



## 00 - First Tour

# Data Science with R · Summer 2021

Uli Niemann · Knowledge Management & Discovery Lab

<https://brain.cs.uni-magdeburg.de/kmd/DataSciR/>

# What is R?

- Statistical programming language with focus on reproducible data analysis
- Free, open source and available for every major OS
- Since January 2017 more than 10,000 packages available
  - Comprehensive statistics and machine learning packages
  - Elaborate packages to create aesthetically appealing graphics and charts

## What is RStudio?

- Open source Integrated Development Environment (IDE) for R
- First release in 2011
- Two major versions: RStudio Desktop and RStudio Server
- Download & Installation of R and RStudio:  
<https://www.rstudio.com/products/rstudio/download/>



# RStudio IDE

The screenshot displays the RStudio IDE interface with several open panes:

- Script Editor:** Shows an R script named "r\_script.R" with the code:

```
1 a <- 7
2 a + 3
```
- Console:** Displays the R startup message and a session transcript:

```
R version 3.4.1 (2017-06-30) -- "Single Candle"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (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.

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.

> source('D:/Dropbox/R/RStudio_Demo/r_script.R', echo=TRUE)
> a <- 7
> a + 3
[1] 10
>
```
- Workspace Browser + History:** Shows the Global Environment pane with variable "a" set to 7, and the History tab.
- Directory + Plots + Packages + Documentation:** Shows the Files pane listing the contents of the RStudio\_Demo directory:

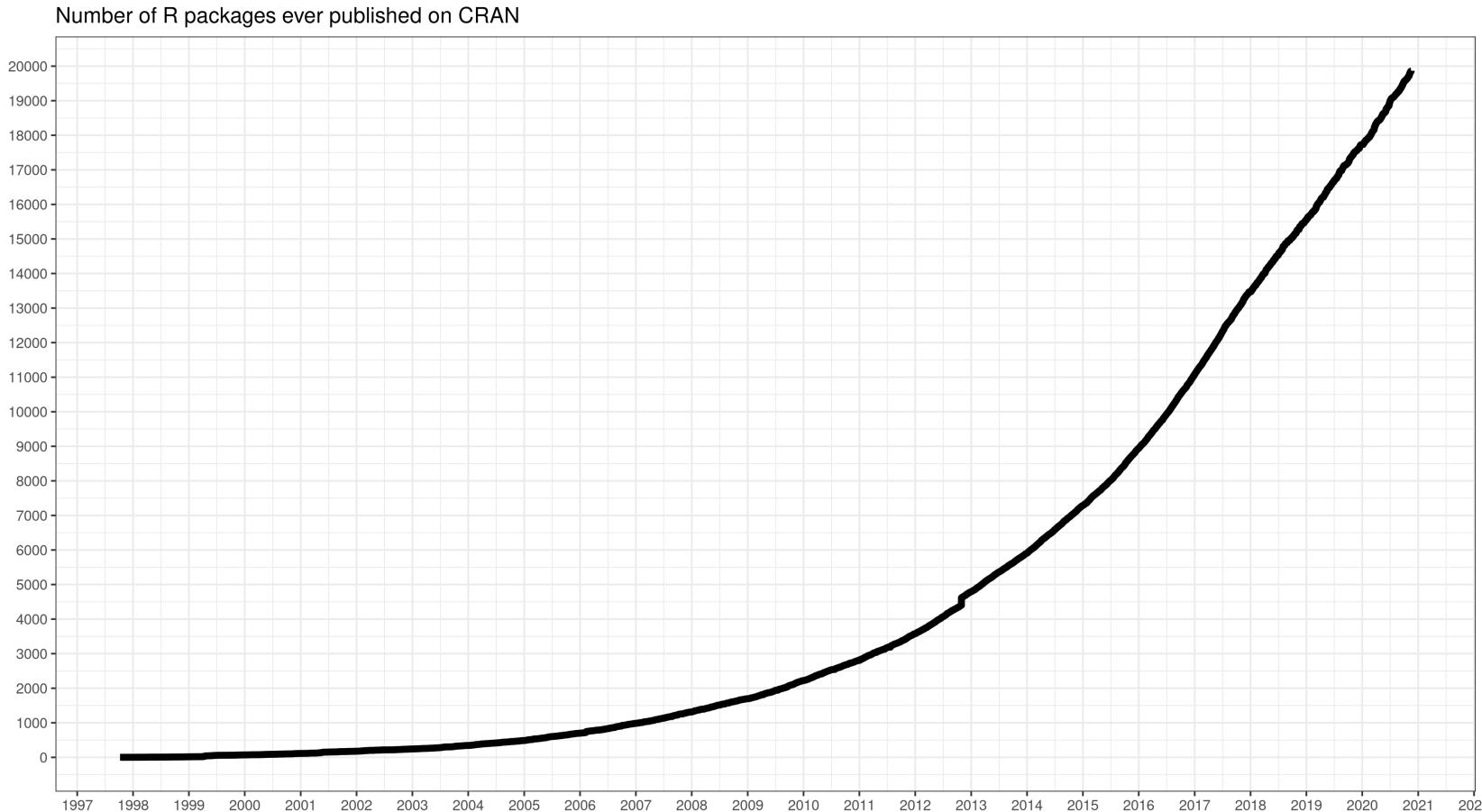
Name	Size	Modified
..		
.Rproj.user		
r_script.R	13 B	Jul 6, 2018, 6:5
RStudio_Demo.Rproj	218 B	Jul 6, 2018, 6:5

# R packages

- Base `R` contains functions that are needed by the majority of users.
- Additional functions can be used on demand by loading **packages**.

A **package** is a **collection of functions, datasets and documentation** that extends the capabilities of base R.

# Packages on CRAN<sup>1</sup>



<sup>1</sup> CRAN, the *The Comprehensive R Archive Network*, consists of multiple worldwide mirror servers, used to distribute R and R packages.  
Figure source: <https://gist.github.com/daroczig/3cf06d6db4be2bbe3368>

# Tour: R and RStudio

# Calling functions

Basic function call scheme:

```
some_function(arg_1 = val_1,  
              arg_2 = val_2,  
              ...)
```

The screenshot shows the R Help Viewer window. The title bar says "R: Arithmetic Mean". The main content area is titled "Arithmetic Mean". It includes sections for "Description", "Usage", "Arguments", and "Details". The "Usage" section shows the command `mean(x, ...)`. The "Arguments" section details three parameters: `x` (An R object), `trim` (the fraction of observations to be trimmed), and `na.rm` (a logical value indicating whether NA values should be stripped before the computation proceeds). The "Details" section notes that the function is generic and provides the source code for the S3 method.

Example: the `mean()` function:

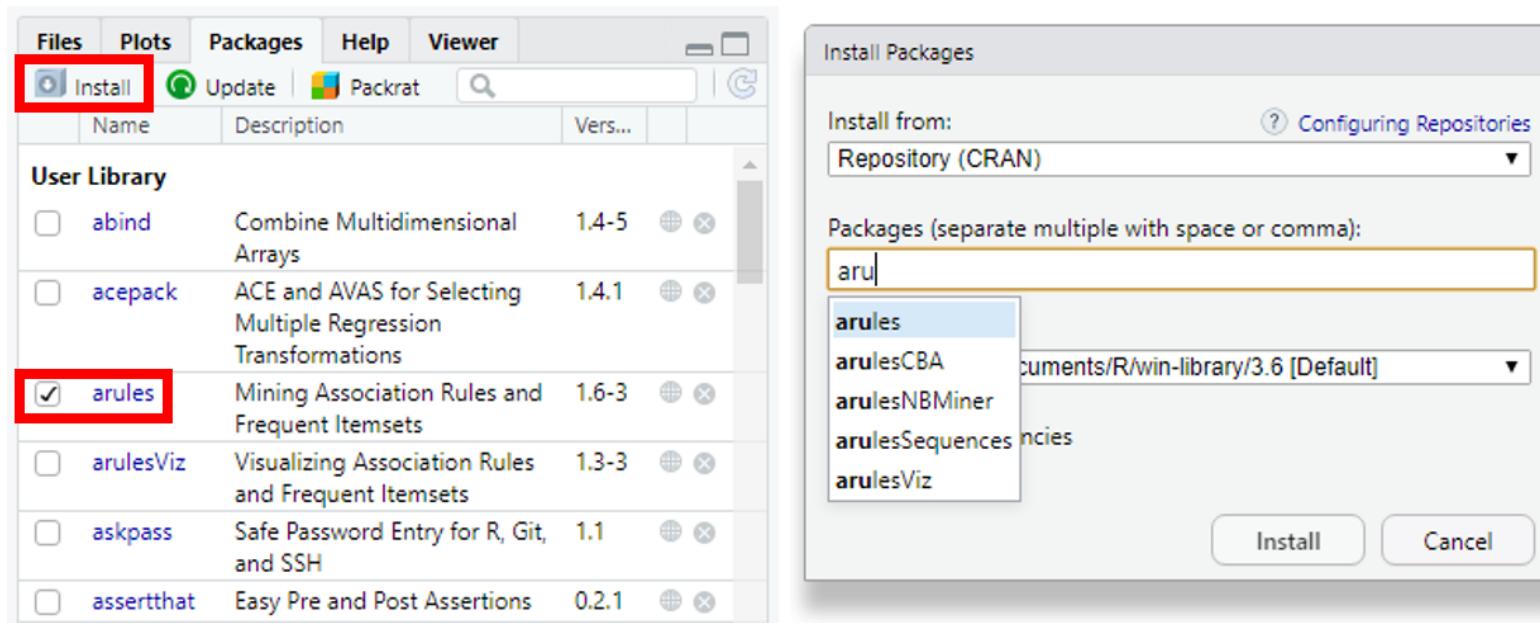
- `x` is the only mandatory argument
- arguments `trim` and `na.rm` have default values

```
x <- 1:10  
mean(x) # trim = 0 and na.rm = FALSE  
  
## [1] 5.5  
  
x <- c(1:10, NA)  
mean(x)  
  
## [1] NA  
  
mean(x, na.rm = TRUE) # NA's will be ignored  
  
## [1] 5.5  
  
mean(x, TRUE) # match unnamed args to their position  
  
## Error in mean.default(x, TRUE): 'trim' must be numeric of length one
```

# Downloading, installing and loading packages

```
# A package has to be installed only once:  
install.packages("dplyr") # -> install package "dplyr" from CRAN  
# Load the package once per session:  
library(dplyr)
```

Alternatively, you can install and load packages in RStudio using the Packages tab.



To install a package that is hosted on GitHub use `remotes::install_github("<REPOSITORY>")`.

# Data Frames

- `data.frame` is R's data structure for a **table**.
- A data frame is a rectangular collection of data, arranged by **variables (columns)** and **observations (rows)**.

Columns of data frames are accessed with `$`:

```
dataframe$var_name
```

# The Tidyverse



Quote from the [Tidyverse website](#):

**"R packages for data science.** The tidyverse is an **opinionated collection of R packages designed for data science**. All packages share an underlying **design philosophy, grammar, and data structures**."

- collection of open-source `R` packages mainly for data wrangling and visualization
- shared conventions and common APIs across all Tidyverse packages

# R Markdown

**R Markdown** is a file format to combine **code**, the associated **results** and **narrative text** in a simple text file, to create **reproducible reports** which can be **flexibly distributed in multiple ways**.

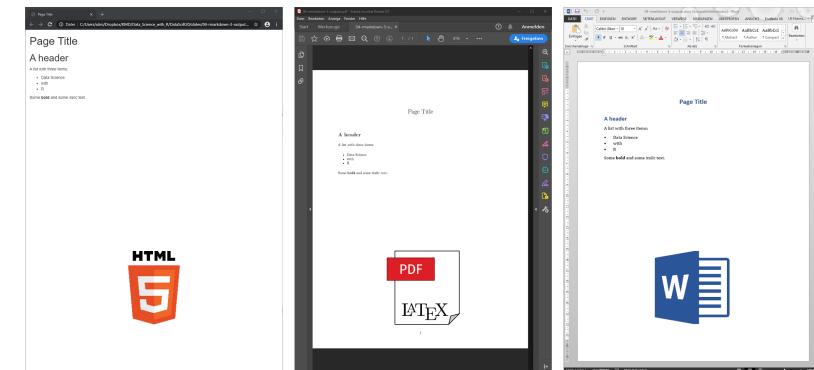
An R Markdown document is saved as `.Rmd` file. It contains both the (`R`) code and a prose description of a data analysis task. The R Markdown document can be rendered as HTML, PDF, Word and various other output formats.

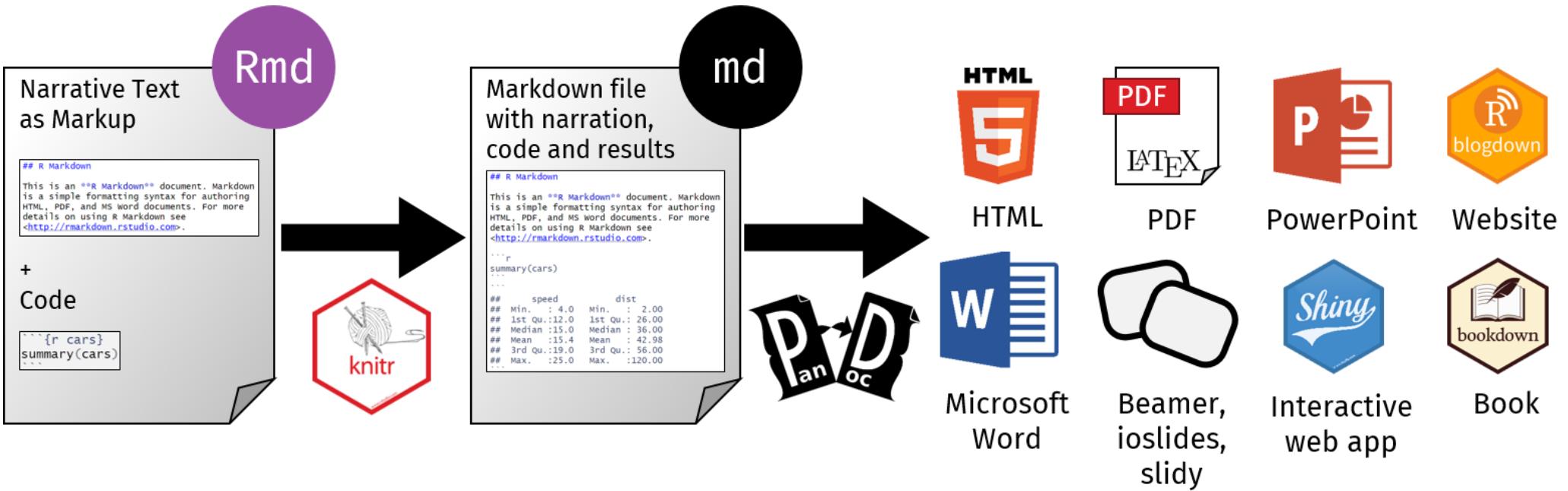
**Pros** of R Markdown documents:

- **reproducibility** (reduce **copy&paste**)
- **simple Markdown syntax** for text
- a **single source document** that can be rendered for different target audiences and purposes
- **simple, future-proof file format** that can be managed by a version control system like Git or SVN



<https://rmarkdown.rstudio.com/>





The image shows an R Markdown file named "rmarkdown\_intro.Rmd" in a code editor. The file contains the following structure:

```
1 ---  
2 title: "RMarkdown Introduction"  
3 date: 2018-07-30  
4 output: html_document  
5 ---  
6  
7 ```{r setup, include = FALSE}  
8 library(readr)  
9 library(ggplot2)  
10 library(dplyr)  
11  
12 census <- read_rds("Data/german_census_2011_clean.rds")  
13 young <- census %>%  
14   filter(avg_age <= 45)  
15 ````  
16  
17 We have data about `r nrow(census)` districts.  
18 `r nrow(census) - nrow(young)` of them are older than  
19 45 years. The distribution of the remainder is shown  
20 below:  
21  
22 ```{r, echo = FALSE}  
23 young %>%  
24   ggplot(aes(avg_age)) +  
25   geom_density()  
26 ````
```

Annotations on the right side of the code editor identify the components:

- YAML header**: Points to the first five lines of the file.
- Code Chunk**: Points to the first code chunk (lines 7-15).
- Inline Code**: Points to the string `r nrow(census)` within the inline R code at line 17.
- Markdown text**: Points to the explanatory text at lines 17-20.
- Code Chunk**: Points to the second code chunk (lines 22-26).

# Tour: R Markdown

# Structure of an `.Rmd` file

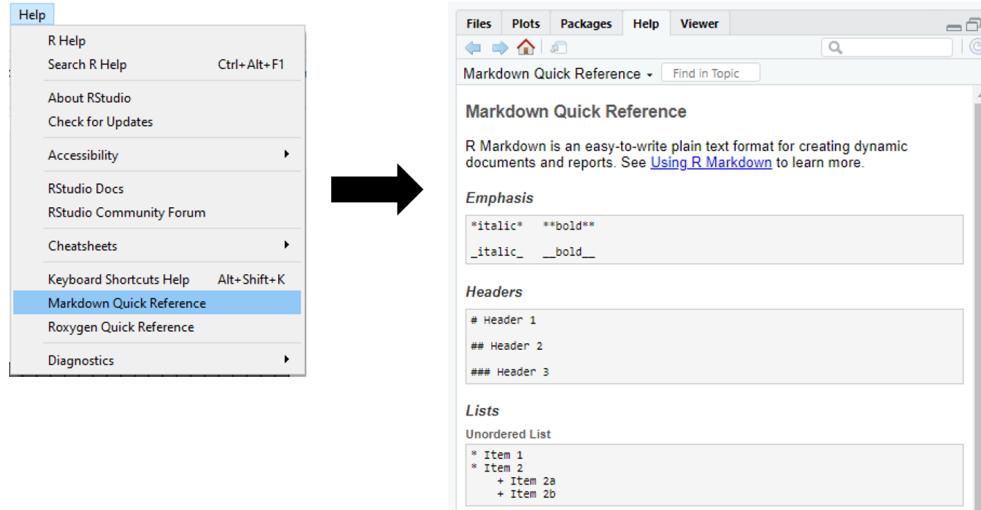
```
markdown_intro.Rmd x
Insert Run Knit ABC

1 ---  
2 title: "R Markdown Intro"  
3 date: 2018-07-30  
4 output:  
5   html_document:  
6     fig_height: 4  
7     fig_width: 5  
8 ---  
9  
10 ``{r analysis, message = FALSE}  
11 library(tidyverse)  
12 age_threshold <- 45  
13 census <- read_rds("Data/german_census_2011_clean.rds")  
14 (young <- census %>%  
15   filter(avg_age <= age_threshold))  
16 ``  
17  
18 We have data about `r nrow(census)` districts,  
19 `r nrow(census) - nrow(young)` of them with an  
20 average age above `r age_threshold` years. The  
21 distribution of the remainder is shown below:  
22  
23 ``{r, echo = FALSE}  
24 young %>%  
25   ggplot(aes(avg_age)) +  
26   geom_histogram(color = "black")  
27 ``
```

- **YAML header** (metadata)
  - enclosed by lines with three dashes
  - collection of key-value pairs separated by colons
- narrative text as **Markdown markup**
- **R code**
  - as code chunks surrounded by ````{r}` and `````
  - as inline code surrounded by ``r`` and ```

# R Markdown help

**Markdown Quick Reference:**  
Help → Markdown Quick Reference



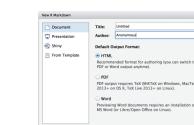
**R Markdown Cheat Sheet:**  
Help → Cheatsheets → R Markdown Cheat Sheet

## R Markdown :: CHEAT SHEET

### What is R Markdown?

- **.Rmd files** - An R Markdown (.Rmd) file is a record of your research. It contains the code that a scientist needs to reproduce your work along with the narration that a reader needs to understand your work.
- **Reproducible Research** - At the click of a button, or the type of a command, you can rerun the code in an R Markdown file to reproduce your work and export the results as a finished report.
- **Dynamic Documents** - You can choose to export the finished report in a variety of formats, including html, pdf, MS Word, or RTF documents; html or pdf based slides, Notebooks, and more.

### Workflow



- ① Open a new .Rmd file at File ► New File ► R Markdown. Use the wizard that opens to pre-populate the file with a template
- ② Write document by editing template
- ③ Knit document to create report; use knit button or render() to knit
- ④ Preview Output in IDE window
- ⑤ Publish (optional) to web server
- ⑥ Examine build log in R Markdown console
- ⑦ Use output file that is saved along side .Rmd

A screenshot of the RStudio interface. The top menu bar shows 'File', 'Edit', 'Code', 'View', 'Plots', 'Session', 'Build', 'Debug', 'Tools', and 'Help'. A tooltip over the 'File' menu shows options like 'report.Rmd', 'run code chunk(s)', 'publish', 'show outline', 'run all previous chunks', 'modify chunk options', 'run current chunk', and 'for more http://'. The main workspace shows an R Markdown file 'report.Rmd' with code like `# R Markdown` and `knitr::opts\_chunk\$set(echo = TRUE)`. Below it is a 'Console' window with commands like `library(rmarkdown)` and `render("report.Rmd", output\_file = "report.html")`. The bottom navigation bar includes 'Files', 'New I...', 'Home', 'Recent', 'File', 'Edit', 'View', 'Plots', 'Session', 'Build', 'Debug', 'Tools', and 'Help'.

### render

Use `rmarkdown::render()` to render/knit at cmd line. Important args:

<code>input</code> - file to render	<code>output_options</code> - List of render options (as in YAML)	<code>output_file</code>	<code>params</code> - params to
<code>output_format</code>		<code>output_dir</code>	

# Environments

⚠️ Each R Markdown document has its own environment.

→ Consequence: Objects from the global environment are unavailable in the .Rmd document when knitting.

## Example:

1. Run the following code in the console:

```
x <- 5  
x * 3
```

😊 All good!

2. Add the following code within a code chunk to your .Rmd file and knit the document:

```
x * 3
```

❗ Knitting fails!

# How will we use R Markdown?

- Every exercise comes with an R Markdown document which contains some code scaffolding.
- You submit your project report as an R Markdown document.
- The course slides are made with R Markdown.



# Thank you! Questions?