

# Handling experimental data with R

## Day 1

SALOS 2025

Alena Witzlack-Makarevich

## Overview

---

- Today
  - Getting ready: installations, packages and DataCamp leader board
  - R and RStudio basics: a quick reminder
  - Reports with R `markdown`
  - Importing datasets into R
- In the next days
  - Data manipulation with `dplyr`
  - Data visualization with `ggplot2`
  - Some inferential statistics and modeling
  - And whatever else we can squeeze into the 5 days

R

## About R

---

- R is an open source programming language and software environment for statistical computing and graphics
- The R language is widely used among statisticians and data miners for developing statistical software and data analysis.
- R is freely available under the GNU General Public License at <http://cran.r-project.org/>  
CRAN = Comprehensive R Archive Network

## Basic installation and packages

---

- The basic installation enables you to
  - manage your data
  - perform some calculations
  - use widespread statistical methods
  - represent data using basic plot types
- to use R for specific objectives (e.g. nicer plots), you need additional **packages** (as of Friday, 22447 packages)
- e.g. the package collection **tidyverse** with **dplyr** and **ggplot2**
- developed by R-users for specialized functions
- it is possible to program your own package

## Getting ready

---

- Expectation: you have some R foundations and you are about to finish the three DataCamp courses
- Reality: we'll do a 25-min crash course

RANK	NAME	EMAIL	COURSES COMPLETED <span>↑↓</span>	CHAPTERS COMPLETED <span>↑↓</span>	LAST XP <span>↑↓</span>	XP EARNED <span>≡↓</span>
		anna.daugavet@flf.vu.lt	1	7	2025-07-09	16770
	 Aigul Zakirova	aigul.n.zakirova@gmail.com	3	11	2025-07-17	16200
		pkk@cc.au.dk	3	15	2025-07-17	15513
4		gaelle.metois@univ-poitiers.fr	3	14	2025-07-18	15301
5		micom@uji.es	3	14	2025-07-19	14455
6	 Inesa Gilyte	inesa.gilyte@flf.stud.vu.lt	2	8	2025-07-19	14076
7		richard.udes@flf.stud.vu.lt	2	13	2025-07-19	13966

## Getting ready

---

- Tools: you have **R** and **RStudio** installed, you also have a **spreadsheet** application and a good **text editor** (e.g. ...) on your computer just in case
- Open RStudio and run this command in the console now, let R download the package:  
`install.packages("tidyverse")`
- Check on your neighbor whether they have it and ask them why I think we need them all

## R studio

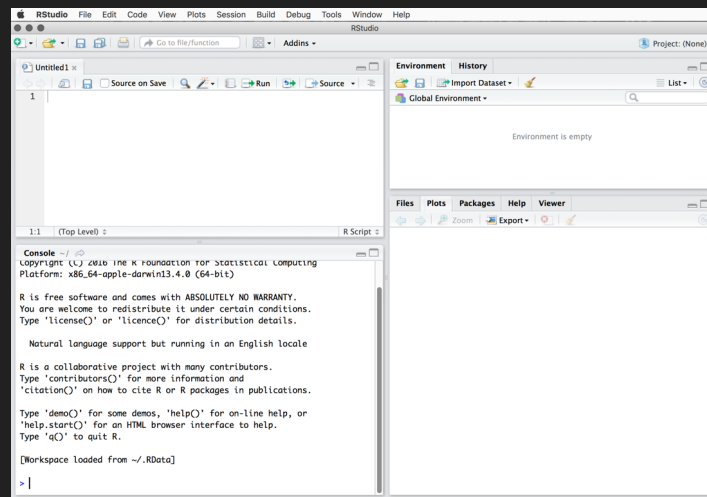
Advanced users:

help your neighbor

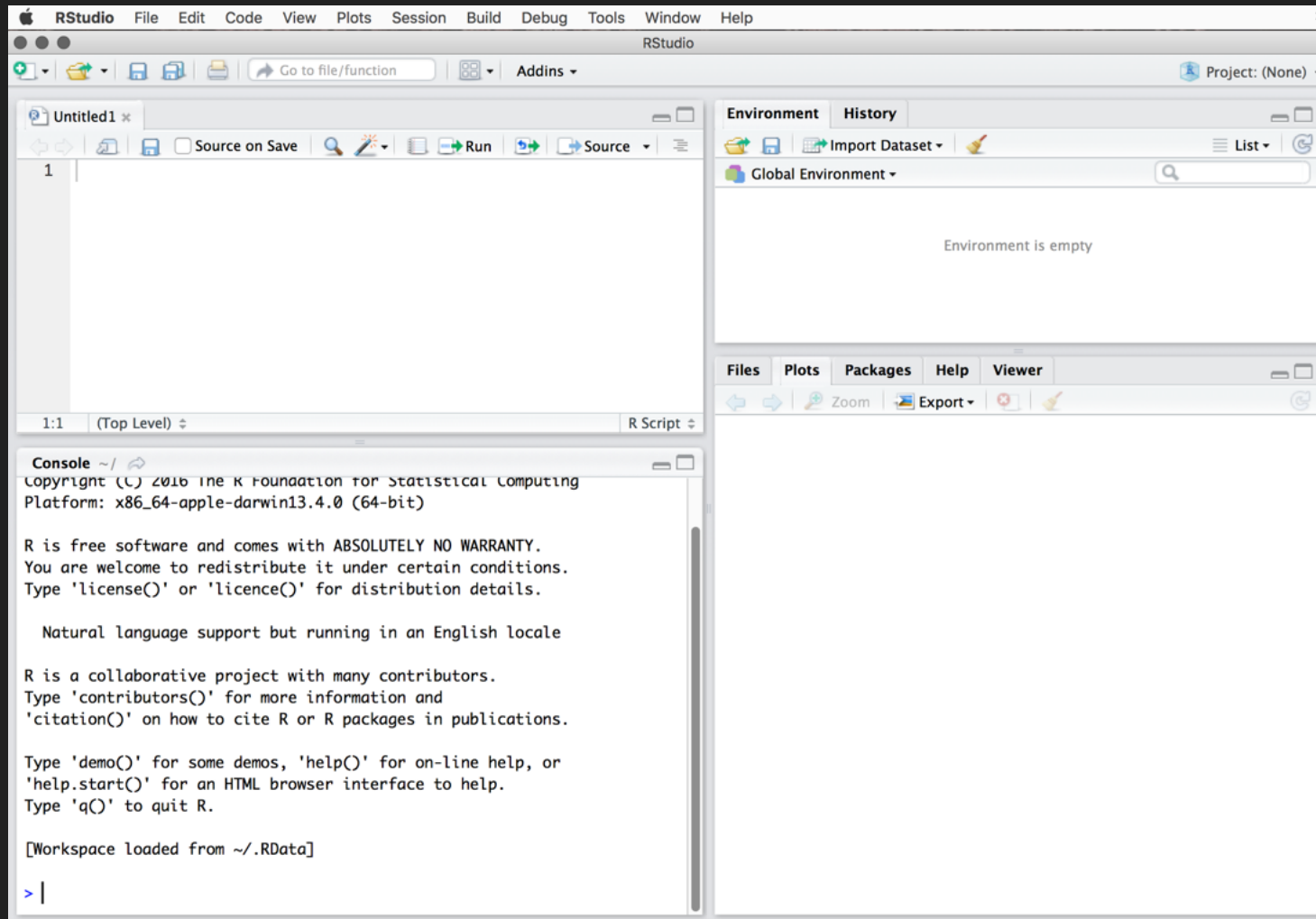
or check out RStudio Tutorial in DataCamp (12 min. read)

## RStudio: Four windows

- Open RStudio, it will automatically launch R software
- You should see **four** windows. What are they?
- Click here and there, **resize** them if needed
- Figure out how to **increase the font** so that you we can see all the missed commas and brackets



# RStudio: Four windows



# RStudio: Four windows

The screenshot displays the RStudio application window. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Tools, Window, and Help. Below the menu is a toolbar with icons for file operations and a search bar. The main workspace is divided into four panes:

- Source:** Contains an editor for 'Untitled1.R'. A red arrow points to the window control buttons (minimize, maximize, close) in the top right corner of this pane. A yellow box with blue text is overlaid on the editor, reading "If you are missing a window".
- Environment:** Shows the current environment, which is empty. It includes an 'Import Dataset' button and a search bar.
- Console:** Displays the R startup message, including copyright information and instructions on how to use the software.
- Files/Plots/Packages/Help/Viewer:** A multi-tabbed pane for viewing files, plots, packages, help, and viewer content.

```
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)

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.

[Workspace loaded from ~/.RData]
> |
```

# RStudio: Four windows

The screenshot shows the RStudio interface with four windows highlighted in yellow boxes:

- Scripts / Notebooks:** The main editor window showing a notebook with R code and Markdown text. The code includes `plot(cars)` and `write.csv` commands.
- Environment / History:** The Environment pane showing a list of objects in the Global Environment, such as `KKG_nogloss_tokens` (832362 obs. of 3 variables) and `KU_eaf` (88839 obs. of 15 variables).
- Console:** The console window showing the execution of R code, including `genre` and `write.csv` commands.
- Plots / Help / ...:** The Help pane showing the documentation for the `write.table` function, including its description and usage.

**R 101**

## R basics revision: 15 min.

- For the next 15 min. work in the **console**
- Show your neighbor that you know how to do
  - a. basic **maths** in R
  - b. introduce a **variable**
  - c. assign to it a **vector** of numbers, look up what `c()` is for
  - d. apply some **functions** to this vector (do “**vectorized calculation**”)
  - e. **access** individual values of the vector
  - f. **change/overwrite** individual values of the vector
  - g. check out how to use the **Environment** and **History** window
  - h. check whether your neighbor knows the basics you know
  - i. produce three different **error** types  
and interpret the **error messages**

## R basics revision: 15 min.

---

- Best coding practice: **spaces** around the operators!

<b>Addition</b>	+
<b>Subtraction</b>	-
<b>Multiplication</b>	*
<b>Division</b>	\
<b>Exponentiation</b>	^

```
> 1 + 1
[1] 2
> 23-1
[1] 22
> 134 - 23 * 4
[1] 42
> 1 +      2
[1] 3
> 2 * (6 - 1)
[1] 10
```

Source



Console

Render x

Backgro



R 4.5.1 · ~/



```
[1] 4
> 2 + 2
[1] 4
> 2 * 2
[1] 4
> 2 - 2
[1] 0
> (2 + 2) * 6
[1] 24
> length <- c(4, 6, 3, 7, 4, 5,
> length
[1] 4 6 3 7 4 5 4
> length <- c(5, 6, 3, 7, 4, 5,
> length
[1] 5 6 3 7 4 5 4
> length * 10
[1] 50 60 30 70 40 50 40
> sum(length)
```

Environment

History

Connections

T



To Console



To Source



```
2 + 2
2 * 2
2 - 2
(2 + 2) * 6
length <- c(4, 6, 3, 7, 4, 5, 4)
length
length <- c(5, 6, 3, 7, 4, 5, 4)
length
length * 10
sum(length)
mean(length)
length[3]
length[3] <- 4
length
length
median(length)
str(length)
```

Source



Console

Render x

Backgro



R 4.5.1 · ~/



```
[1] 4
> 2 + 2
[1] 4
> 2 * 2
[1] 4
> 2 - 2
[1] 0
> (2 + 2) * 6
[1] 24
> length <- c(4, 6, 3, 7, 4, 5,
> length
[1] 4 6 3 7 4 5 4
> length <- c(5, 6, 3, 7, 4, 5,
> length
[1] 5 6 3 7 4 5 4
> length * 10
[1] 50 60 30 70 40 50 40
> sum(length)
```

Environment

History

Connections

T



To Console





To Source



```
2 + 2
2 * 2
2 - 2
(2 + 2) * 6
length <- c(4, 6, 3, 7, 4, 5, 4)
length
length <- c(5, 6, 3, 7, 4, 5, 4)
length
length * 10
sum(length)
mean(length)
length[3]
length[3] <- 4
length
length
median(length)
str(length)
```

## R basics revision: 15 min.

---

- The history window allows you to choose previous commands or edit these
- The arrows  and  allow you to navigate through the history
- The command appears at the bottom of your console after the command prompt (>)
- If you find the command you wish to re-use, just press the enter key to execute this command again
- Or edit the command first by using the arrows  and 
- Try to avoid retyping commands

## R basics revision: 15 min.

---

- For the next 15 min. work in the **console**
- Show your neighbor that you know how to do

provoke three different **error** types  
and try to interpret the **error messages**

## Errors

---

```
> 5numbers <- c(4, 5, 3, 6, 4)
Error: unexpected symbol in "5numbers"
> numbers <- c(4, 5, 3, 6, 4}
Error: unexpected '}' in "numbers <- c(4, 5, 3, 6, 4}"
> numbers <- c(4, 5, 3, 6. 4)
Error: unexpected numeric constant in "numbers <- c(4, 5, 3, 6. 4"
> numbers <- c(4, 5. 3, 6, 4)
Error: unexpected numeric constant in "numbers <- c(4, 5. 3"
```

## Errors

```
> numbers <- c(4, 5, 3, 6, 4)
> numbers
[1] 4 5 3 6 4
> numbers * + 3
[1] 12 15 9 18 12
> numbers * (3 + 3
+
+
+ )
[1] 24 30 18 36 24
> 5numbers
Error: unexpected symbol in "5numbers"
> number
Error: object 'number' not found
```

## Functions

---

- There are many predefined functions (or commands) in R
- Which ones did you use just now?
- Which ones do you know or use a lot?
  
- Just type the name of the function followed by **round brackets**
- The arguments of a function are given in the brackets
- They are separated by **commas**

e.g. 

```
> sum(3, 3, 3)
[1] 9
```

## Functions

---

- Functions can include 0, 1, 2,... obligatory arguments and optional arguments
- `data()` doesn't need any arguments.
- What do the following functions do?
  - `colors()` doesn't need arguments
  - `ls()` doesn't need arguments
  - `sqrt()` needs one numeric argument
  - `seq()` needs two numeric arguments
  - `str()` needs a vector or a data frame as an argument
  - `length()` needs a vector or a data frame as an argument

# R Markdown

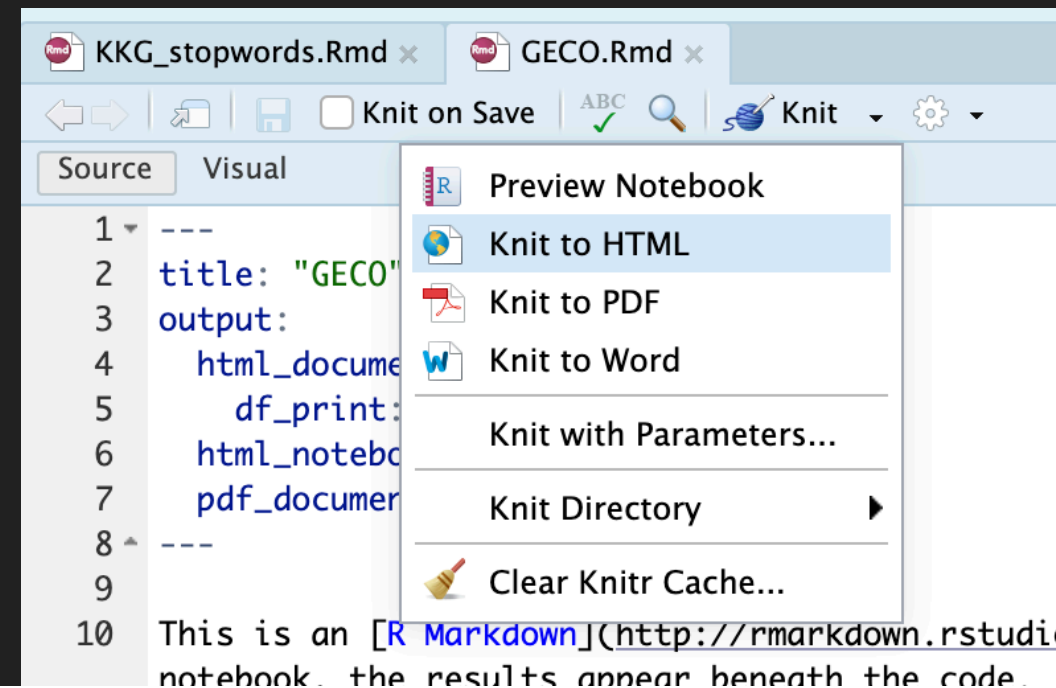
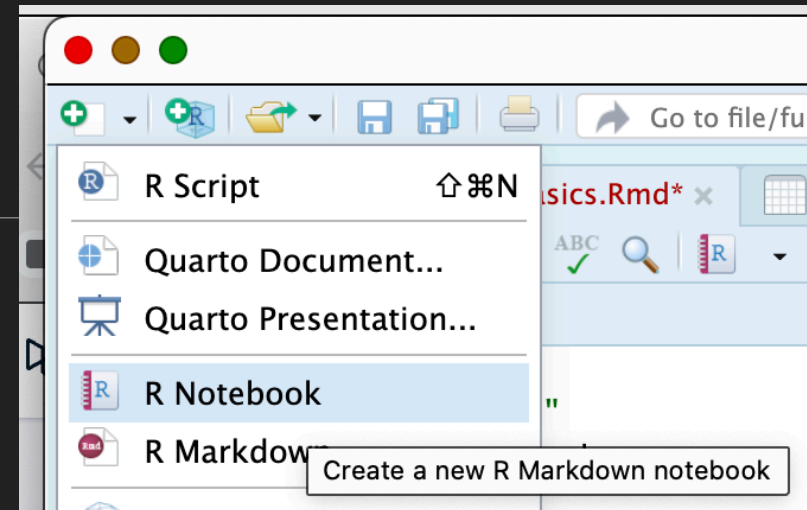
## RStudio: Working with the notebook

---

- Console is ok for small things and one-liners
- In this course we'll work with R **Markdown** and its **notebook** interface
- R Markdown documents are fully **reproducible**
- “Use a productive notebook interface to weave together narrative text and code to produce elegantly formatted output.”
- In your case, use it to add comments and notes to what you do
- And produce a not-so-elegant output you can reuse in a year if needed

## R Markdown: Notebook

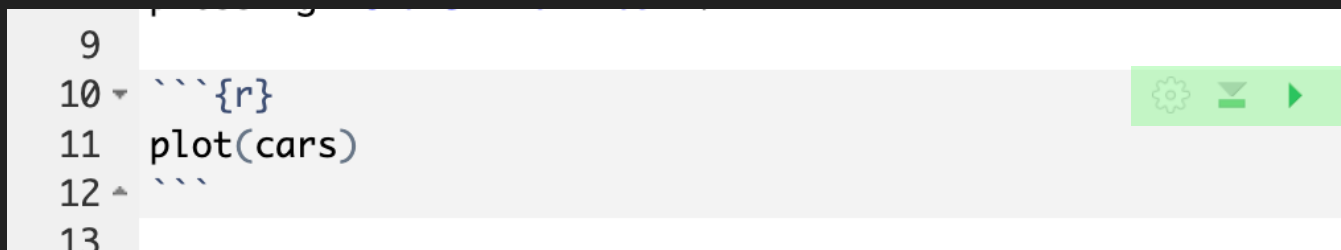
- Create a new R notebook
- Save it (e.g. as **RBasics**)
- It comes with a **template**
- **Knit** the template to HTML to see what happens
- Change a few things, decipher the syntax
- Knit again
- Does it work?



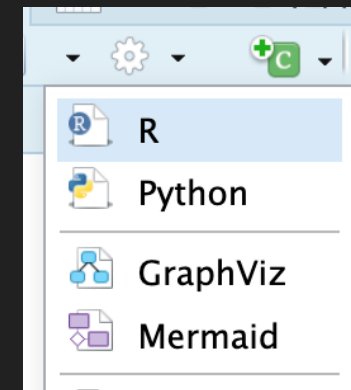
## R Markdown: Notebook

- Which line contains a R code chunk?
- How is it formatted?
- You can execute one chunk (**green arrow** or a shortcut)

```
9  
10 ▾ ```{r}  
11 plot(cars)  
12 ▲ ```  
13
```



- Create five more chunks and write some of the basic maths and functions we tried a few minutes ago in the console
- Copy-paste or use **the green +C button**



## R Markdown workflow

---

- Open a new .Rmd file
- Embed code in chunks
- Run code by line, by chunk, or all at once
- Write text and add tables, figures, images, and citations.  
Format with Markdown syntax
- Customize themes or add parameters to execute
- Save and render the whole document
- **Knit** periodically to preview your work as you write
- Share your work!

# Importing data into R

# **L2 Lexical decision dataset (part of GECO)**

## GECO

---

- the **G**hent **E**ye-Tracking **C**orpus
- a monolingual and bilingual corpus of the **eyetracking** data of participants reading a complete novel
- English monolinguals and **Dutch–English bilinguals** read an entire novel, which was presented in paragraphs on the screen
- the bilinguals read half of the novel in their first language,
- and the other half in their second language
- to answer questions about **naturalistic reading processes** in a meaningful context

## What is our dataset? A tiny part of GECCO

---

- All participants completed a “battery of language proficiency tests”, including a vocabulary test, a spelling test, a lexical decision task, and a self-report language questionnaire
- A dataset of a lexical decision task
- Performed in English (L2) by **81** Dutch L1 participants
- The dataset contains raw reaction times (**RT**) and accuracy (**ACC**) scores for 800 stimuli
- As well as some participant information
- In the larger eye-tracking study, the dataset is used to compare the L1 and L2 word-level age-of-acquisition effect between lexical decision and eye movement measures

# What is our dataset? A tiny part of GECCO

- Download the dataset
- We'll need the `.csv` file, but you can also check out `.xlsx`
- Open the dataset (either in the spreadsheet or better in a text editor)
- Inspect it with your neighbor
- How large is the dataset?
- What kind of variables does it contain?

PPNR	AGE	SEX	HANDED	STIMULUS	WORDTYPE	RT	ACC	AVG_ACC	LEXTALE_EN	LEXTALE_DU
9	41	18 male	right	bugtime	NonWord	675	1	93	68.75	83.75
0	41	18 male	right	poison	Word	480	1	93	68.75	83.75
1	41	18 male	right	marvel	Word	540	0	93	68.75	83.75
2	41	18 male	right	pame	NonWord	557	1	93	68.75	83.75
3	41	18 male	right	hole	Word	511	1	93	68.75	83.75
4	41	18 male	right	witness	Word	806	1	93	68.75	83.75

# What is our dataset? A tiny part of GECCO

- A dataset of a lexical decision task performed in English as L2 by Dutch L1 participants
- In this study, the dataset was used to compare the L1 and L2 word-level age-of-acquisition effect between lexical decision and eye movement measures
- The dataset has also been used in a more general comparison of lexical decision RTs and timed eye movement measures

PPNR	AGE	SEX	HANDEDNESS	STIMULUS	WORDTYPE	RT	ACC	AVG_ACC	LEXTALE_EN	LEXTALE_DU	
9	41	18	male	right	bugtime	NonWord	675	1	93	68.75	83.75
0	41	18	male	right	poison	Word	480	1	93	68.75	83.75
1	41	18	male	right	marvel	Word	540	0	93	68.75	83.75
2	41	18	male	right	pame	NonWord	557	1	93	68.75	83.75
3	41	18	male	right	hole	Word	511	1	93	68.75	83.75
4	41	18	male	right	witness	Word	806	1	93	68.75	83.75

# What is our dataset? A tiny part of GECCO

PPNR	AGE	SEX	HANDEDN	STIMULUS	WORDTYPE	RT	ACC	AVG_ACC	LEXTALE_EN	LEXTALE_DU	
9	41	18	male	right	bugtime	NonWord	675	1	93	68.75	83.75
0	41	18	male	right	poison	Word	480	1	93	68.75	83.75
1	41	18	male	right	marvel	Word	540	0	93	68.75	83.75
2	41	18	male	right	pame	NonWord	557	1	93	68.75	83.75
3	41	18	male	right	hole	Word	511	1	93	68.75	83.75
4	41	18	male	right	witness	Word	806	1	93	68.75	83.75

Participants' details

**Speeded** lexical decision task:  
reaction times (**RT**)  
and accuracy (**ACC**) scores  
for 800 stimuli

**Un-speeded**  
lexical decision tasks

## LEXTALE variables (<https://www.lextale.com/>)

- a quick test of vocabulary knowledge for medium to highly proficient speakers of English as L2
- a simple **un-speeded** visual lexical decision task
- designed to meet the needs of cognitive researchers

### *What is LexTALE?*



LexTALE: Lexical Test for  
Advanced Learners of English...

### *Take the test*



Participants can take the  
LexTALE test online in English,  
Dutch and German...

*abergy*

no

yes

Score 95 %

# What is our dataset? A tiny part of GECCO

- Which information in this dataset is repeated?
- Which one is not?
- What implications does it have for the further analysis?

PPNR	AGE	SEX	HANDEDN	STIMULUS	WORDTYPE	RT	ACC	AVG_ACC	LEXTALE_EN	LEXTALE_DU
41	18	male	right	bugtime	NonWord	675	1	93	68.75	83.75
41	18	male	right	poison	Word	480	1	93	68.75	83.75
41	18	male	right	marvel	Word	540	0	93	68.75	83.75
41	18	male	right	pame	NonWord	557	1	93	68.75	83.75
41	18	male	right	hole	Word	511	1	93	68.75	83.75
41	18	male	right	witness	Word	806	1	93	68.75	83.75

## L2LexDec\_GECO.csv

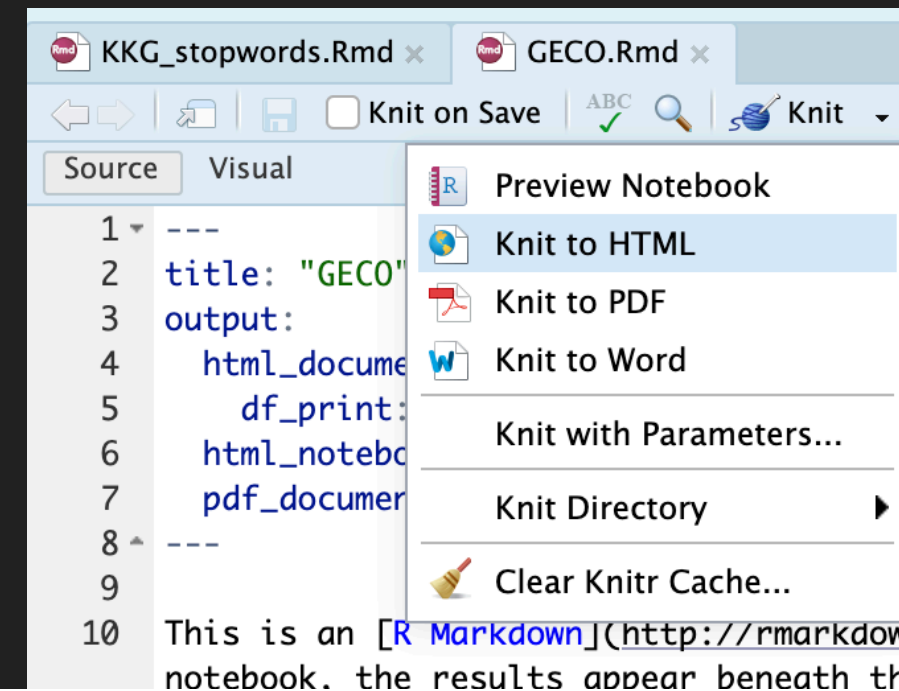
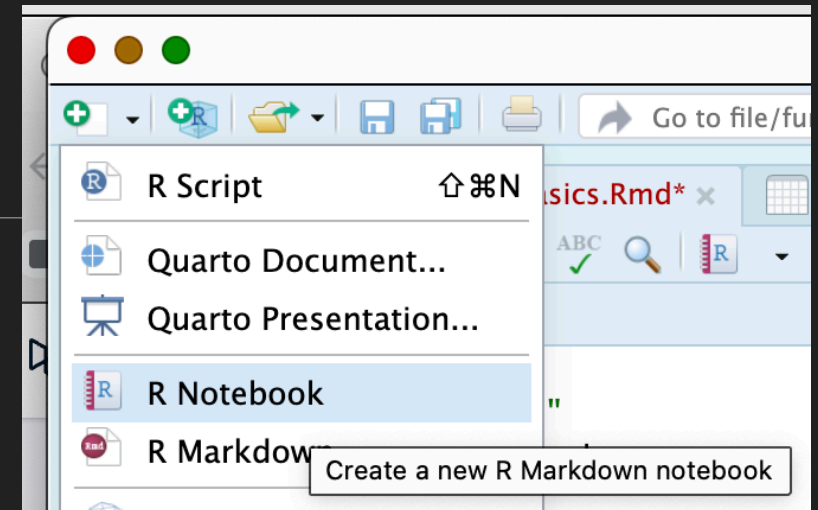
- What does the extension `.csv` stand for? What is it good for?
- Do not open it in Excel!
- And if you do, do not overwrite the original file!
- Open it in a text editor
- What is the column **separator**?

1	PPNR, AGE, SEX, HANDEDNESS, STIMULUS, WORDTYPE, RT, ACC, AVG_ACC, LEXTALE_EN, LEXTALE_DU
2	1, 20, female, right, ways, Word, 940, 1, 92, 80, 85
3	1, 20, female, right, post, Word, 531, 1, 92, 80, 85
4	1, 20, female, right, booding, NonWord, 1264, 1, 92, 80, 85
5	1, 20, female, right, text, Word, 517, 1, 92, 80, 85
6	1, 20, female, right, business, Word, 769, 1, 92, 80, 85
7	1, 20, female, right, lale, NonWord, 1130, 1, 92, 80, 85

**Do you have the dataset  
L2LexDec\_GECO.csv ready?  
Does your neighbor?**

## A new notebook

- Create a new R notebook
- Save it (e.g. as **GECO**)
- Write down a few words about the dataset
- **Knit** to HTML
- Does it work?



## Almost import the .csv file

---

- But we need one more thing

# Tidyverse

- tidyverse is an opinionated collection of R packages (incl. `dplyr`, `ggplot2`, `readr`)
- designed for data science
- all packages share an underlying design philosophy, grammar, and data structures
- and we will be using this approach in r
- in contrast to the old-school “base R”



## ggplot2

ggplot2 is a system for declaratively creating graphics, based on the grammar of graphics. You provide the data, tell ggplot2 how to map variables to aesthetic elements, and it takes care of the details. [Learn more ...](#)



## dplyr

dplyr provides a grammar of data manipulation, providing a consistent set of verbs that make it easy to learn and use. [Learn more ...](#)



## tidyr

tidyr provides a set of functions that help you get to tidy data. In brief, every variable goes in a column, and every column has one value. [Learn more ...](#)



## readr

readr provides a fast and friendly way to read rectangular data files. It can flexibly parse many types of data found in the wild, while staying consistent across changes. [Learn more ...](#)



## purrr

purrr enhances R's functional programming (FP) toolkit by providing a set of tools for working with functions and vectors. Once you master purrr, you can replace many for loops with code that is easier to write and maintain. [Learn more ...](#)



## tibble

tibble is a modern re-imagining of the data frame, keeping what works and throwing out what it has not. Tibbles are data.frames that are more forgiving when you do something wrong, typically leading to more helpful error messages. [Learn more ...](#)



## stringr

stringr provides a cohesive set of functions designed to make working with strings a breeze. It is built on top of stringi, which uses the ICU C library to provide common string manipulations. [Learn more ...](#)

# Tidyverse

- First, you need to install the tidyverse package (only once, which you have):  
`install.packages("tidyverse")`
- You do it one time and usually in the console
- Once the package is installed, load it (activate it):  
`library("tidyverse")`
- Good practice: all packages are loaded at the very beginning of an R script:

```
10 ▾ # Packages
11
12   First, we load the packages we need:
13
14 ▾ ```{r}
15   library(tidyverse)
16 ▲ ```
17
```



**Finally!**

**Importing data into R with `read_csv()`**

## `read_csv` (from tidyverse)

---

- `read_csv()` for reading the most common types of flat file data, comma separated values with `.` decimal point
- what is `read_tsv()` for?

## read\_csv (from tidyverse)

---

- The function has one obligatory argument:  
`read_csv(filepath)`
- If your script is where you file is:  
`GEC0 <- read_csv("L2LexDec_GEC0.csv")`
- Otherwise, get the correctly formatted path of the file with:  
in the console, type `file.choose()` (nothing in the brackets!)
- A pop-up window will allow you to choose the correct file:

```
> file.choose()
```

```
[1] "/Users/alena2/Library/CloudStorage/Dropbox/Teaching_Students/Classes  
6/SALOS2025/Case_studies/GEC02017Lexical/L2LexDec_GEC0.csv"
```

## read\_csv (from tidyverse)

- `GECO <- read_csv("L2LexDec_GECO.csv")`
- What assignment operator is used here?
- What variable is create?
- Why do you need it?
- What does the message after reading the file say?
- What could you do next?

```
23 ▾ `` `{r}`  
24 GECO <- read_csv("L2LexDec_GECO.  
25 ▲ `` `
```

Rows: 32400 Columns: 11

— Column specification —

Delimiter: ","

**chr** (6): SEX, HANDEDNESS, STIMU

**dbl** (5): PPNR, AGE, RT, ACC, AV

**i** Use ``spec()`` to retrieve the

**i** Specify the column types or s

## read\_csv (from tidyverse)

---

- `read_csv()` for reading the most common types of flat file data, comma separated values with `.` decimal point
- what is `read_tsv()` for?
- The function has one obligatory argument:  
`read_csv(file)`
- If your script is where you file is:  
`GECO <- read_csv("L2LexDec_GECO.csv")`
- If your script is where you file is:  
`GECO <- read_csv("L2LexDec_GECO.csv")`

# Inspecting the dataset in R

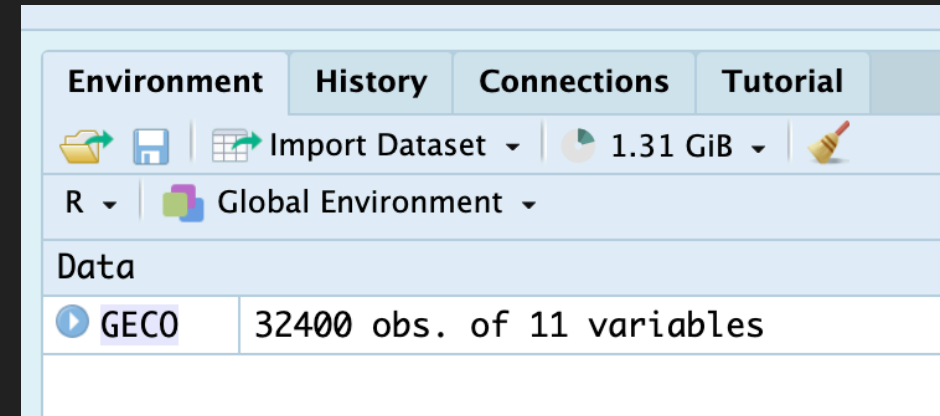
## Inspecting the dataset in

---

- Before starting any kind of analysis or further processing in R, always inspect your dataset:
  - Do you have all the variables?
  - Are they encoded the way they should?
  - Do you have all the rows?
  - Does anything look funny?
- What functions and other options you remember of?

## Inspecting the dataset in

- Check the Environment `head(GECO)`
- Click on the dataset there. What happens?  
Is it useful?
- Notice what happened in the console



The screenshot shows the RStudio Data Viewer for the 'GECO' dataset. The table has 11 columns: PPNR, AGE, SEX, HANDEDNESS, STIMULUS, WORDTYPE, RT, ACC, and AVG\_ACC. The first five rows are visible, showing data for subjects 1 through 5.

	PPNR	AGE	SEX	HANDEDNESS	STIMULUS	WORDTYPE	RT	ACC	AVG_ACC
1	1	20	female	right	ways	Word	940	1	92
2	1	20	female	right	post	Word	531	1	92
3	1	20	female	right	booding	NonWord	1264	1	92
4	1	20	female	right	text	Word	517	1	92
5	1	20	female	right	business	Word	760	1	92

## Inspecting the dataset in

---

- try `head(GECO)`

a convention: `head(df)`

`df` = any data frame,  
replace with your variable name

- Could you guess what `tail(df)` does?
- Try `str(df)`. Can you interpret its output?

## Inspecting the dataset in

- try `head(GECO)`

a convention: `head(df)`

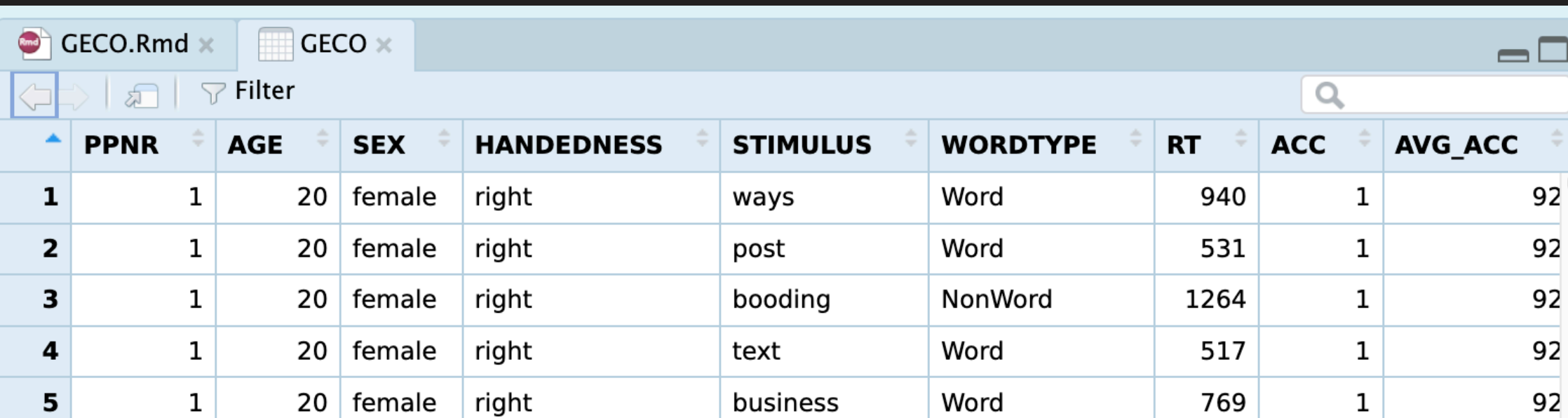
`df` = any data frame,  
replace with your variable name

- Could you guess what `tail(df)` does?
- Try `str(df)`. Can you interpret its output?
- Check on your neighbor. Have they written the commands in R notebook? How many code chunks do they have?

**First statistical exploration:  
the five-number summary**

## Inspecting the dataset

- Let's pick a good and easy variable to explore it further
- Which ones qualify and which ones maybe need more preparation?

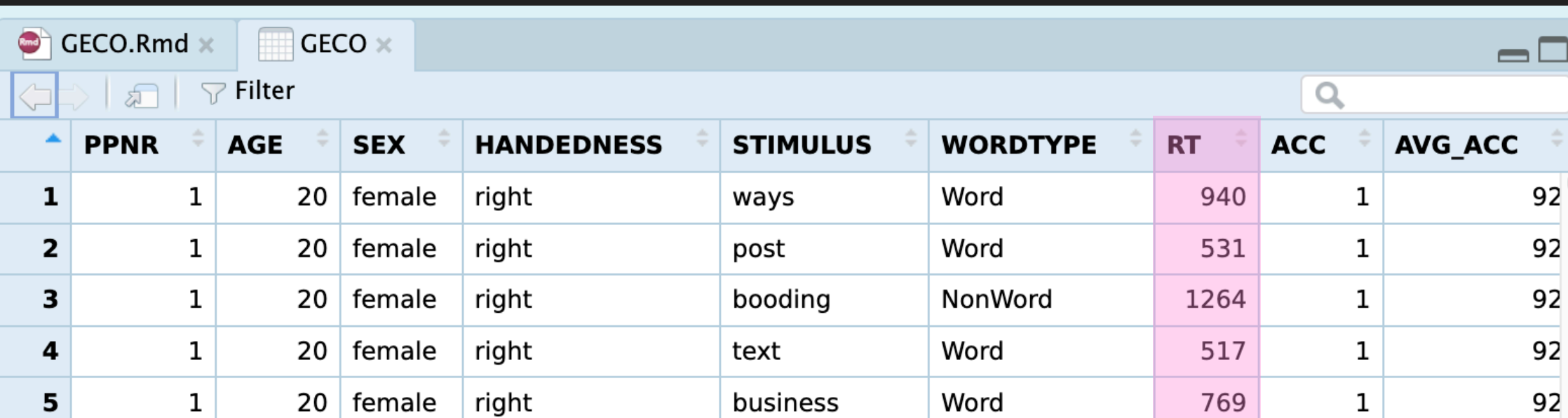


The screenshot shows a data viewer interface with a toolbar at the top containing navigation arrows, a filter icon, and a search box. The data is presented in a table with the following columns and rows:

	PPNR	AGE	SEX	HANDEDNESS	STIMULUS	WORDTYPE	RT	ACC	AVG_ACC
1	1	20	female	right	ways	Word	940	1	92
2	1	20	female	right	post	Word	531	1	92
3	1	20	female	right	booding	NonWord	1264	1	92
4	1	20	female	right	text	Word	517	1	92
5	1	20	female	right	business	Word	769	1	92

## Inspecting the dataset

- Let's pick a good and easy variable to explore it further
- Which ones qualify and which ones maybe need more preparation?



The screenshot shows a data viewer interface with a table of data. The table has 10 columns: PPNR, AGE, SEX, HANDEDNESS, STIMULUS, WORDTYPE, RT, ACC, and AVG\_ACC. The RT column is highlighted in pink. The data is as follows:

	PPNR	AGE	SEX	HANDEDNESS	STIMULUS	WORDTYPE	RT	ACC	AVG_ACC
1	1	20	female	right	ways	Word	940	1	92
2	1	20	female	right	post	Word	531	1	92
3	1	20	female	right	booding	NonWord	1264	1	92
4	1	20	female	right	text	Word	517	1	92
5	1	20	female	right	business	Word	769	1	92

# Inspecting the dataset

- try `summary(GECO)`
- Could you interpret the outcome?
- Focus on **RT**: Can you run `summary()` only on this variable? How?

```
31 # Get the first impression about th edistributio
32
33 ```{r message = FALSE, echo = FALSE}
34 summary(GECO)
35 ^```
```

PPNR	AGE	SEX	HANDEDNESS	STIMULUS
Min. : 1	Min. :17.00	Length:32400	Length:32400	Length:32400
1st Qu.:21	1st Qu.:18.00	Class :character	Class :character	Class :character
Median :41	Median :18.00	Mode :character	Mode :character	Mode :character
Mean :41	Mean :18.44			
3rd Qu.:61	3rd Qu.:18.00			
Max. :81	Max. :24.00			

RT	ACC	AVG_ACC	LEXTALE_EN	LEXTALE_DU
Min. : 0.0	Min. :0.0	Min. :77.00	Length:32400	Length:32400
1st Qu.: 538.0	1st Qu.:1.0	1st Qu.:87.00	Class :character	Class :character
Median : 641.0	Median :1.0	Median :91.00	Mode :character	Mode :character
Mean : 715.9	Mean :0.9	Mean :90.02		
3rd Qu.: 805.0	3rd Qu.:1.0	3rd Qu.:93.00		
Max. :2478.0	Max. :1.0	Max. :97.00		

## Inspecting the dataset

---

- try `summary(GECO)`
- Could you interpret the outcome?
- Focus on `RT`: Can you run `summary()` only on this variable? How?

# Tasks and suggestions for advanced students

## L2 Lexical decision (GECO)

---

- Things to try out
  - Create an R markdown notebook
  - Read the dataset
  - Provide some basic visualizations for the variables where it makes sense
  - Is **AVG\_ACC** (average accuracy) indeed the mean of **ACC** (accuracy)?
  - Who is the fastest participant (in terms of **RT**)?
  - Describe the participants' demographics
  - What variables can be correlated?