



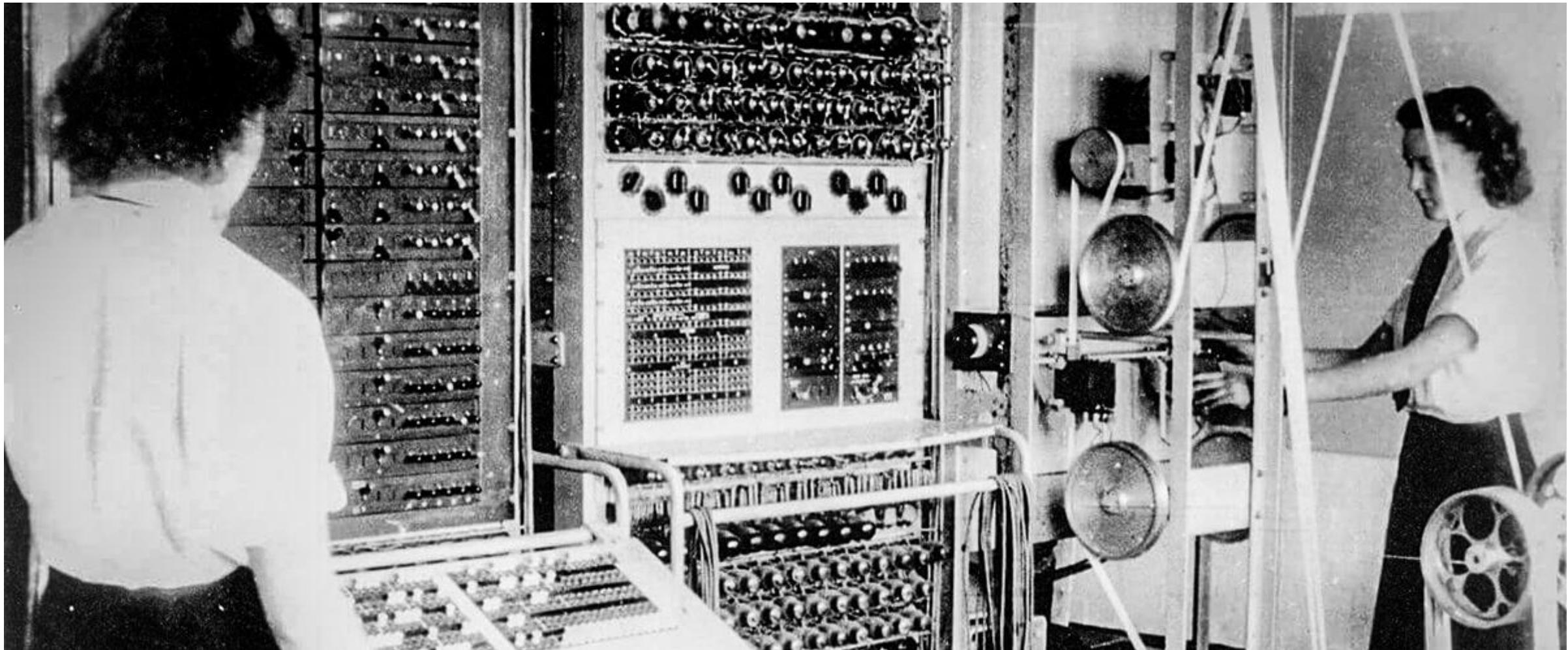
Introduction to R programming

NEUROSYSM930

Dimitrios Kleftogiannis

dimitrios.kleftogiannis@uib.no

Computers and Programming



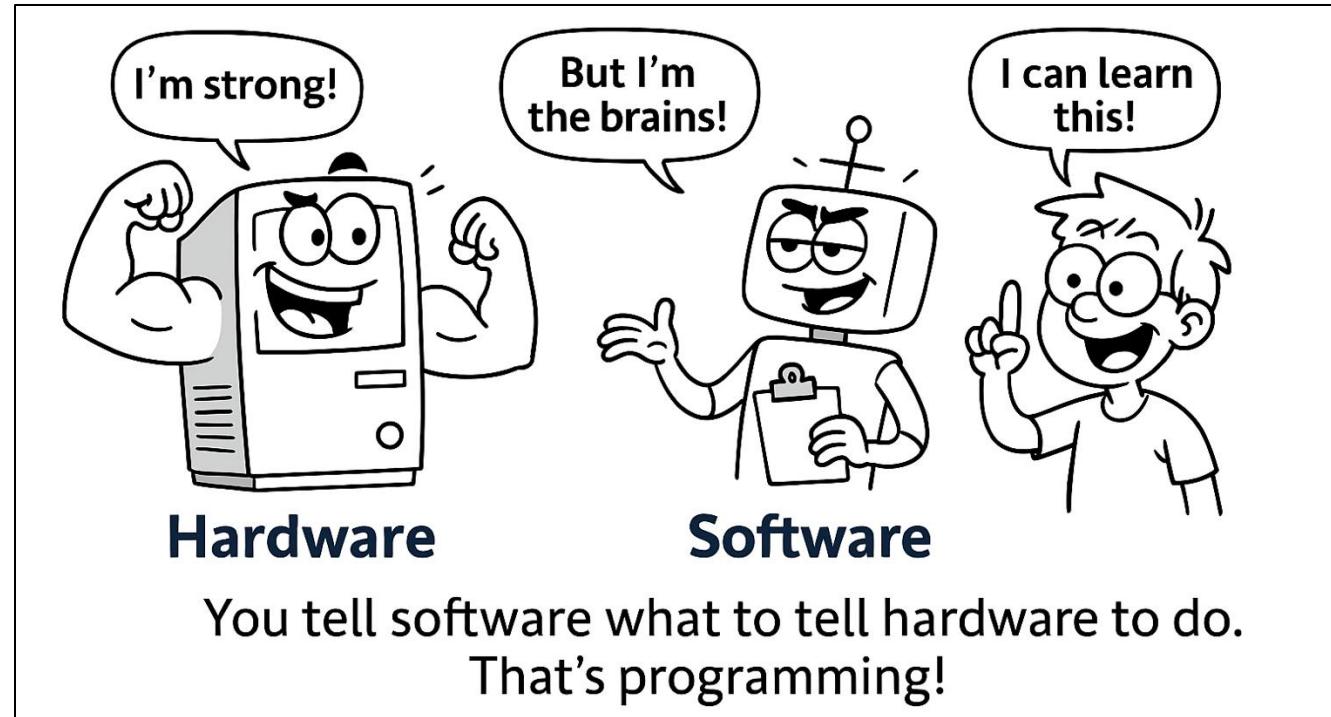
(c) Image taken from the www.
All rights reserved.

Computers and Programming

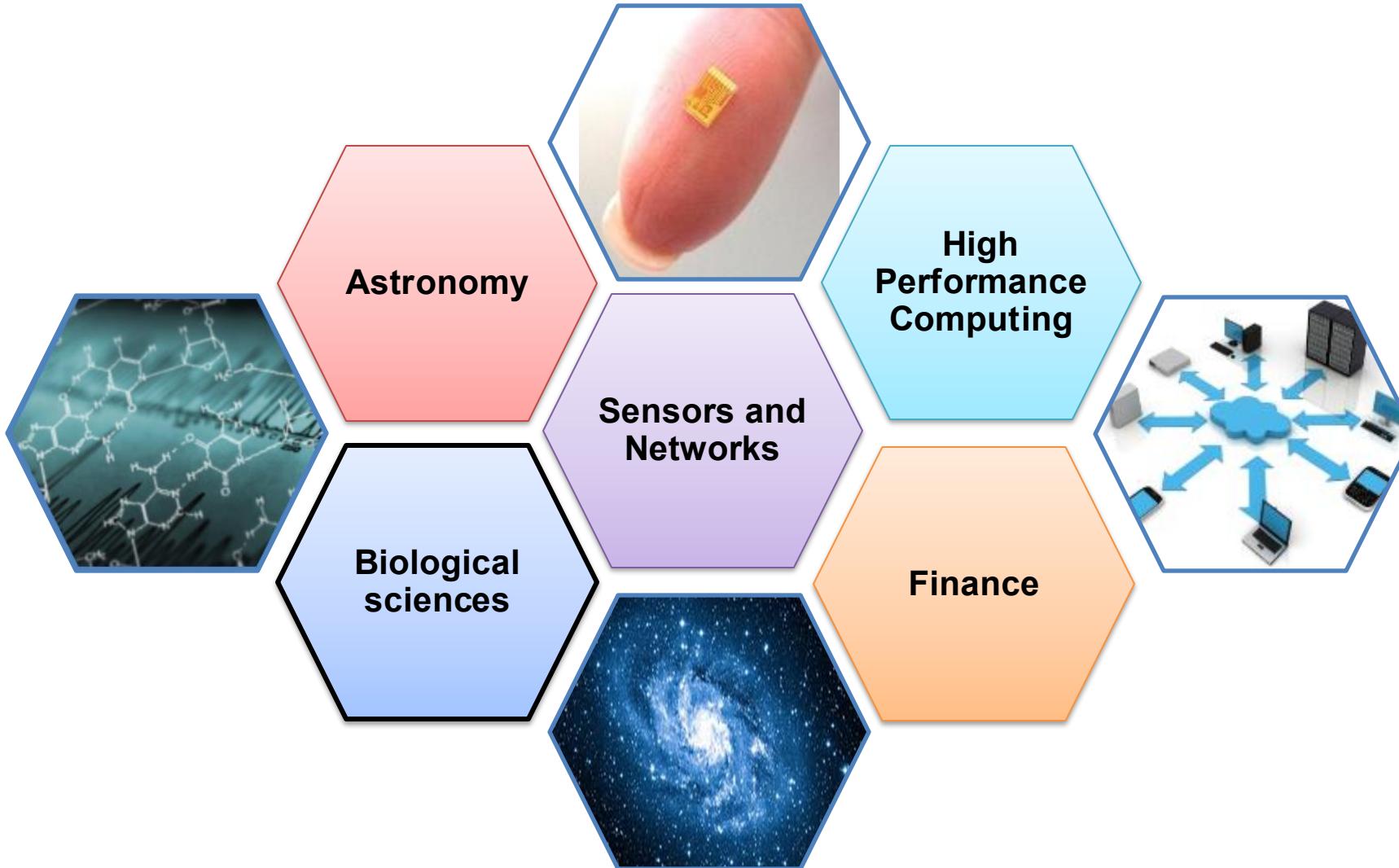
- ❖ A computer is just a machine (**hardware**) for executing programs (**software**)

- ❖ 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



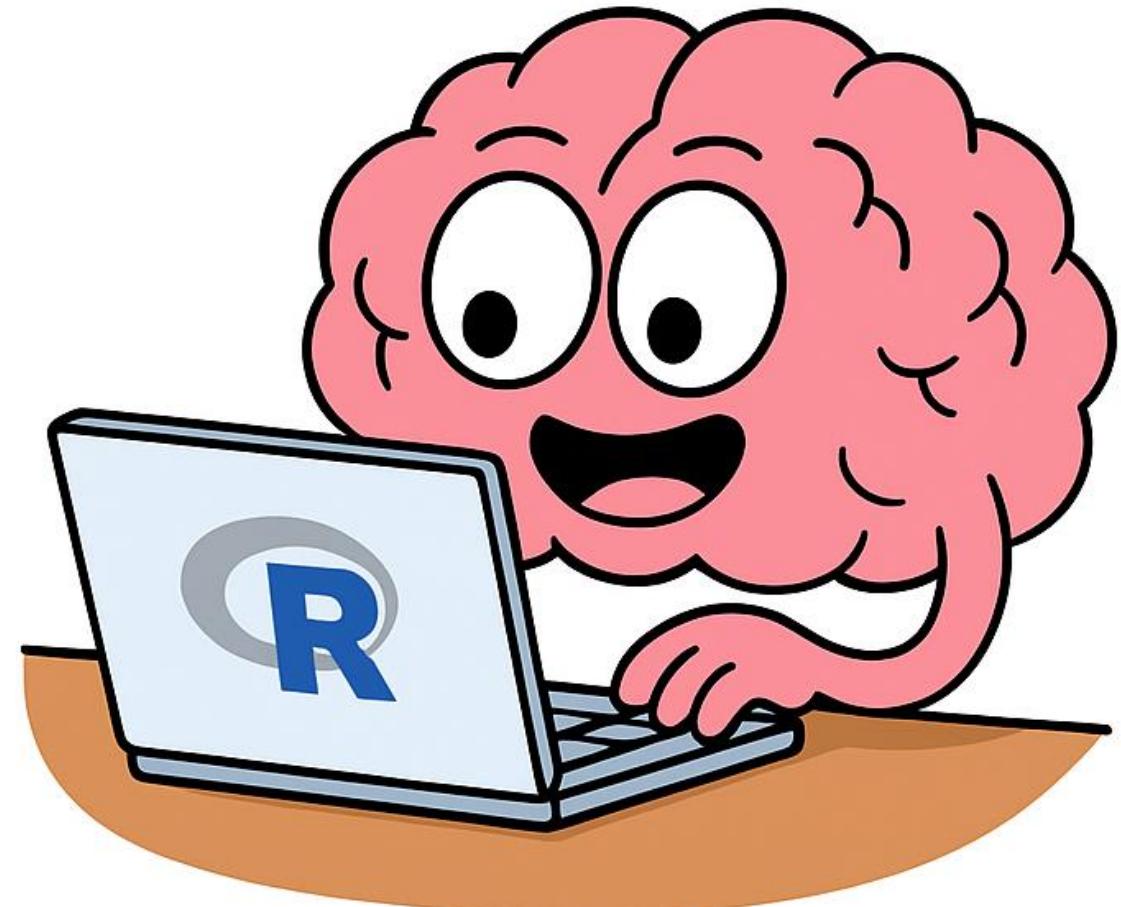
Compute p-values and
other statistics



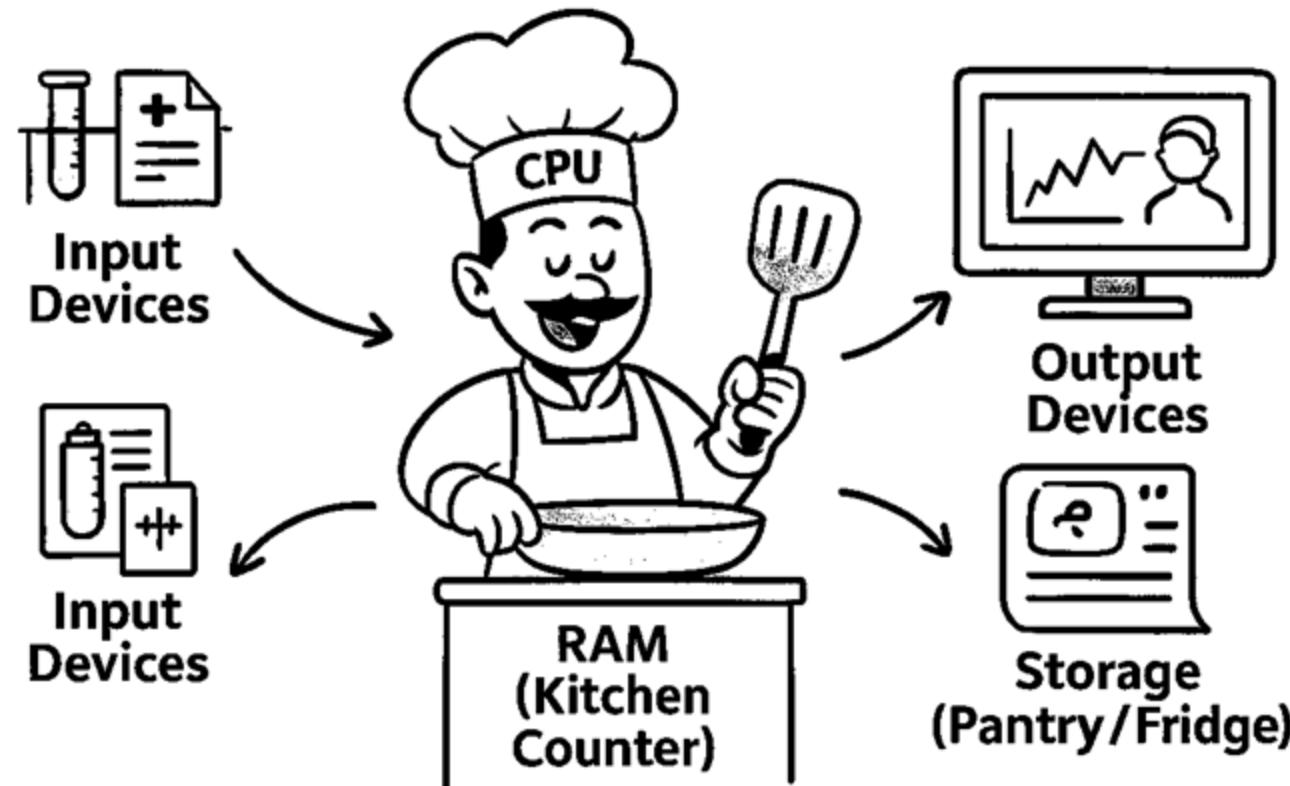
High-throughput
data analysis



Reproducible research



Computers explained simply



Hardware is the kitchen. Software is the recipe.
Programming = writing the recipe

Programming languages

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

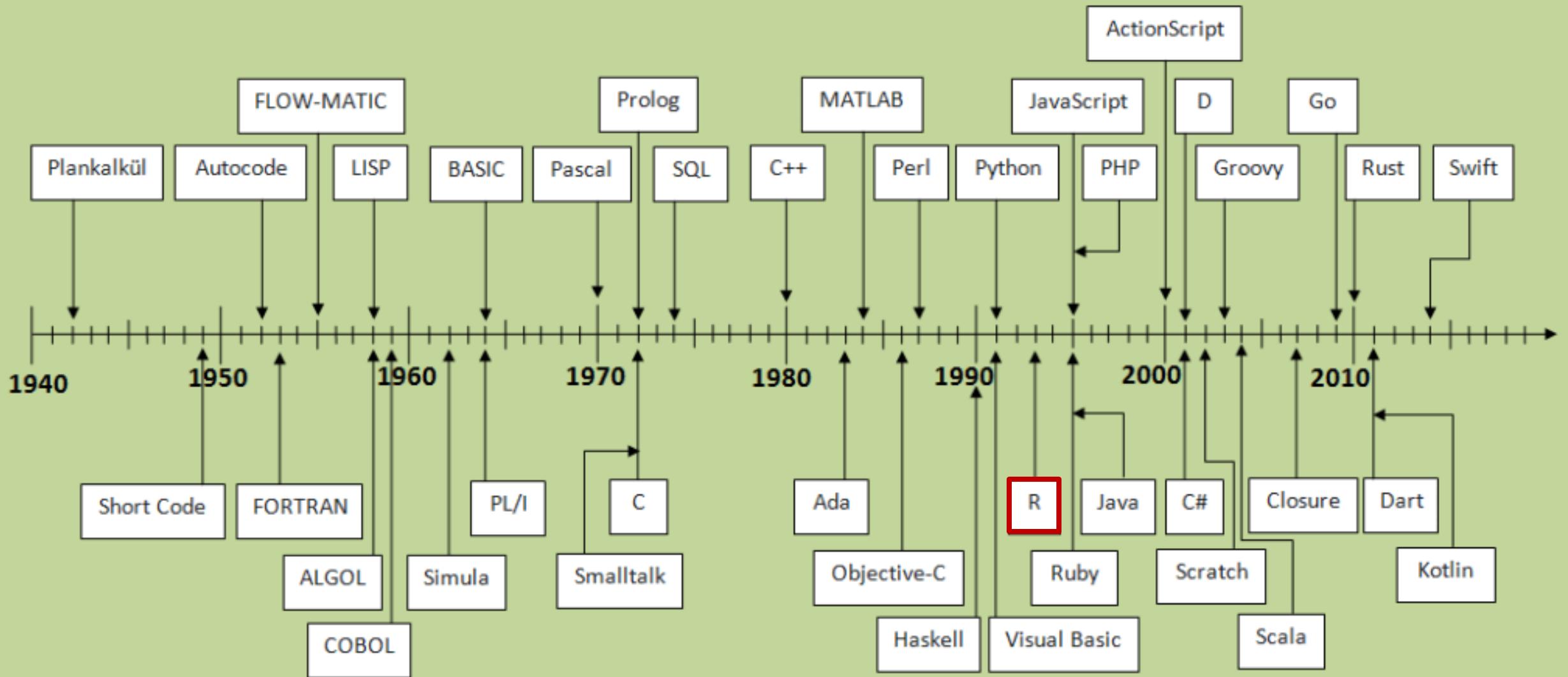
For example:

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

$$\left. \begin{array}{l} 1. \text{ Load the number from memory location 2001 into the CPU} \\ 2. \text{ Load the number from memory location 2002 into the CPU} \\ 3. \text{ Add the two numbers in the CPU} \\ 4. \text{ Store the result into location 2003} \end{array} \right\} A + B = C$$

- 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 **Ross Ihaka** and **Robert 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



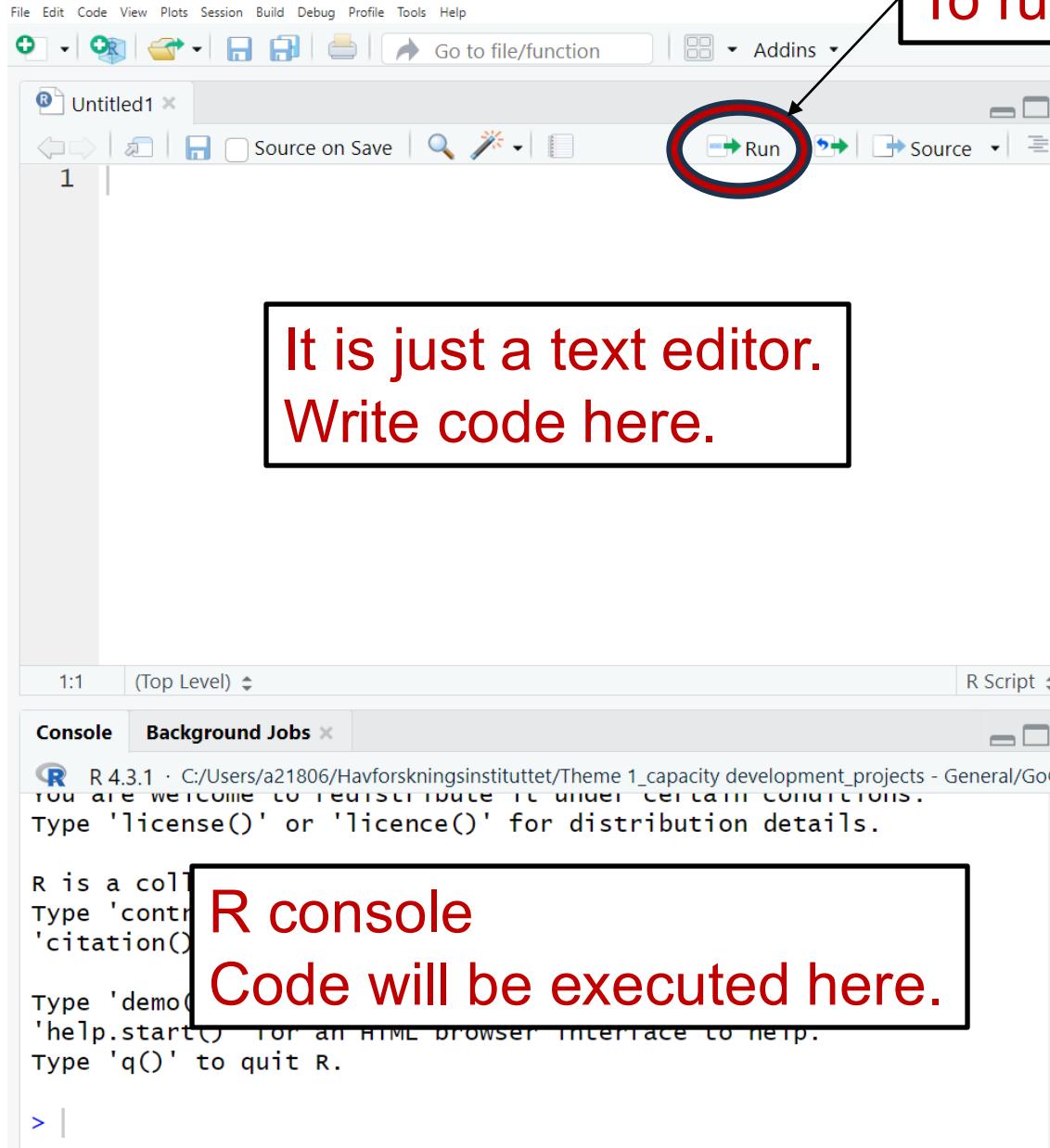
Use R from the terminal

```
R  
(base) c02z46udlvc:NEUROSYSMED930_R_workshop kleftogi$ R  
  
R version 4.4.1 (2024-06-14) -- "Race for Your Life"  
Copyright (C) 2024 The R Foundation for Statistical Computing  
Platform: x86_64-apple-darwin20  
  
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.  
  
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
  
> |  
  
Click to add notes
```

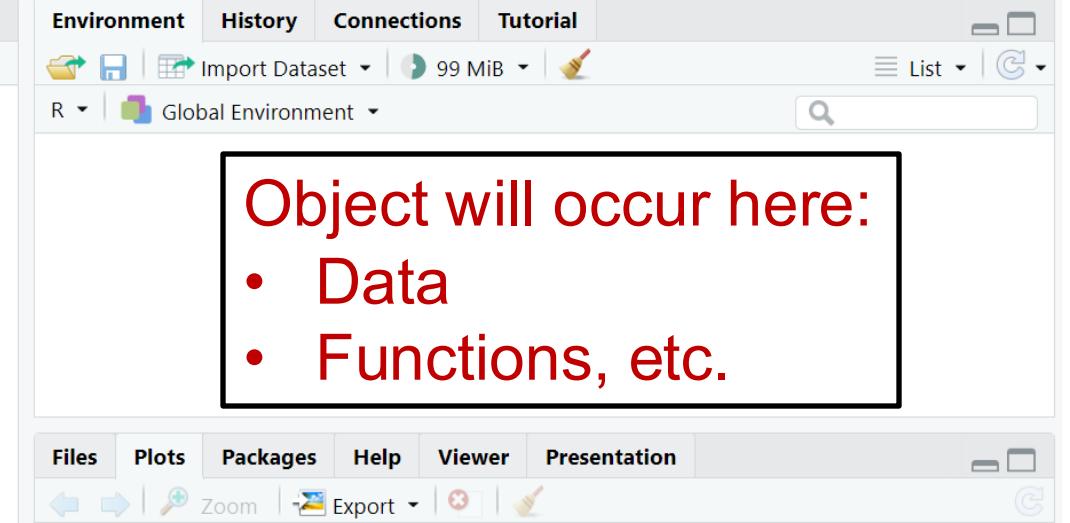
```
R Console  
STOP R Help Search  
  
R version 4.4.1 (2024-06-14) -- "Race for Your Life"  
Copyright (C) 2024 The R Foundation for Statistical Computing  
Platform: x86_64-apple-darwin20  
  
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.  
  
Natural language support but running in an English locale  
  
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
  
During startup - Warning messages:  
1: Setting LC_CTYPE failed, using "C"  
2: Setting LC_COLLATE failed, using "C"  
3: Setting LC_TIME failed, using "C"  
4: Setting LC_MESSAGES failed, using "C"  
5: Setting LC_MONETARY failed, using "C"  
[R.app GUI 1.80 (8416) x86_64-apple-darwin20]  
  
WARNING: You're using a non-UTF8 locale, therefore only ASCII characters will work.  
Please read R for Mac OS X FAQ (see Help) section 9 and adjust your system preferences  
accordingly.  
[Workspace restored from /Users/kleftogi/.RData]  
[History restored from /Users/kleftogi/.Rapp.history]  
  
2024-11-18 10:20:30.006 R[48455:10555424] +[IMKClient subclass]: chose IMKClient_Modern  
2024-11-18 10:20:30.823 R[48455:10555424] +[IMKInputSession subclass]: chose  
IMKInputSession_Modern  
> |
```

Use the R console

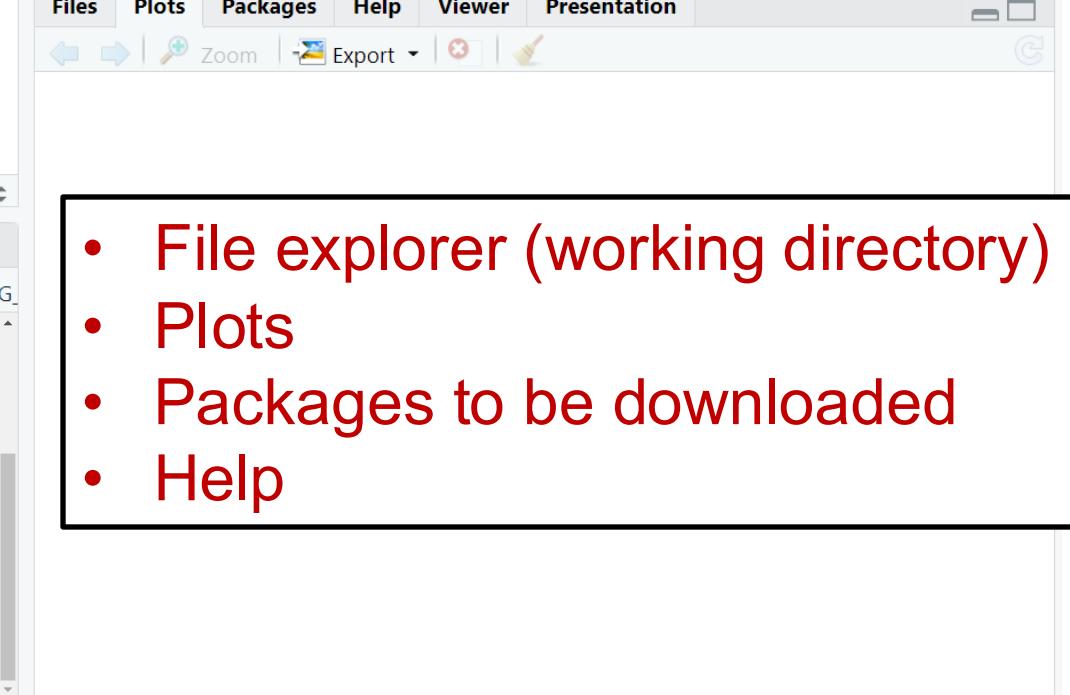
The R studio environment



To run code, click on “Run”



Object will occur here:
• Data
• Functions, etc.



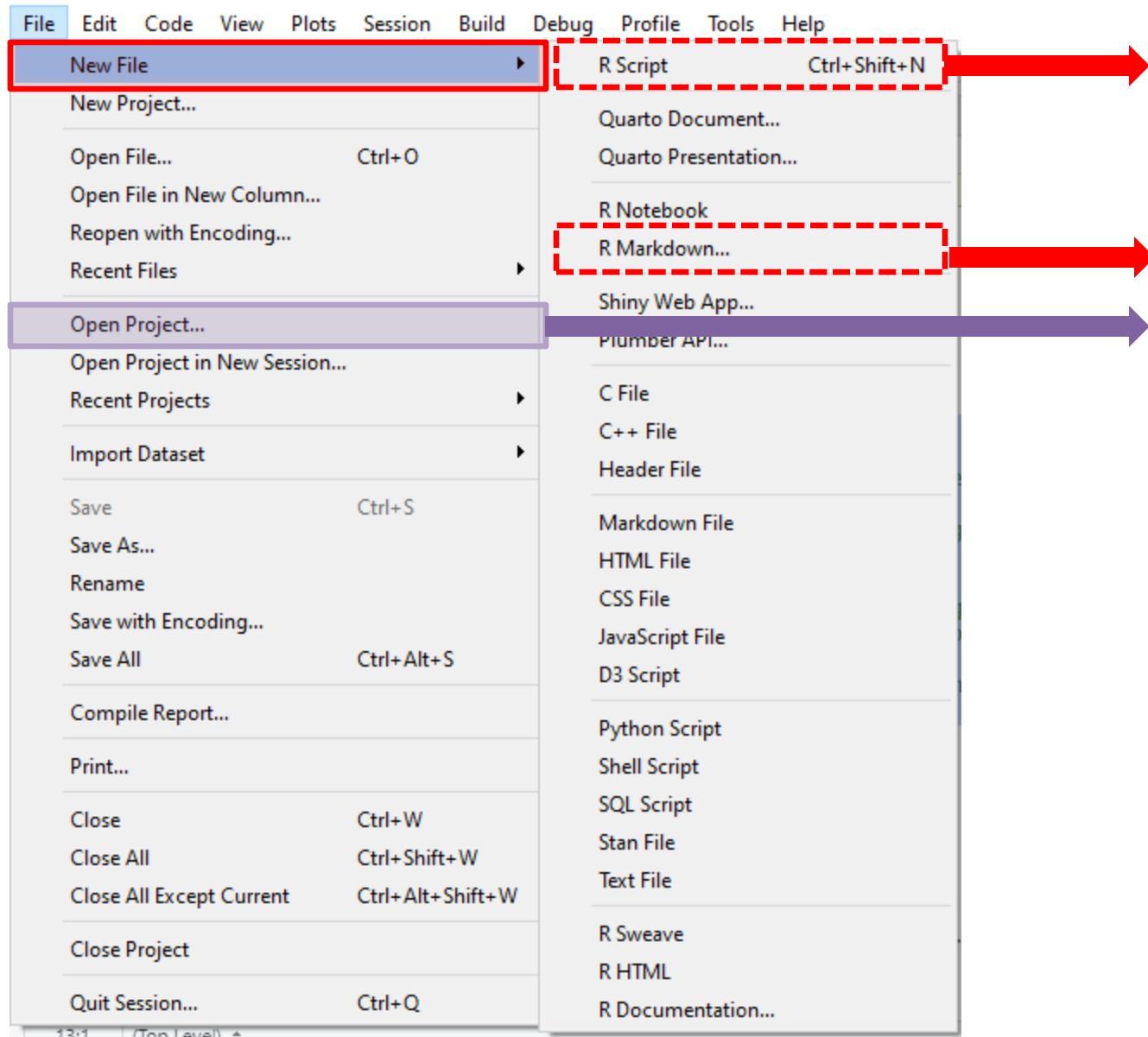
- File explorer (working directory)
- Plots
- Packages to be downloaded
- Help

R console
Code will be executed here.

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 colleagues

R Scripts vs. R Markdowns

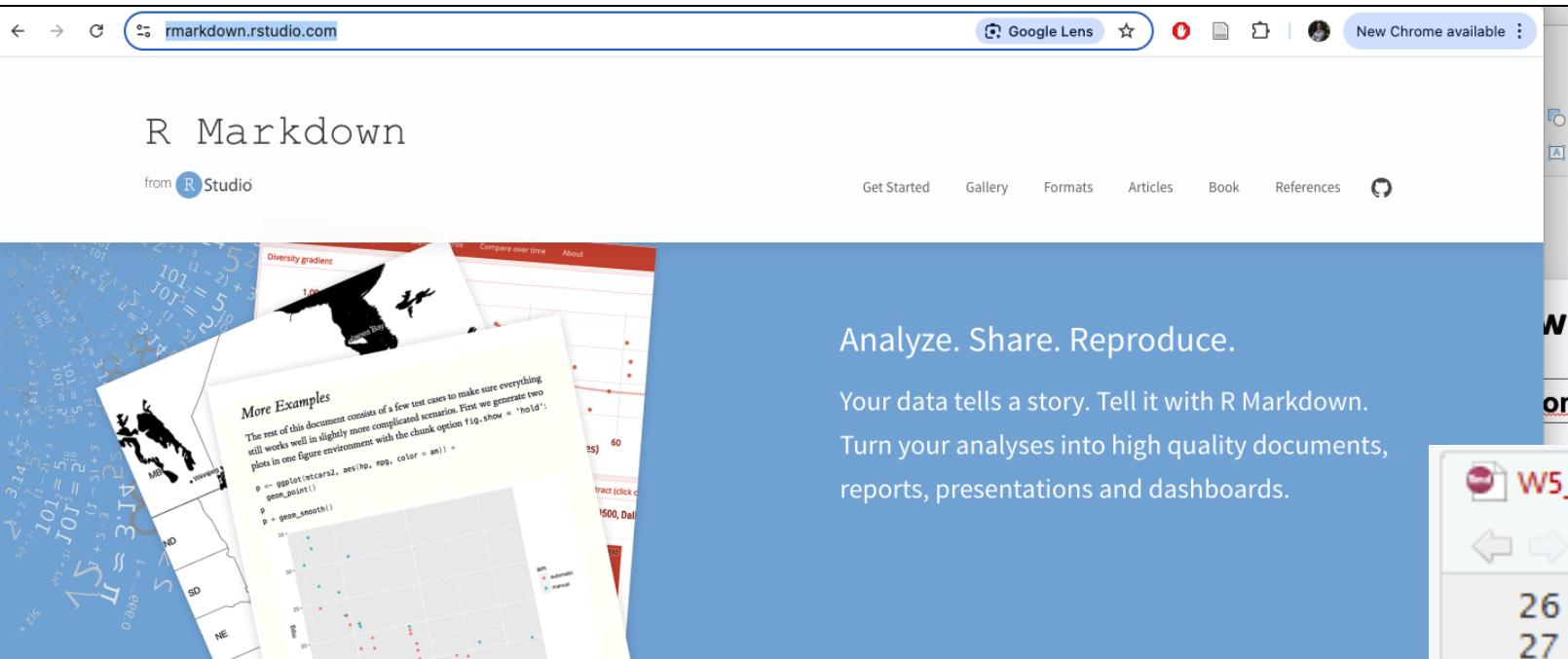
The screenshot shows the RStudio interface with the following details:

- Title Bar:** The title bar displays "First script.R x".
- Toolbar:** Includes icons for back, forward, source control, search, and file operations.
- Code Editor:** The main pane contains the following R script:

```
1 # this is my first R script
2 # do some things
3 x = 34
4 y = 16
5 z = x + y    # addition
6 w = y/x      # division
7 # display the results
8 x
9 y
10 z
11 w
12 # change x
13 x = "some text"
14 # display the results
15 x
16 y
17 z
18 w
19
```
- Status Bar:** Shows "1:1 (Top Level) ▾" and "R Script ▾".
- Console:** The console pane at the bottom shows the command prompt and the output of running the script:

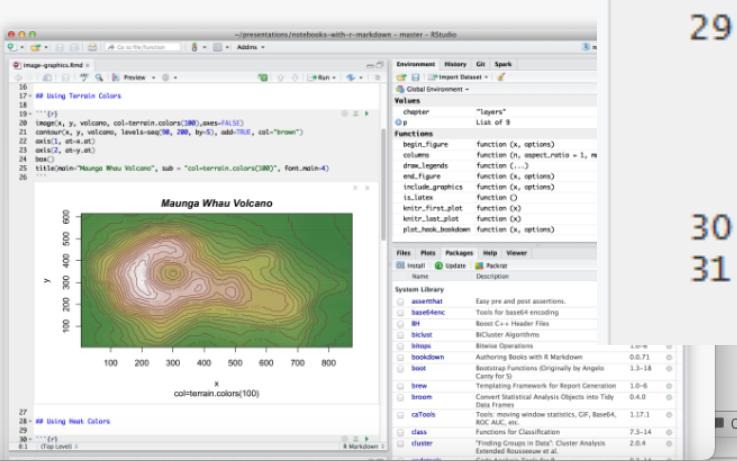
```
> z = x + y
> w = y/x
```
- Environment View:** A sidebar on the right titled "Envir" showing variable values: W=34, X=16, Y=16, Z=50, and W="some text".
- Files View:** Shows the current file "First script.R" is selected.
- System View:** Shows the current user "lisa" is selected.

R Scripts vs. R Markdowns

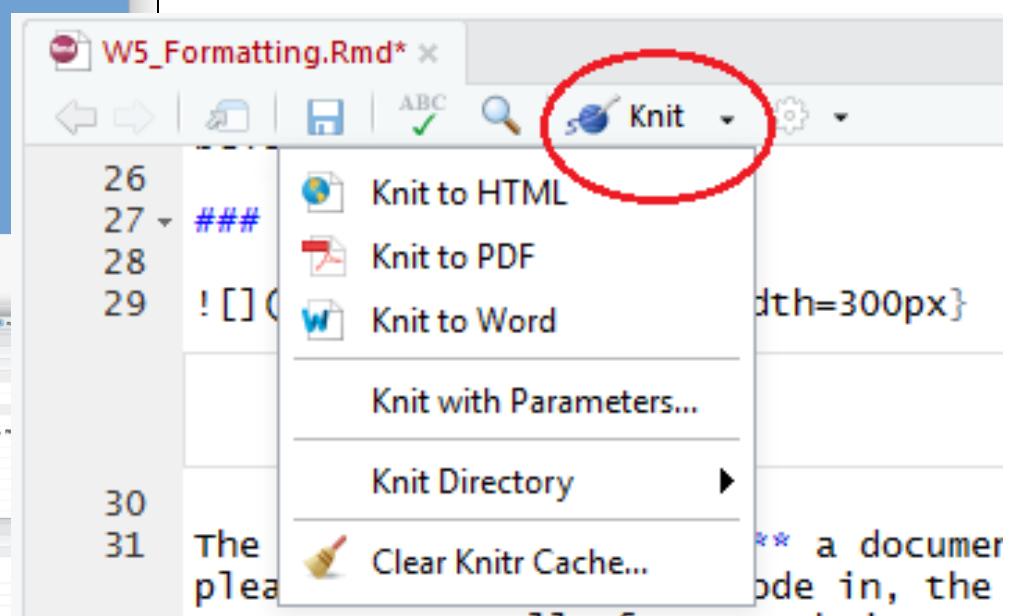


The screenshot shows the official R Markdown website at rmarkdown.rstudio.com. It features a header with navigation links like "Get Started", "Gallery", "Formats", "Articles", "Book", "References", and "About". Below the header, there's a large image of a collage of various R Markdown documents and plots, including a map of the United States with data overlays, a scatter plot, and a histogram. The main text area contains the slogan "Analyze. Share. Reproduce." followed by "Your data tells a story. Tell it with R Markdown. Turn your analyses into high quality documents, reports, presentations and dashboards."

R Markdown documents are fully reproducible. Use a productive [notebook interface](#) to weave together narrative text and code to produce elegantly formatted output. Use [multiple languages](#) including R, Python, and SQL.



This screenshot shows an RStudio interface with an R Markdown notebook titled "image-graphics.Rmd". The code section contains R code for generating a contour plot of Maunga Whau Volcano. The plot is displayed below the code, showing a green and yellow color gradient over a grid of points. The RStudio environment pane shows the global environment and a file browser.



This screenshot shows the RStudio interface with a red circle highlighting the "Knit" button in the toolbar. A dropdown menu is open, listing options: "Knit to HTML", "Knit to PDF", "Knit to Word", "Knit with Parameters...", "Knit Directory", and "Clear Knitr Cache...". The background shows a portion of the RStudio code editor with some R code and a contour plot.

** a document code in, the

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 addition:

In file(fil

cannot op

```
>
```

1:1 (Top Level) ▾

R Script ▾

Console

Terminal ×

Jobs ×

~ /

> library("dplyr")

Attaching pac

The followin

between

The followin

filter,

The followin

interse



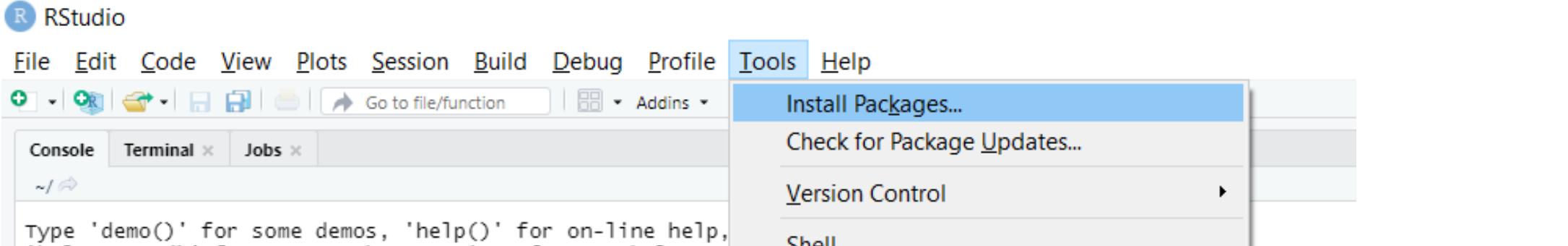
Warning message:

package 'dplyr' was built under R version 3.6.2

oup"

Packages

- Different ways to install packages exist



The screenshot shows the RStudio interface. The title bar says "R RStudio". The menu bar has "File", "Edit", "Code", "View", "Plots", "Session", "Build", "Debug", "Profile", "Tools" (which is highlighted in blue), and "Help". Below the menu bar is a toolbar with various icons. The main workspace shows a console tab with the message "Type 'demo()' for some demos, 'help()' for on-line help,". The "Tools" menu is open, showing "Install Packages...", "Check for Package Updates...", "Version Control", and "Shell".

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

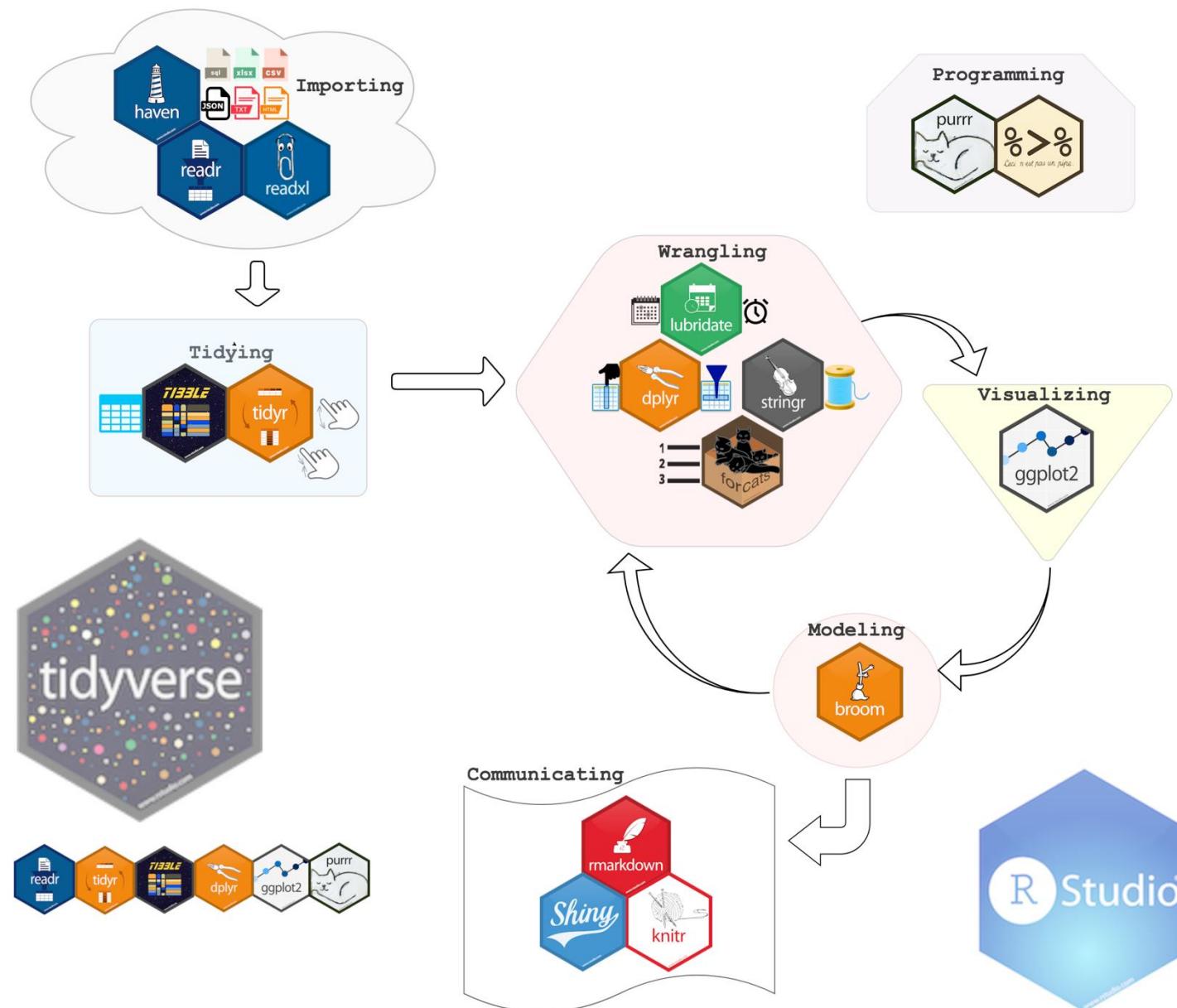
- Never forget to load the installed packages....



```
library("ggplot2")
```

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





(c) Image from the www
All rights reserved.

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>

