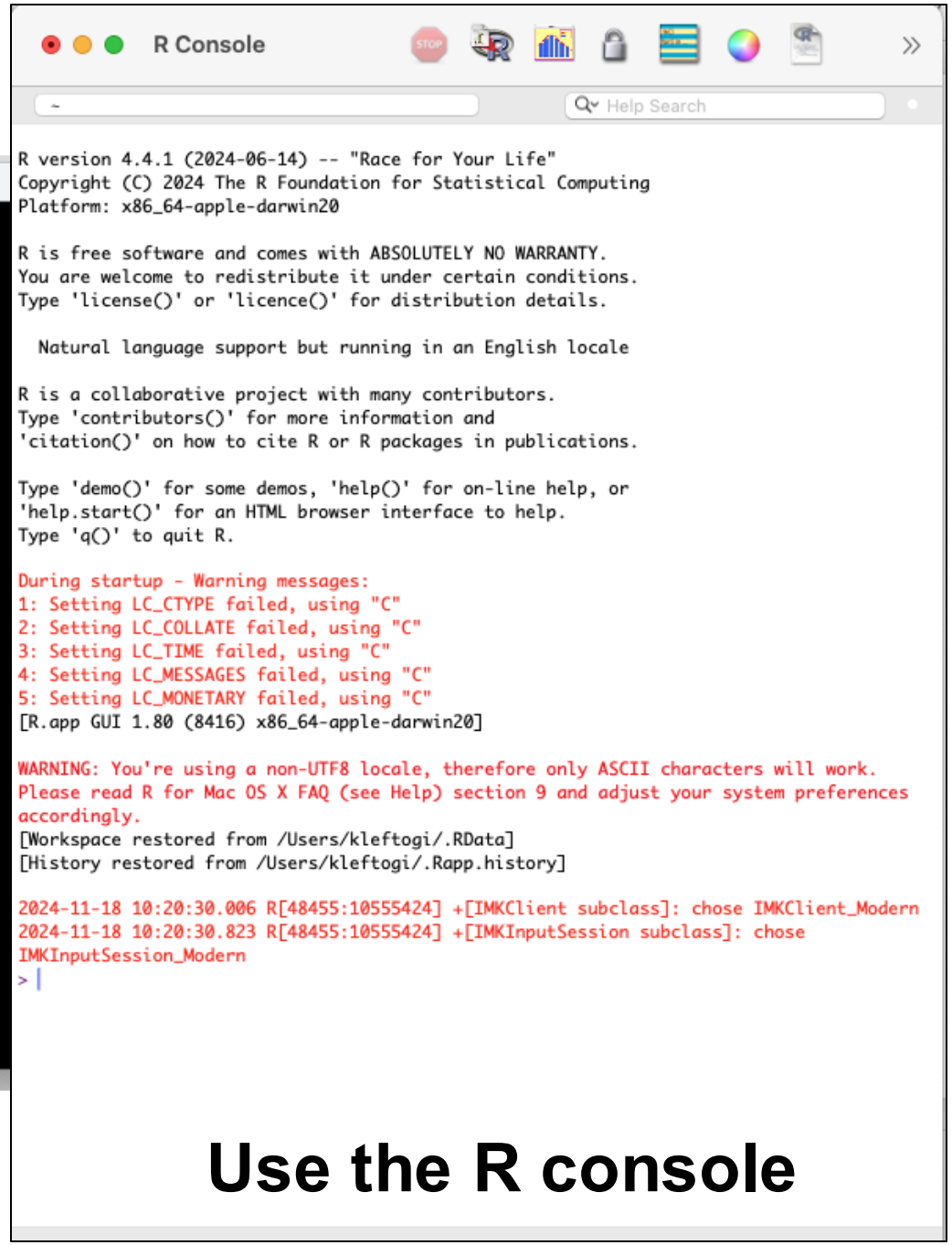
Explore the R architecture

R is the engine

RStudio is the dashboard

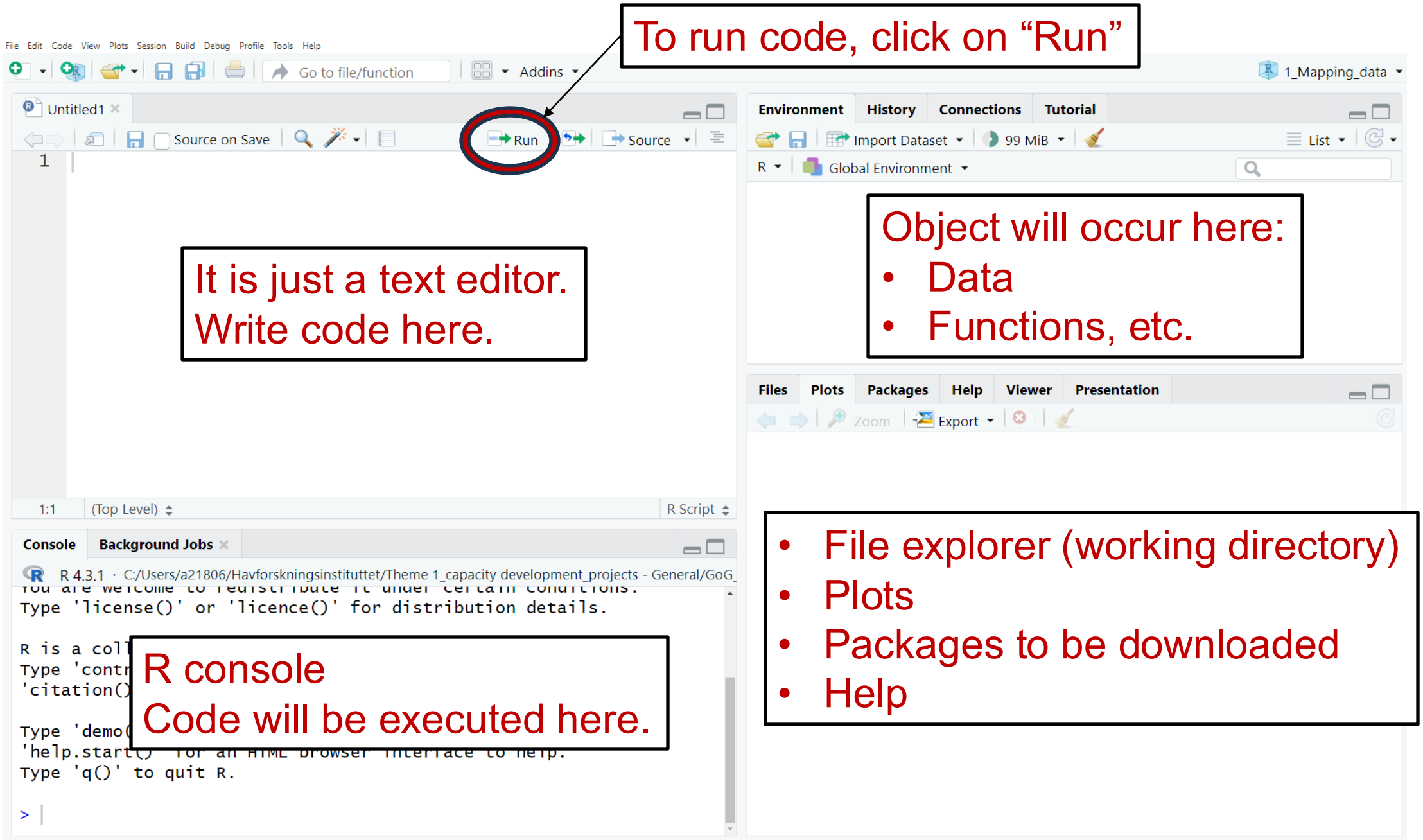
Use R from the terminal





Use the R console

The R studio environment



What shall I do now ?

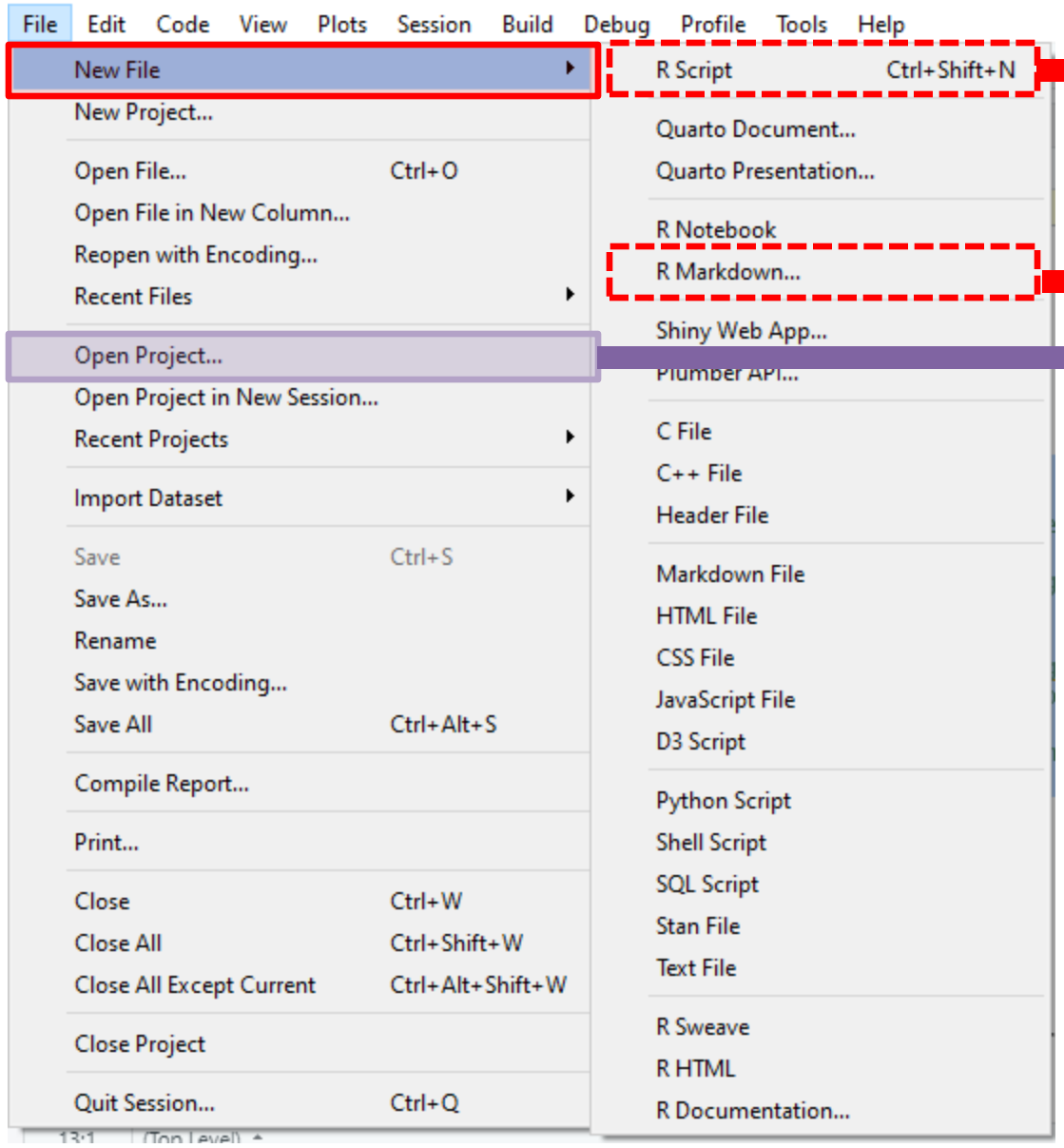
- As any other programming language R has its own syntax and semantics
- No buttons and clicking to perform computations…
- **But you need to learn how to program and you must invest time on that !**

  Because…

  - ✓ **You can do basically everything you need in your daily data wrangling life with only one software solution…**
  - ✓ **Statistics and machine learning**
  - ✓ **Small and big data exploration**
  - ✓ **Great visualization capabilities**
  - ✓ **Develop web-apps even databases and more complex workflows**

- R packages are essentially free R programs that provide extra functionalities
  - ✓ Need to be downloaded and installed

Different ways to create code



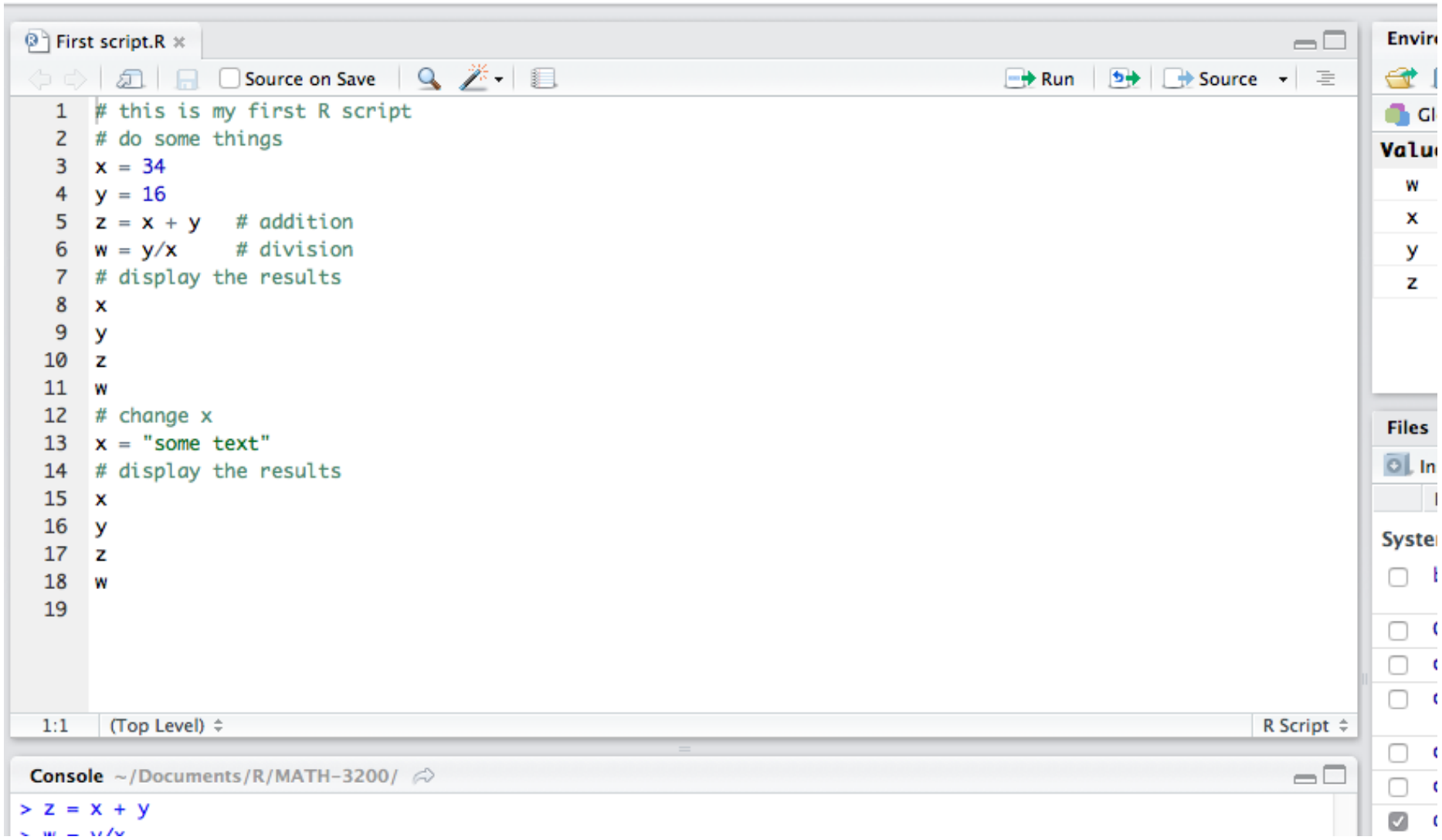
New text editor (to write code)

Another type of text editor great for data presentation

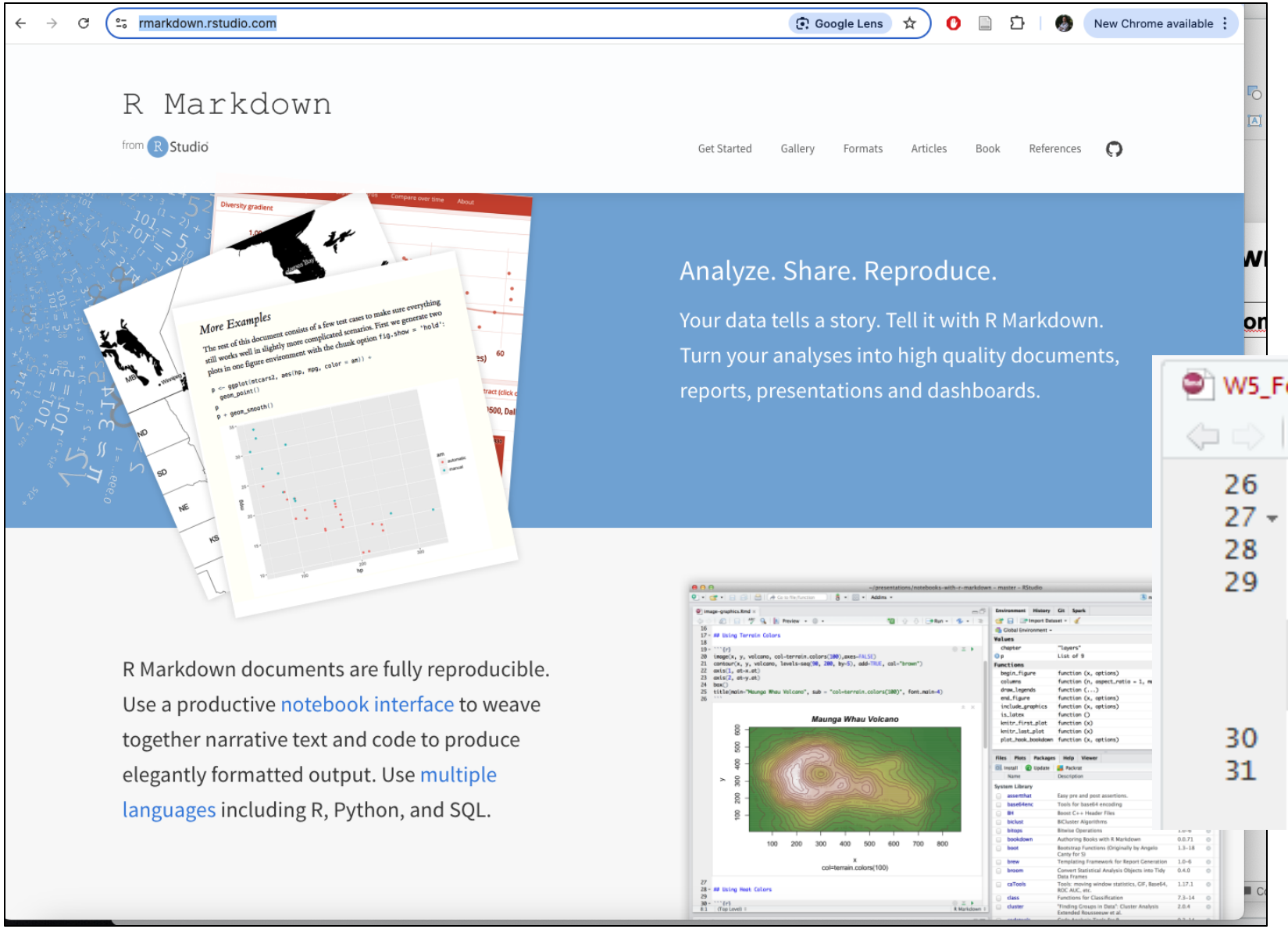
Includes several files and generates a nice working space that:

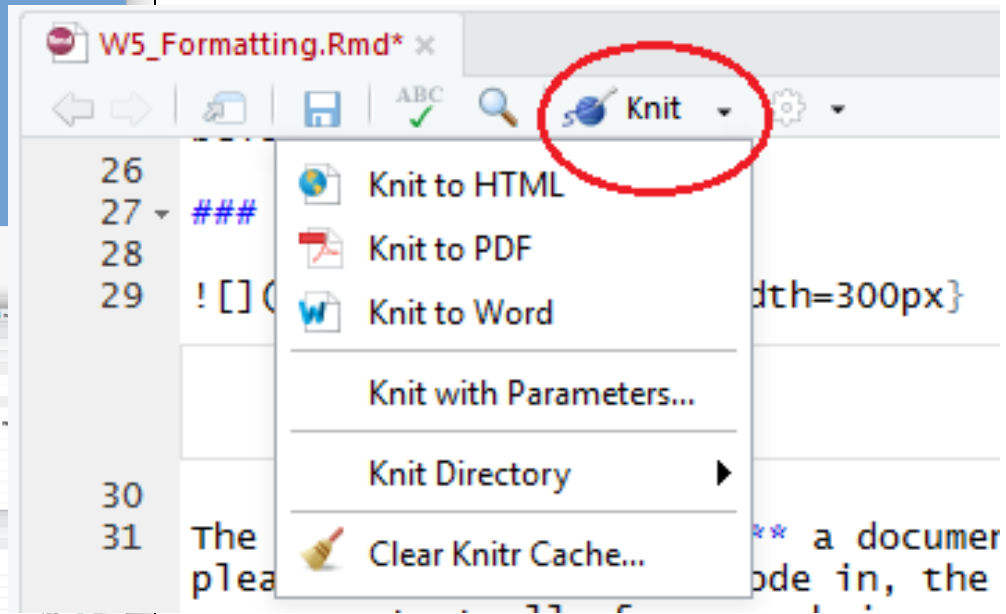
- ✓ You can copy the folder anywhere
- ✓ You can share the whole folder with collogues

R Scripts vs. R Markdowns



R Scripts vs. R Markdowns





Basic R concepts and terminology

- **Console** → where you enter in commands
- **Run** → the act of telling R to perform a computation by giving it commands in the console
- **Objects** → where values are saved in R, check the **Environment panel**
- **Data types** → integers, doubles/numerics, logicals, and characters.

Errors, warnings and messages

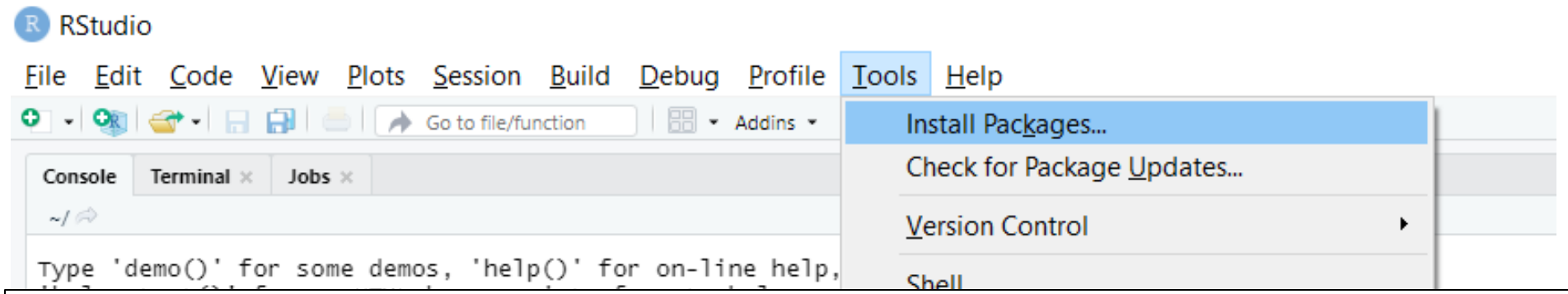
```
> dat <- read.csv("data.csv")
Error in fi
In additior
In file(fil
  cannot op
>
```

```
1:1  (Top Level)                                   R Script
Console  Terminal  Jobs
~/
> library("dplyr")

Attaching p

The followi

    between

The followi

    filter,

The followi

    interse

Warning message:
package 'dplyr' was built under R version 3.6.2
```


```
oup"
```

Packages

- Different ways to install packages exist



Install Bioconductor Packages

To install core packages, type the following in an R command window:

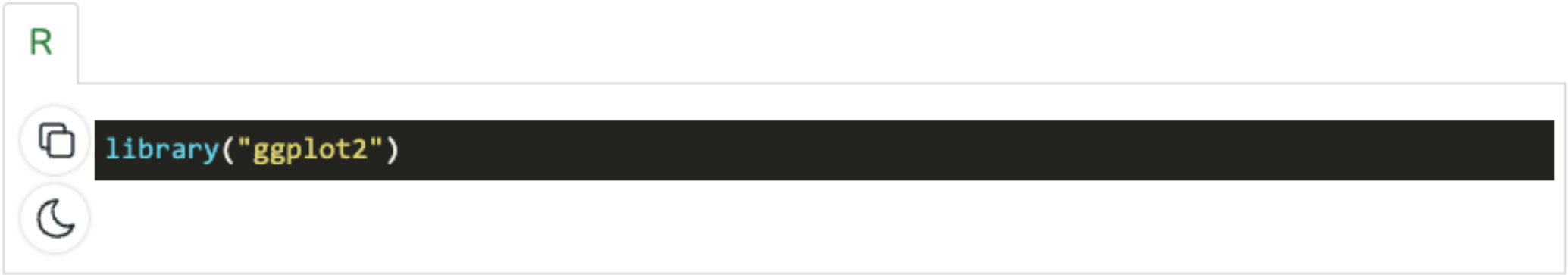
```
if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install(version = "3.20")
```

To install core packages, type the following in an R command window:

```
BiocManager::install(c("GenomicFeatures", "AnnotationDbi"))
```

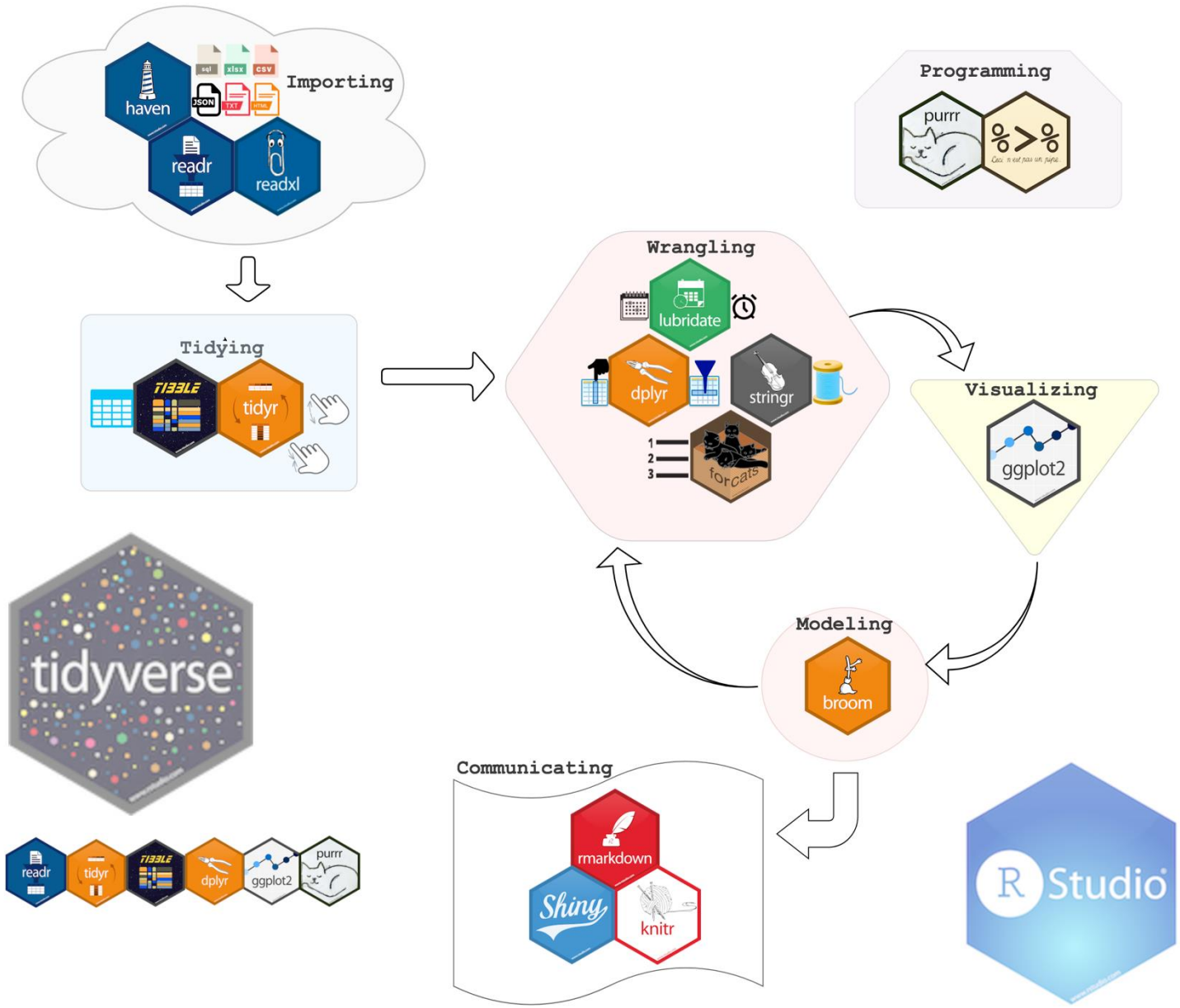
Packages

- Never forget to load the installed packages….



And never forget that R is distributed with fifteen "base packages": *base, compiler, datasets, grDevices, graphics, grid, methods, parallel, splines, stats, stats4, tcltk, tools, translations, and utils*

By combining R packages, we create workflows



Take home message

Nobody expects you to become an advanced programmer just because you attended this course.

You need to invest time on programming.

So practice, practice and practice…

During the practical session, we will just illustrate few out of many things you can do with R

Thanks for your attention !

Q&A

dimitrios.kleftogiannis@uib.no

https://www.uib.no/en/persons/Dimitrios.Kleftogiannis