Reproducible Workflow

Ponrawee Prasertsom & Georgia-Ann Carter

PPLS Open Research Facilitators

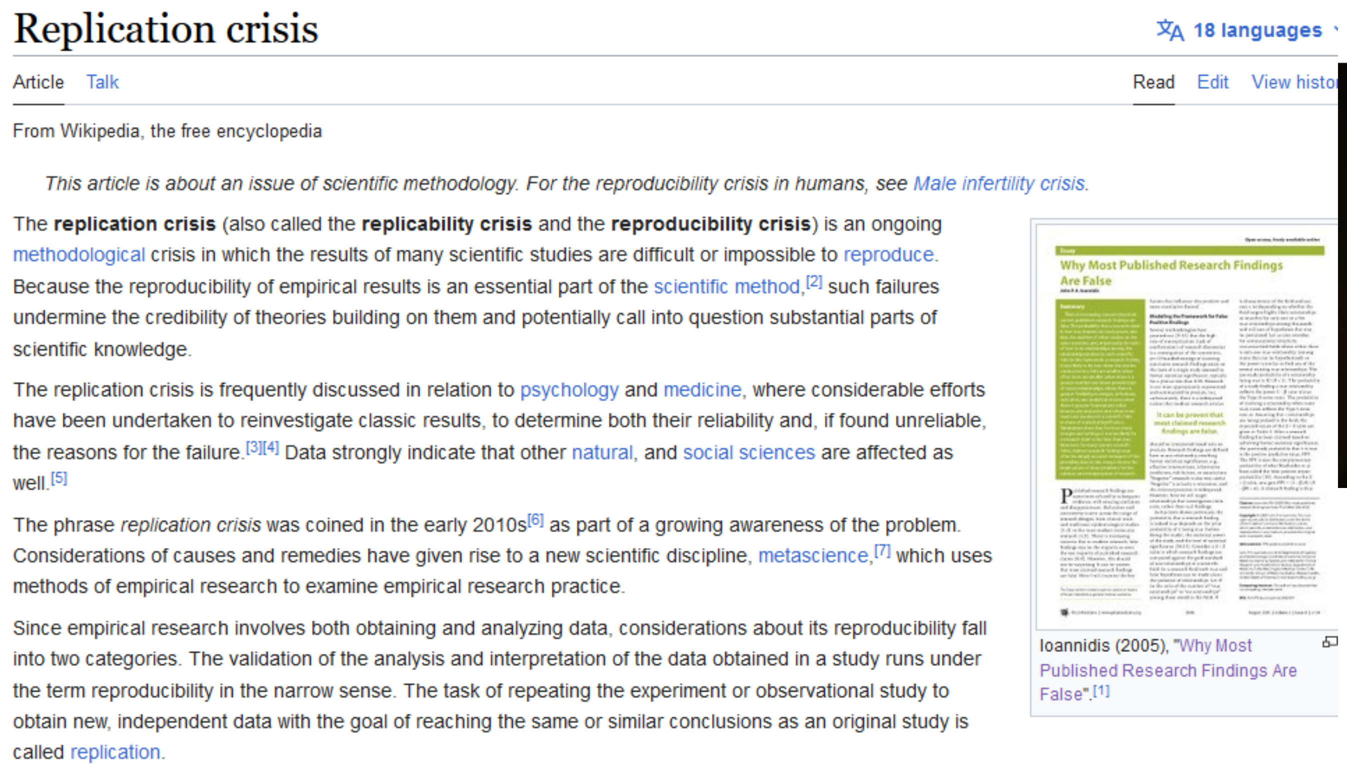
https://pplsopenresearch.github.io

Open Science Collaboration (2015)

SCIENCE

Psychology's Replication Crisis Is Running Out of Excuses

Another big project has found that only half of studies can be repeated. And this time, the usual explanations fall flat.



Replication crisis

From Wikipedia, the free encyclopedia

This article is about an issue of scientific methodology. For the reproducibility crisis in humans, see Male infertility crisis.

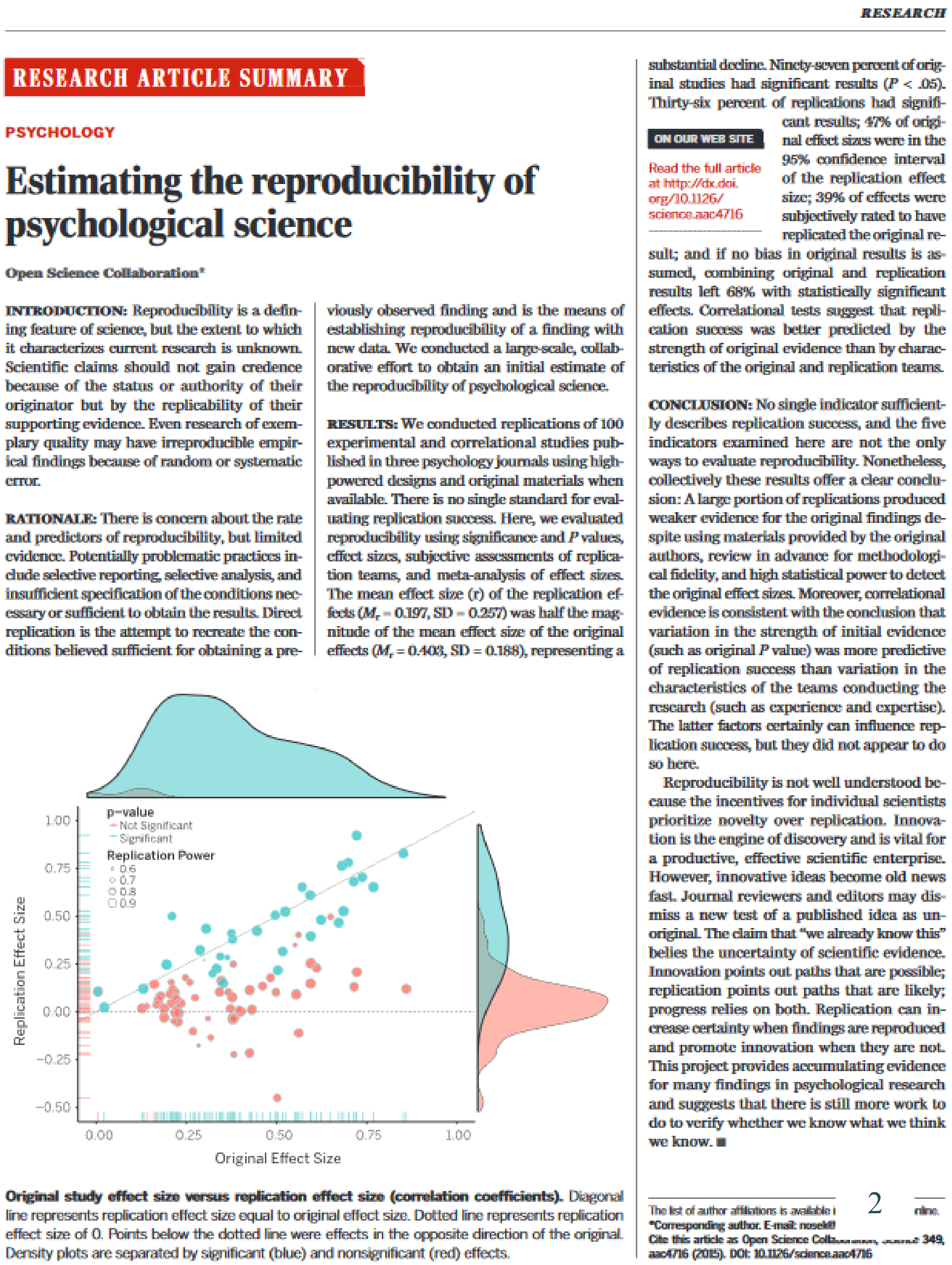
The **replication crisis** (also called the **replicability crisis** and the **reproducibility crisis**) is an ongoing methodological crisis in which the results of many scientific studies are difficult or impossible to reproduce. Because the reproducibility of empirical results is an essential part of the scientific method,[2] such failures undermine the credibility of theories building on them and potentially call into question substantial parts of scientific knowledge.

The replication crisis is frequently discussed in relation to psychology and medicine, where considerable efforts have been undertaken to reinvestigate classic results, to determine both their reliability and, if found unreliable, the reasons for the failure.[3][4] Data strongly indicate that other natural, and social sciences are affected as well.[5]

The phrase *replication crisis* was coined in the early 2010s[6] as part of a growing awareness of the problem. Considerations of causes and remedies have given rise to a new scientific discipline, metascience,[7] which uses methods of empirical research to examine empirical research practice.

Since empirical research involves both obtaining and analyzing data, considerations about its reproducibility fall into two categories. The validation of the analysis and interpretation of the data obtained in a study runs under the term reproducibility in the narrow sense. The task of repeating the experiment or observational study to obtain new, independent data with the goal of reaching the same or similar conclusions as an original study is called replication.

Ioannidis (2005), "Why Most Published Research Findings Are False".[1]



RESEARCH

RESEARCH ARTICLE SUMMARY

PSYCHOLOGY

Estimating the reproducibility of psychological science

Open Science Collaboration*

INTRODUCTION: Reproducibility is a defining feature of science, but the extent to which it characterizes current research is unknown. Scientific claims should not gain credence because of the status or authority of their originator but by the replicability of their supporting evidence. Even research of exemplary quality may have irreproducible empirical findings because of random or systematic error.

RATIONALE: There is concern about the rate and predictors of reproducibility, but limited evidence. Potentially problematic practices include selective reporting, selective analysis, and insufficient specification of the conditions necessary or sufficient to obtain the results. Direct replication is the attempt to recreate the conditions believed sufficient for obtaining a previously observed finding and is the means of establishing reproducibility of a finding with new data. We conducted a large-scale, collaborative effort to obtain an initial estimate of the reproducibility of psychological science.

RESULTS: We conducted replications of 100 experimental and correlational studies published in three psychology journals using high-powered designs and original materials when available. There is no single standard for evaluating replication success. Here, we evaluated reproducibility using significance and P values, effect sizes, subjective assessments of replication teams, and meta-analysis of effect sizes. The mean effect size (r) of the replication effects (M_r = 0.197, SD = 0.257) was half the magnitude of the mean effect size of the original effects (M_r = 0.403, SD = 0.188), representing a substantial decline. Ninety-seven percent of original studies had significant results ($P < .05$). Thirty-six percent of replications had significant results; 47% of original effect sizes were in the 95% confidence interval of the replication effect size; 39% of effects were subjectively rated to have replicated the original result; and if no bias in original results is assumed, combining original and replication results left 68% with statistically significant effects. Correlational tests suggest that replication success was better predicted by the strength of original evidence than by characteristics of the original and replication teams.

ON OUR WEB SITE

Read the full article at http://dx.doi.org/10.1126/science.aac4716

CONCLUSION: No single indicator sufficiently describes replication success, and the five indicators examined here are not the only ways to evaluate reproducibility. Nonetheless, collectively these results offer a clear conclusion: A large portion of replications produced weaker evidence for the original findings despite using materials provided by the original authors, review in advance for methodological fidelity, and high statistical power to detect the original effect sizes. Moreover, correlational evidence is consistent with the conclusion that variation in the strength of initial evidence (such as original P value) was more predictive of replication success than variation in the characteristics of the teams conducting the research (such as experience and expertise). The latter factors certainly can influence replication success, but they did not appear to do so here.

Reproducibility is not well understood because the incentives for individual scientists prioritize novelty over replication. Innovation is the engine of discovery and is vital for a productive, effective scientific enterprise. However, innovative ideas become old news fast. Journal reviewers and editors may dismiss a new test of a published idea as unoriginal. The claim that "we already know this" belies the uncertainty of scientific evidence. Innovation points out paths that are possible; replication points out paths that are likely; progress relies on both. Replication can increase certainty when findings are reproduced and promote innovation when they are not. This project provides accumulating evidence for many findings in psychological research and suggests that there is still more work to do to verify whether we know what we think we know. ■

Original study effect size versus replication effect size (correlation coefficients). Diagonal line represents replication effect size equal to original effect size. Dotted line represents replication effect size of 0. Points below the dotted line were effects in the opposite direction of the original. Density plots are separated by significant (blue) and nonsignificant (red) effects.

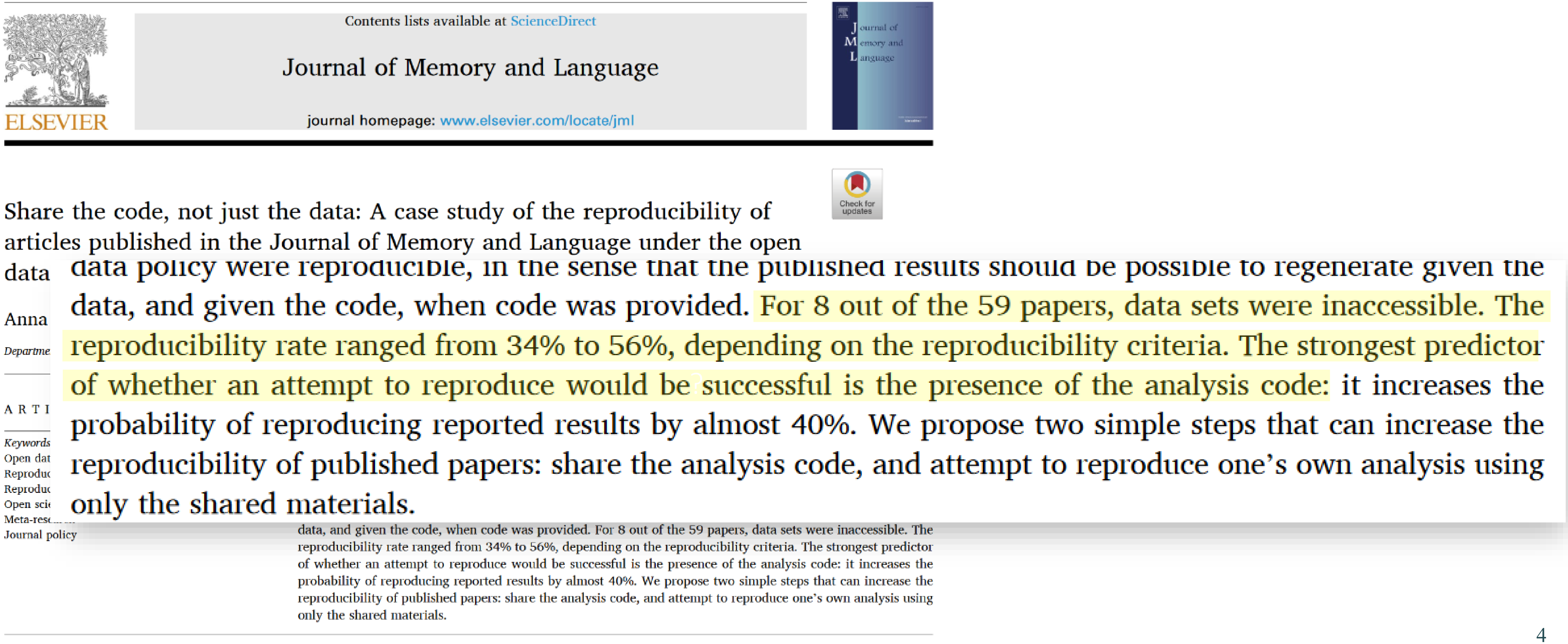
The list of author affiliations is available i nline.
*Corresponding author. E-mail: nosek@
Cite this article as Open Science Colla 349, aac4716 (2015). DOI: 10.1126/science.aac4716

Replication = New data, Same analysis

Reproduction = Same data, Same analysis

Laurinavichyute, Yadav & Vasishth (2022)

Reproduction is easy?



Contents lists available at ScienceDirect

Journal of Memory and Language

journal homepage: www.elsevier.com/locate/jml

Share the code, not just the data: A case study of the reproducibility of articles published in the Journal of Memory and Language under the open data

Anna

Departme

A R T I

Keywords
Open dat
Reproduc
Reproduc
Open scie
Meta-rese
Journal policy

data policy were reproducible, in the sense that the published results should be possible to regenerate given the data, and given the code, when code was provided. For 8 out of the 59 papers, data sets were inaccessible. The reproducibility rate ranged from 34% to 56%, depending on the reproducibility criteria. The strongest predictor of whether an attempt to reproduce would be successful is the presence of the analysis code: it increases the probability of reproducing reported results by almost 40%. We propose two simple steps that can increase the reproducibility of published papers: share the analysis code, and attempt to reproduce one's own analysis using only the shared materials.

data, and given the code, when code was provided. For 8 out of the 59 papers, data sets were inaccessible. The reproducibility rate ranged from 34% to 56%, depending on the reproducibility criteria. The strongest predictor of whether an attempt to reproduce would be successful is the presence of the analysis code: it increases the probability of reproducing reported results by almost 40%. We propose two simple steps that can increase the reproducibility of published papers: share the analysis code, and attempt to reproduce one's own analysis using only the shared materials.

```
Model failed to converge with max|grad| = 0.209594 (tol = 0.002,
component 1
```

“OK, I will change the maximum iterations to 30000 and see if that helps with fitting.”

```
control = glmerControl(maxfun = 30000)
```

“Great! Now it converges and I can get on with my life.”

Method for obtaining *p*-values

No. of iterations

Variable coding

Optimiser

Software version

Method for computing confidence intervals

Data and code sharing

If you share the code along with the data, other people can just run it (and can examine it).

We can do even better than just sharing data and code!

We can structure **our whole project** with reproduction in mind.

We can have a reproducible workflow.

Reproducible Workflow

Aim: **Anyone** can reproduce your results

Including **you**!

https://www.youtube.com/watch?v=s3JldKoA0zw

Reproducible Workflow

Aim: **Anyone** can reproduce your results

Including **you**! (Good because redoing analyses is almost inevitable!)

Kitzes (2018)

Key Practices

File/folder structures and descriptions

Automation and documentation of manual steps

Flow design (How you break the process into steps)

This Workshop

1. Folder and files (Directory organisation and flow design)
2. README (Directory organisation)
3. RMarkdown (Automation and documentation)

Directory (folder) structure

Organise your folders and files to reflect your research process

- 🧪 Data acquisition: Get raw data from experiments, use existing data, …
- 🖥️ Data processing: Anonymise participants, fix missing values, …
- 📊 Data analysis: Plot figures, build a stats model, …

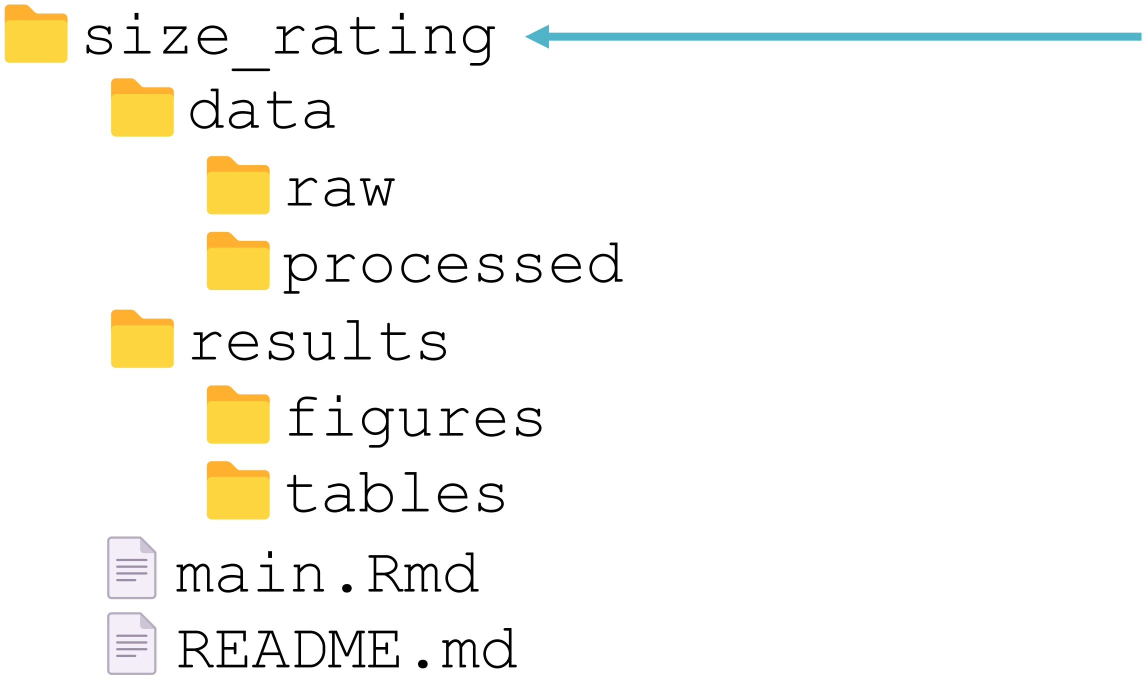
Directory (folder) structure

```
📁 size_rating
    📁 data
        📁 raw
        📁 processed
    📁 results
        📁 figures
        📁 tables
    📄 main.Rmd
    📄 README.md
```

Practice: Create these folders

(Ignore the files for the moment)

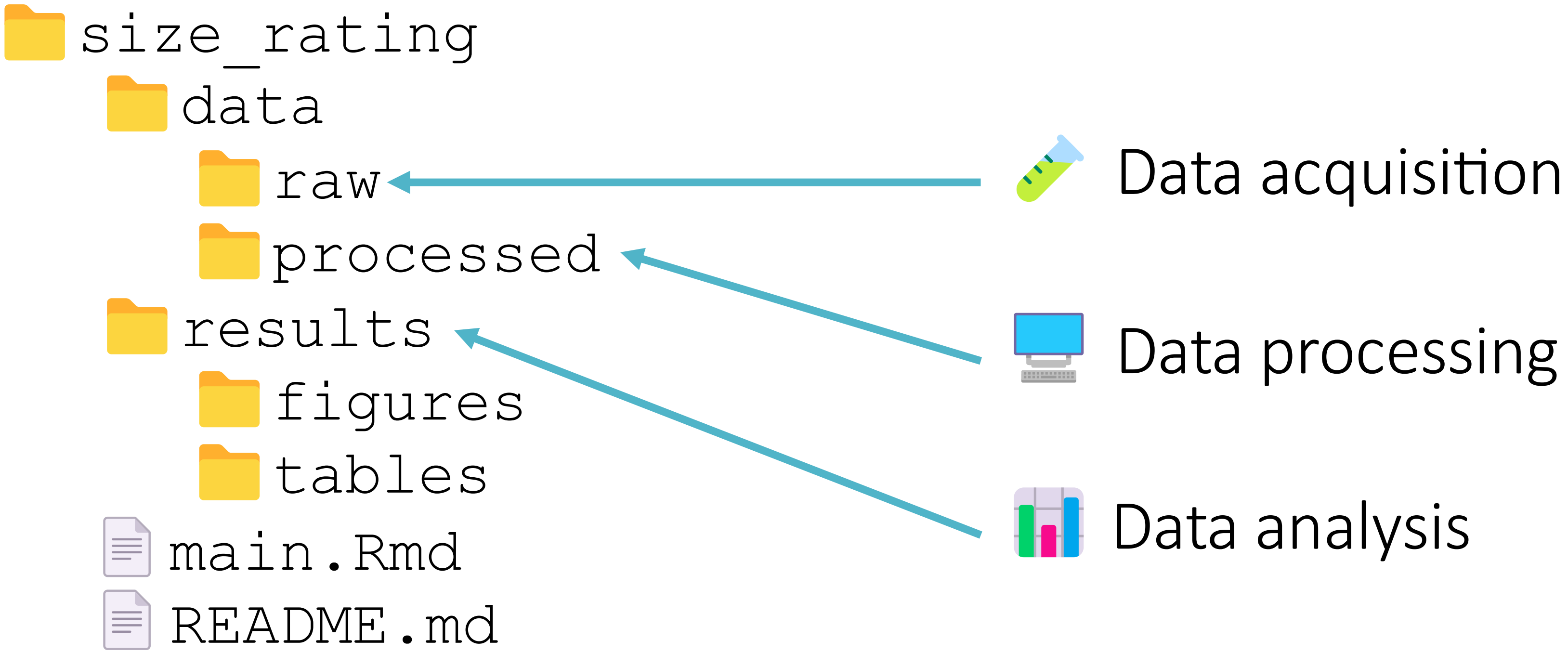
Directory (folder) structure



```
size_rating
    data
        raw
        processed
    results
        figures
        tables
    main.Rmd
    README.md
```

Study title (use _ or – in place of spaces)

Directory (folder) structure



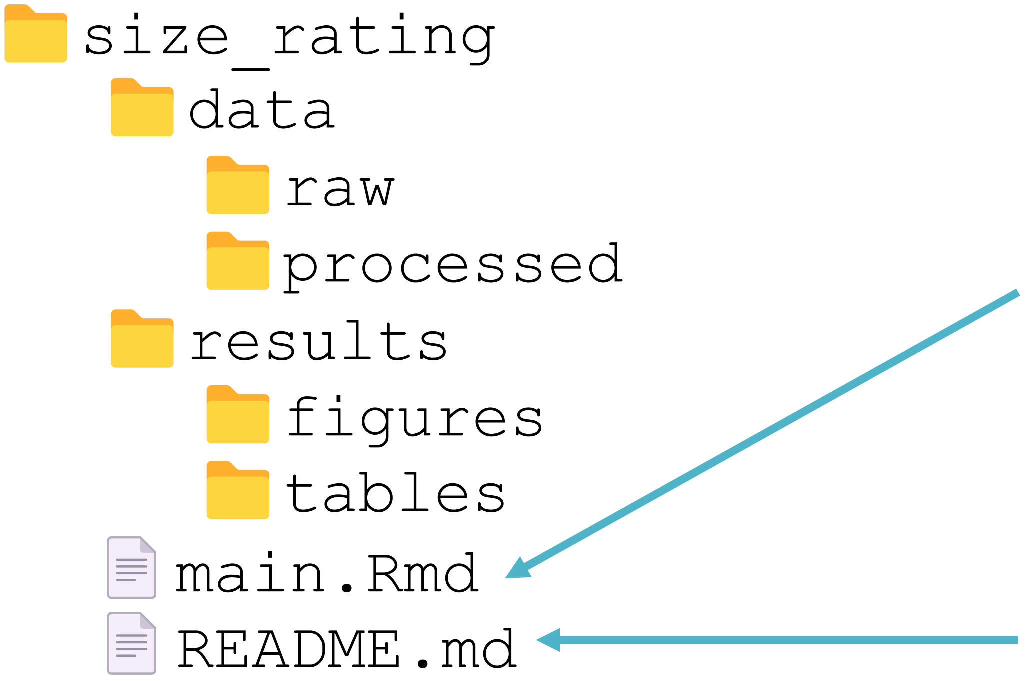
```
size_rating
    data
        raw
        processed
    results
        figures
        tables
    main.Rmd
    README.md
```

Data acquisition

Data processing

Data analysis

Directory (folder) structure

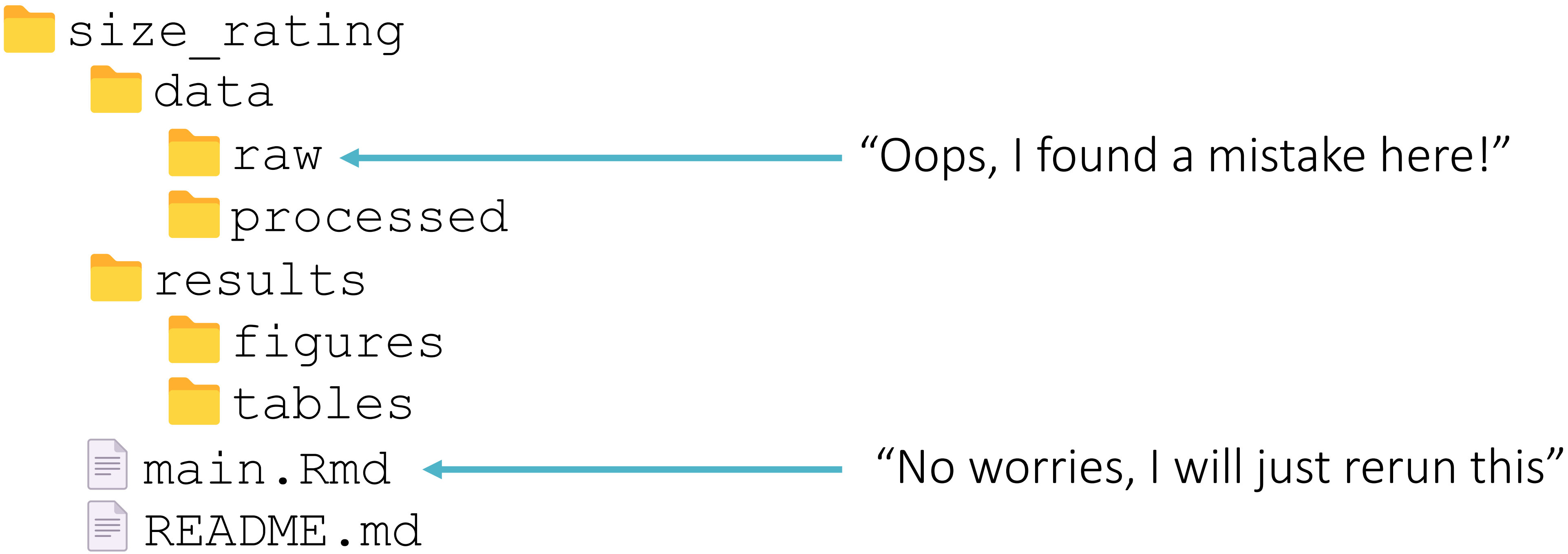


```
size_rating
    data
        raw
        processed
    results
        figures
        tables
    main.Rmd
    README.md
```

Code for processing, analysing, plotting, …
(saves results into the folders above)

Descriptions of the project, folders and files

Directory (folder) structure



```
size_rating
    data
        raw
        processed
    results
        figures
        tables
    main.Rmd
    README.md
```

“Oops, I found a mistake here!”

“No worries, I will just rerun this”

README.md

README is a text file that provides important information, such as:

1. Information about the directory structure (what files are where)
2. Information about the files (what does `main.Rmd` do?)
3. Software dependencies (e.g., which R packages you use? Which version?)
4. Steps to reproduce (what do I do if I want to reproduce the tables, figures, statistics?)

README.md

README is usually written as a Markdown file (.md)

1. # in front of headings (## for subheadings, ### for sub-subheadings, etc.)
2. Hyphens (-) for bullet points
3. 1., 2., 3., … for numbered lists
4. Backticks (`) for marking text as code (usually rendered as text highlighted in gray)
5. *…* for italic text, **…** for bold text

README.md

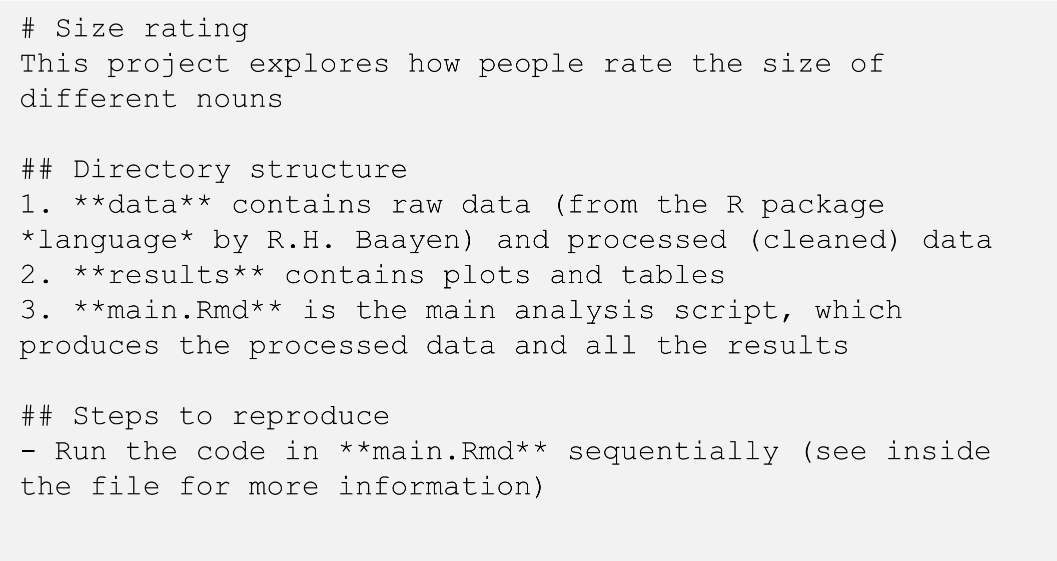
```
📁size_rating
  📁data
    📁raw
    📁processed
  📁results
    📁figures
    📁tables
  📄main.Rmd
  📄README.md
```

```
# Size rating
This project explores how people rate the size of different
nouns

## Directory structure
1. **data** contains raw data (from the R package *language*
by R.H. Baayen) and processed (cleaned) data
2. **results** contains plots and tables
3. **main.Rmd** is the main analysis script, which produces
the processed data and all the results

## Steps to reproduce
- Run the code in **main.Rmd** sequentially (see inside the
file for more information)
```

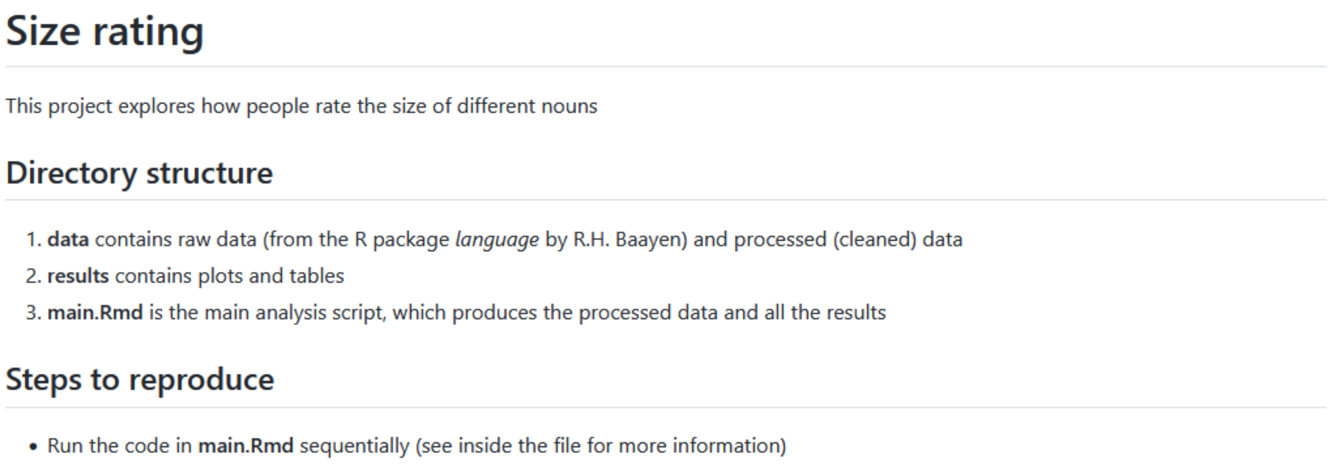
README.md

- Markdown rendered on Github (where you likely share your code and results)

- Readable even when not rendered



```
# Size rating
This project explores how people rate the size of
different nouns

## Directory structure
1. **data** contains raw data (from the R package
*language* by R.H. Baayen) and processed (cleaned) data
2. **results** contains plots and tables
3. **main.Rmd** is the main analysis script, which
produces the processed data and all the results

## Steps to reproduce
- Run the code in **main.Rmd** sequentially (see inside
the file for more information)
```



Size rating

This project explores how people rate the size of different nouns

Directory structure

1. **data** contains raw data (from the R package *language* by R.H. Baayen) and processed (cleaned) data
2. **results** contains plots and tables
3. **main.Rmd** is the main analysis script, which produces the processed data and all the results

Steps to reproduce

- Run the code in **main.Rmd** sequentially (see inside the file for more information)

README.md

You should also have separate **README for your data** describing what each of the columns is

```
📁 data
    📁 raw
        📄 data.csv
        📄 README.md
```

README.md

data/raw/data.csv

p_id	rt	response	buttons
1	100	0	[“a”, “b”]
1	200	1	[“b”, “a”]
2	150	0	[“a”, “b”]
2	120	0	[“a”, “a”]

data/raw/README.md

```
# Timed button response data
`data.csv` contains the data we collected

## Column descriptions

- **p_id**: the ID unique to each participant
- **rt**: response time (milliseconds)
- **response**: which button the participant clicks
(0 = left, 1 = right)
- **buttons**: button labels on the left and right
button of the format [“left label”, “right label”]
```

Directory (folder) structure

```
📁 size_rating
    📁 data
        📁 raw
            📄 data.csv
            📄 README.md
        📁 processed
            📄 processed.csv
            📄 README.md
    📁 results
        📁 figures
        📁 tables
    📄 main.Rmd
    📄 README.md
```

RMarkdown

- Lets you write and execute R code in “chunks” between your texts, images, etc.

- Easier to follow than pure R code with comments because the code chunks track your thought process

- Good for reproducibility because the reproducer will be able to follow the steps easily

- Can be turned into reports (HTML webpage, PDF, …)

RMarkdown

- You already know RMarkdown!
- If you know R and you know Markdown, it’s literally just putting those two things together.

Demo

Dataset: `sizeRatings` (from `languageR` package; Baayen & Hay, 2004)

Contains ratings of how big 81 different nouns (animal/plan) are from 38 different participants

Demo

Goal: Using Rmarkdown, we will

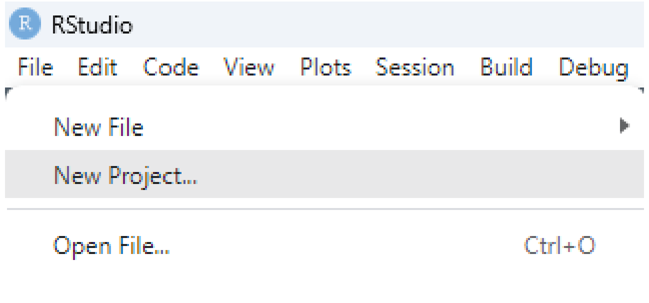
1. Process the data and replace participants' names with IDs
2. Find out which animal/plant receives the highest mean size rating
3. Plot the ratings of the 10 nouns with highest mean size ratings
4. Generate a webpage that renders the Markdown and contains the results of code execution
5. (If there's time left): Fit a linear regression model and generate a table of the model summary. Generate a PDF from RMarkdown

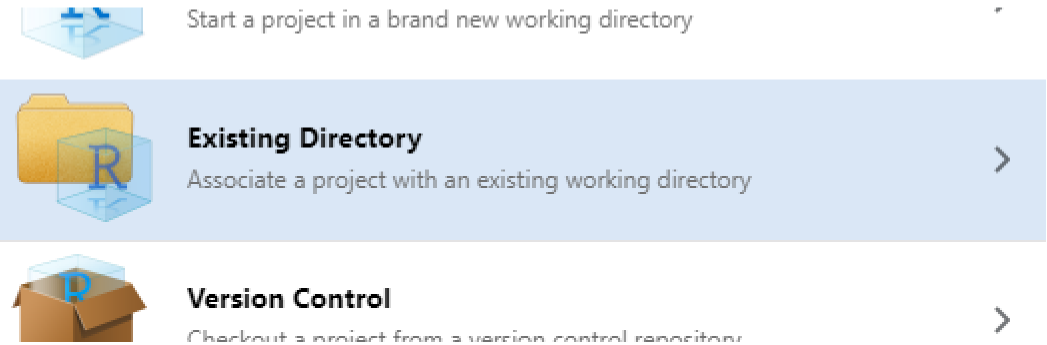
Code/copy along

There will be quite a bit of coding soon, you can go to https://bit.ly/ppls-reproducible to copy-paste parts that you don’t want to type yourself.

Demo

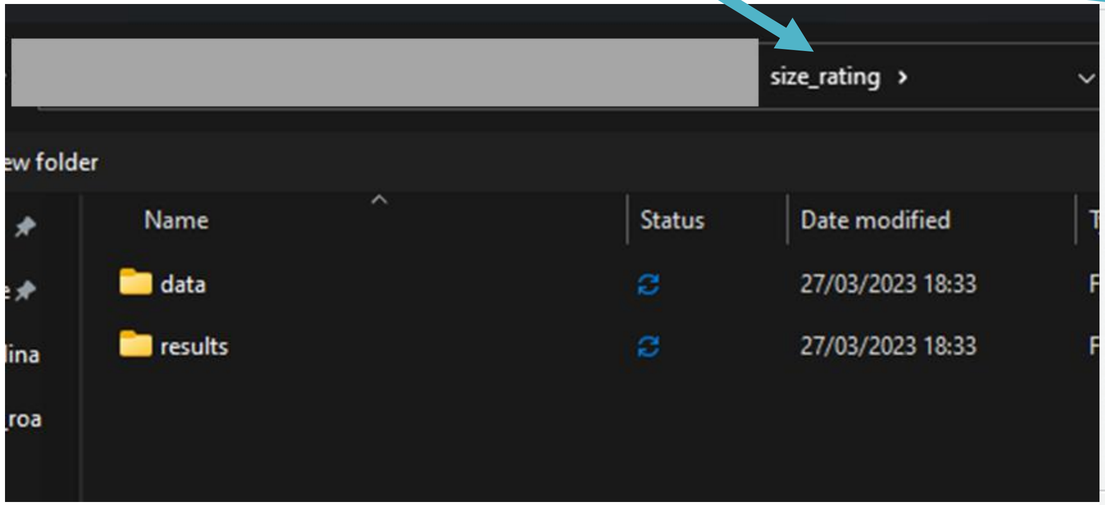
https://bit.ly/ppls-reproducible





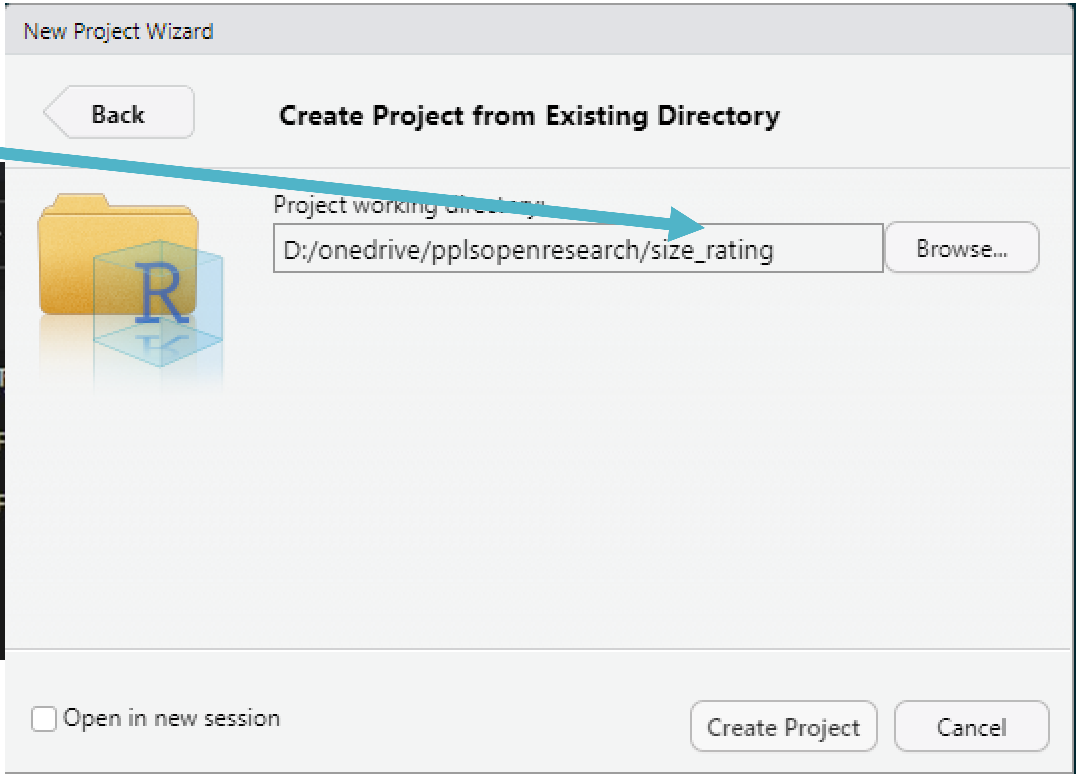
Start a project in a brand new working directory
Existing Directory
Associate a project with an existing working directory
Version Control
Checkout a project from a version control repository

Make sure it's **size_rating** here!



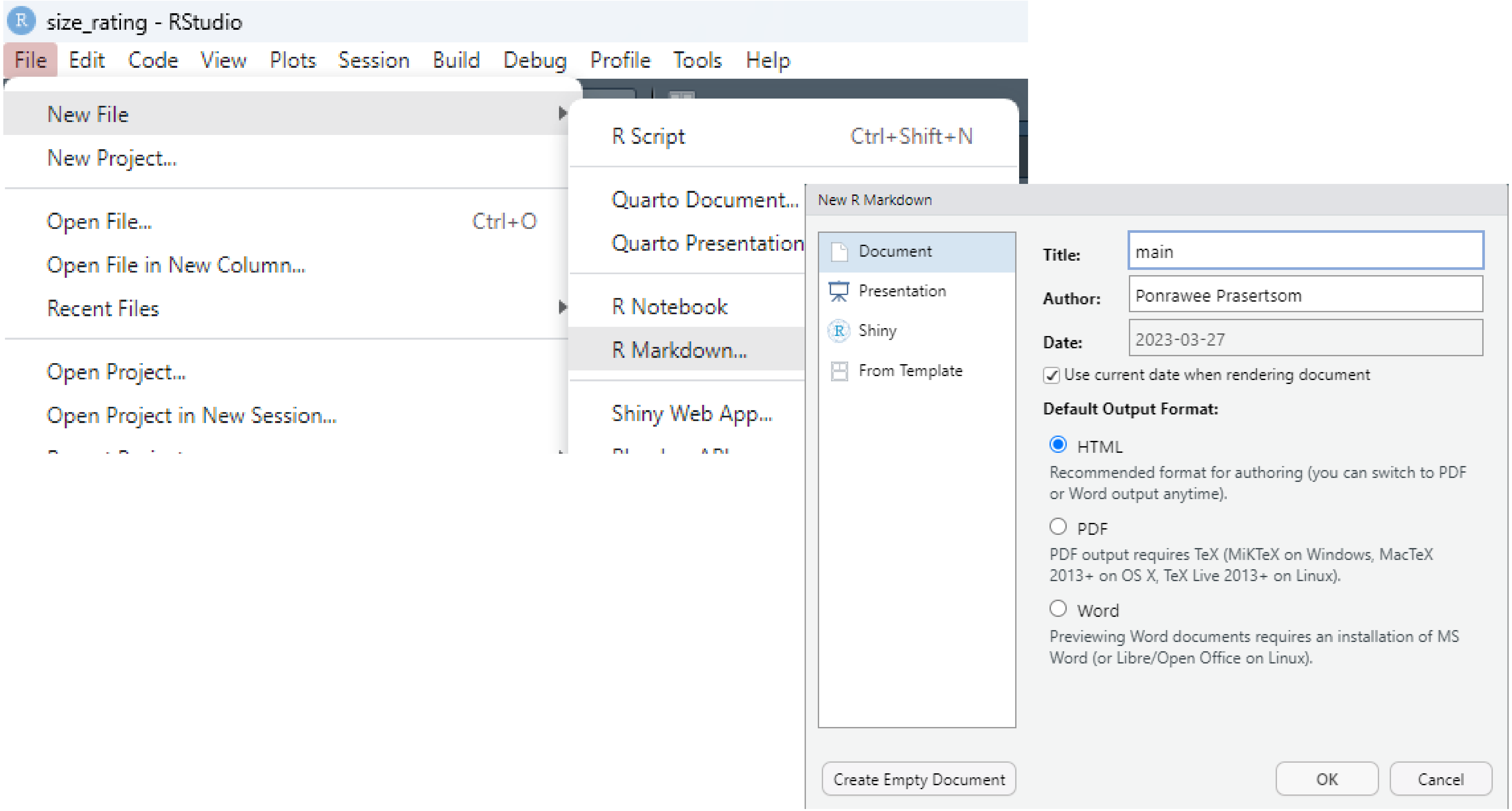
[redacted] size_rating >

Name	Status	Date modified	T
data		27/03/2023 18:33	F
results		27/03/2023 18:33	F



Project working directory:
D:/onedrive/pplsopenresearch/size_rating

□ Open in new session



Title: main

Author: Ponrawee Prasertsom

Date: 2023-03-27

☑ Use current date when rendering document

Default Output Format:

● HTML

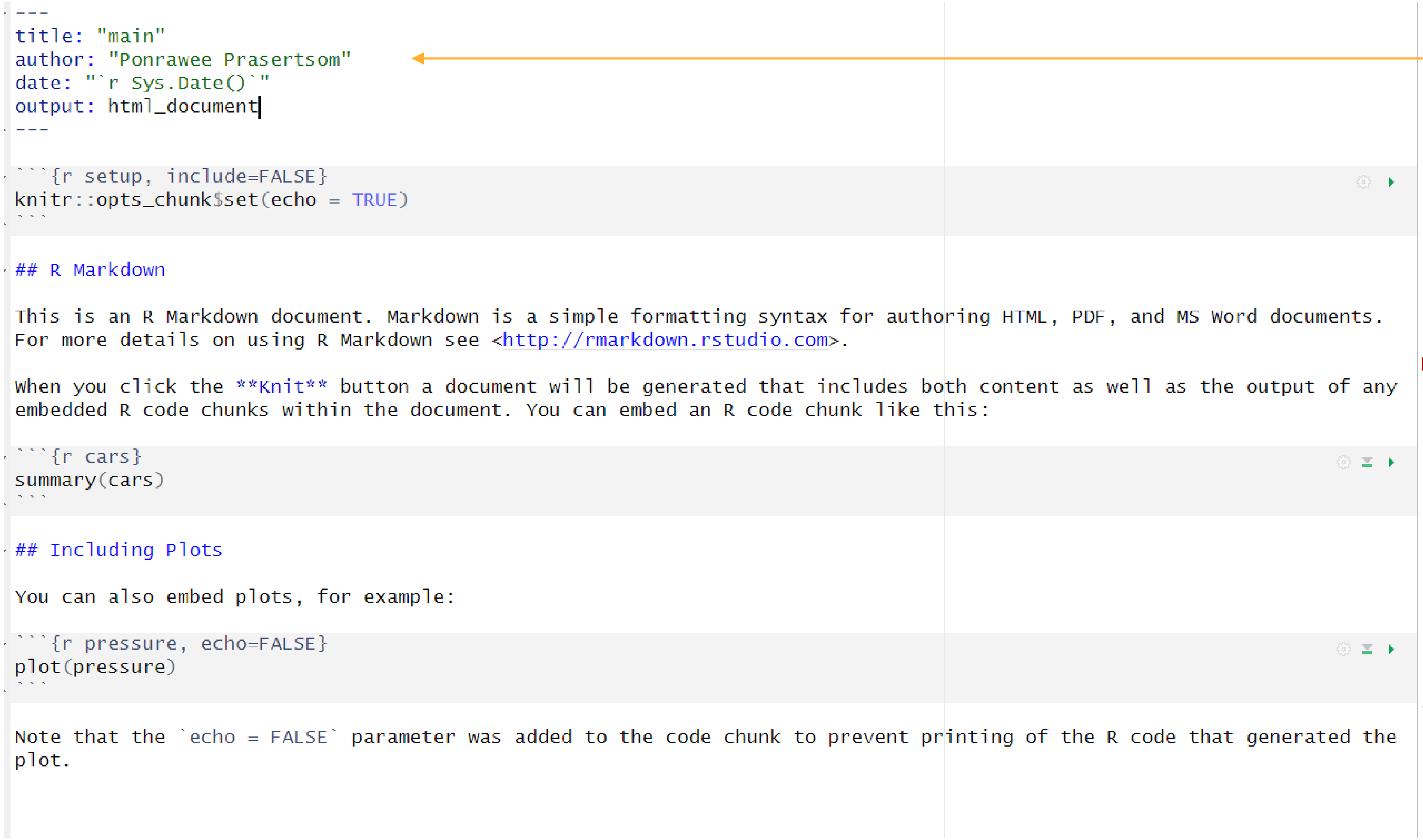
Recommended format for authoring (you can switch to PDF or Word output anytime).

○ PDF

PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux).

○ Word

Previewing Word documents requires an installation of MS Word (or Libre/Open Office on Linux).



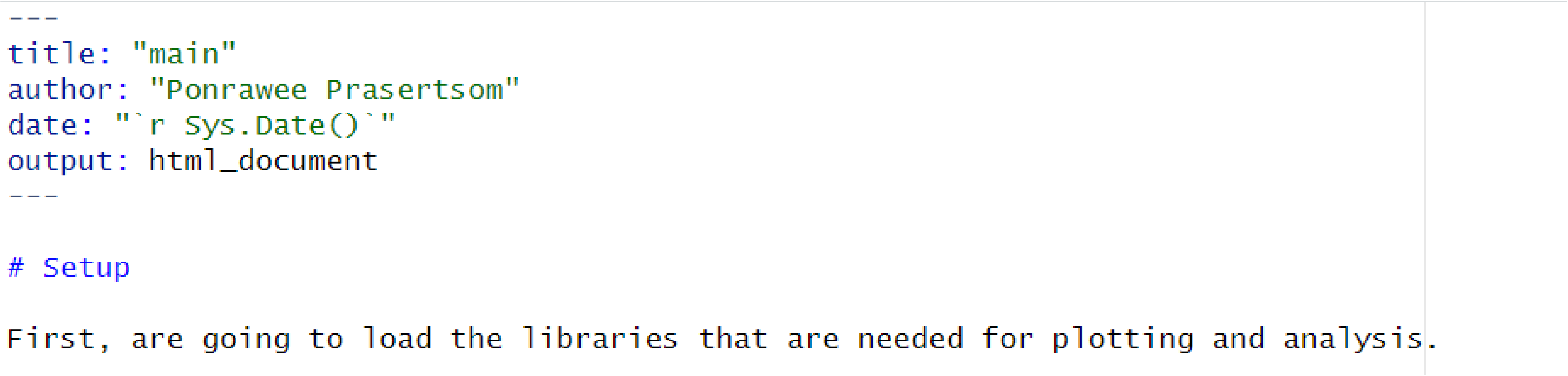
~~~
---
title: "main"
author: "Ponrawee Prasertsom"
date: "`r Sys.Date()`"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents.
For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any
embedded R code chunks within the document. You can embed an R code chunk like this:

```{r cars}
summary(cars)
```

## Including Plots

You can also embed plots, for example:

```{r pressure, echo=FALSE}
plot(pressure)
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the
plot.
~~~

YAML header (for settings)

R code chunks (click the green play button or Ctrl+Alt+C to execute)

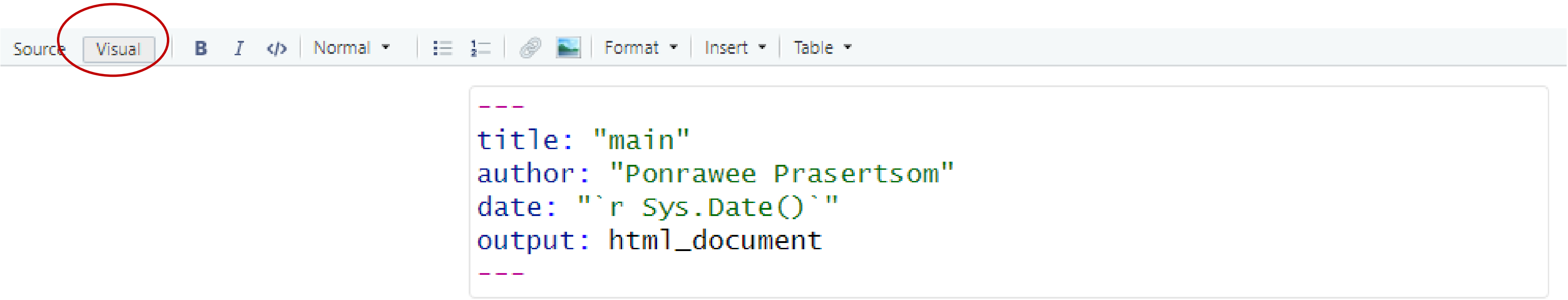
Markdown text

Clear the content, except the header, and create the setup section

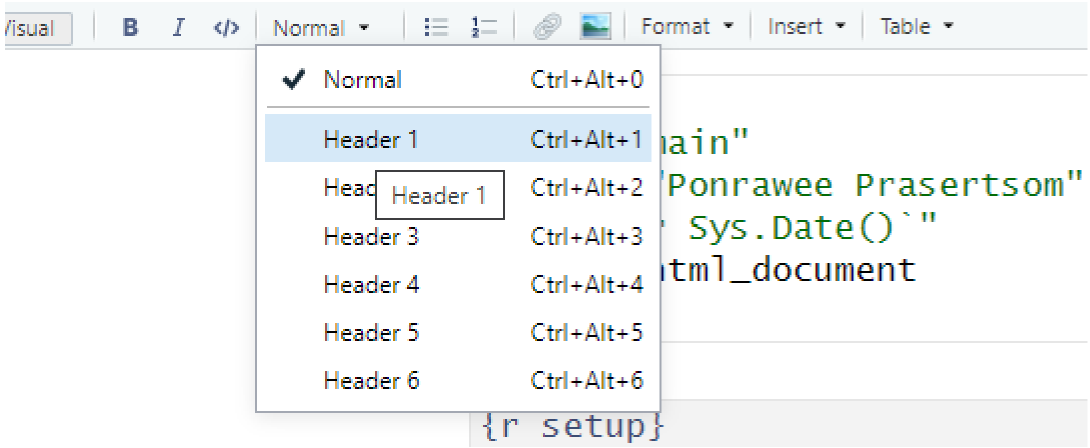


```
---
title: "main"
author: "Ponrawee Prasertsom"
date: "`r Sys.Date()`"
output: html_document
---

# Setup

First, are going to load the libraries that are needed for plotting and analysis.
```

If you’re not familiar with Markdown, use the **visual editor**. (If you’re copy-pasting, you must do it in the source mode!)

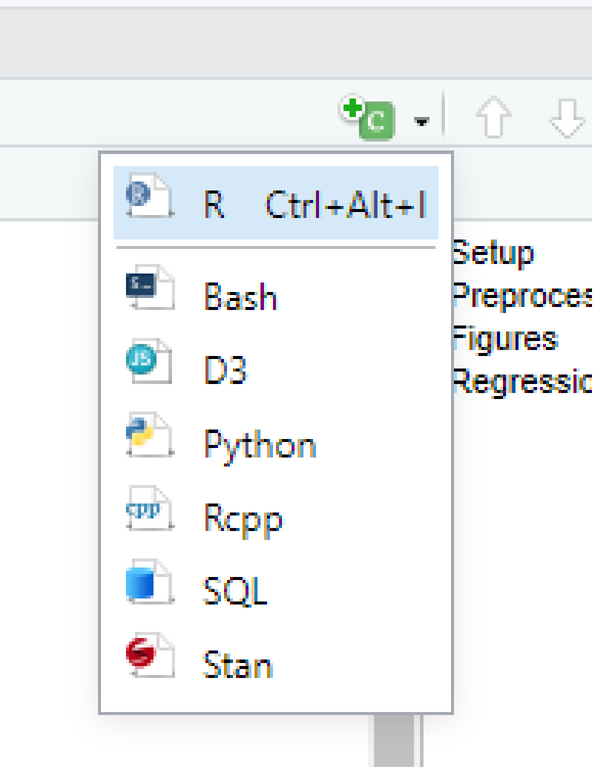


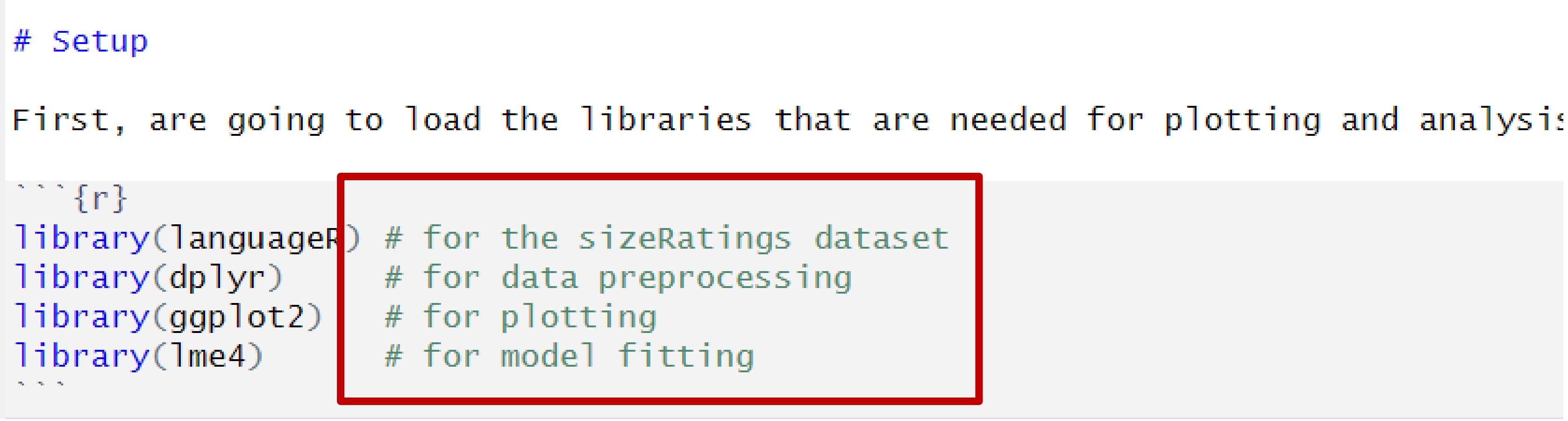
```
---
title: "main"
author: "Ponrawee Prasertsom"
date: "`r Sys.Date()`"
output: html_document
---
```



```
ain"
'Ponrawee Prasertsom"
' Sys.Date()`"
tml_document
{r setup}
```

Add an R code chunk and write code to import libraries

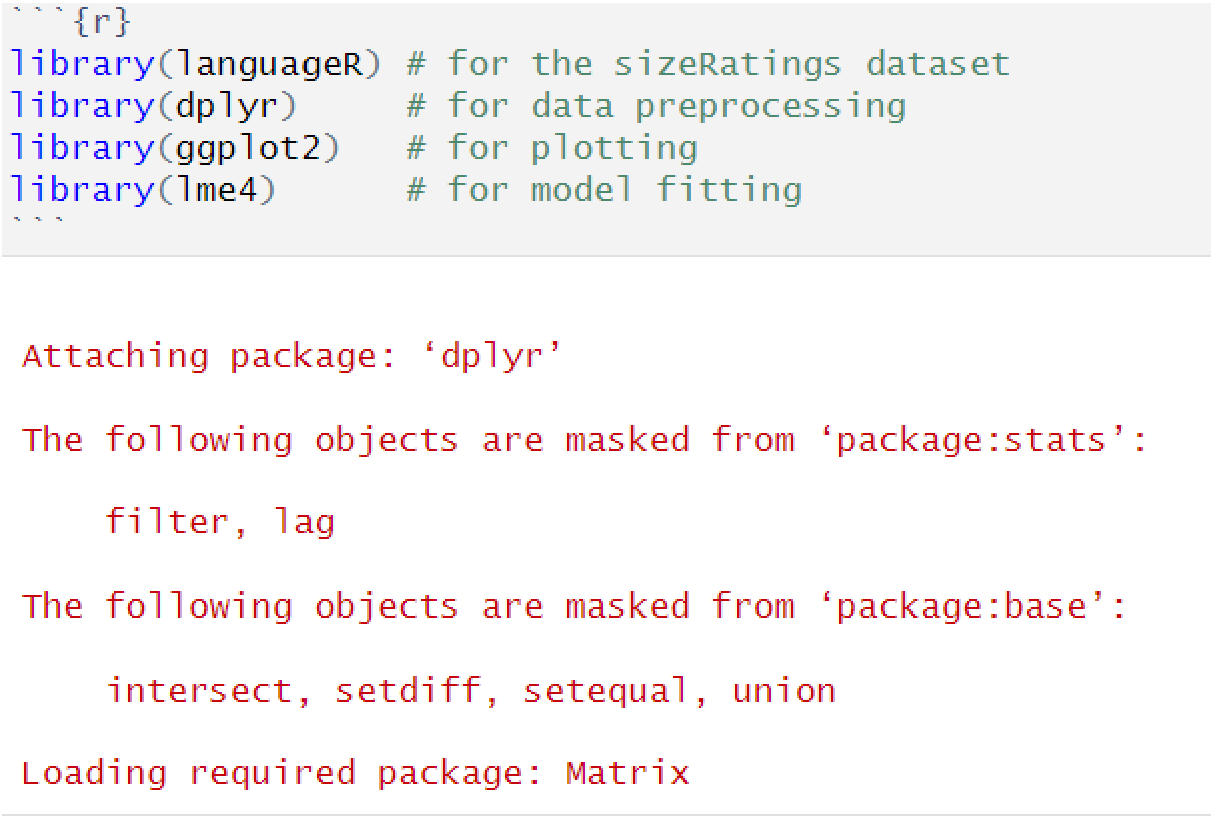




~~~
# Setup

First, are going to load the libraries that are needed for plotting and analysis

```{r}
library(languageR) # for the sizeRatings dataset
library(dplyr)     # for data preprocessing
library(ggplot2)   # for plotting
library(lme4)      # for model fitting
```
~~~

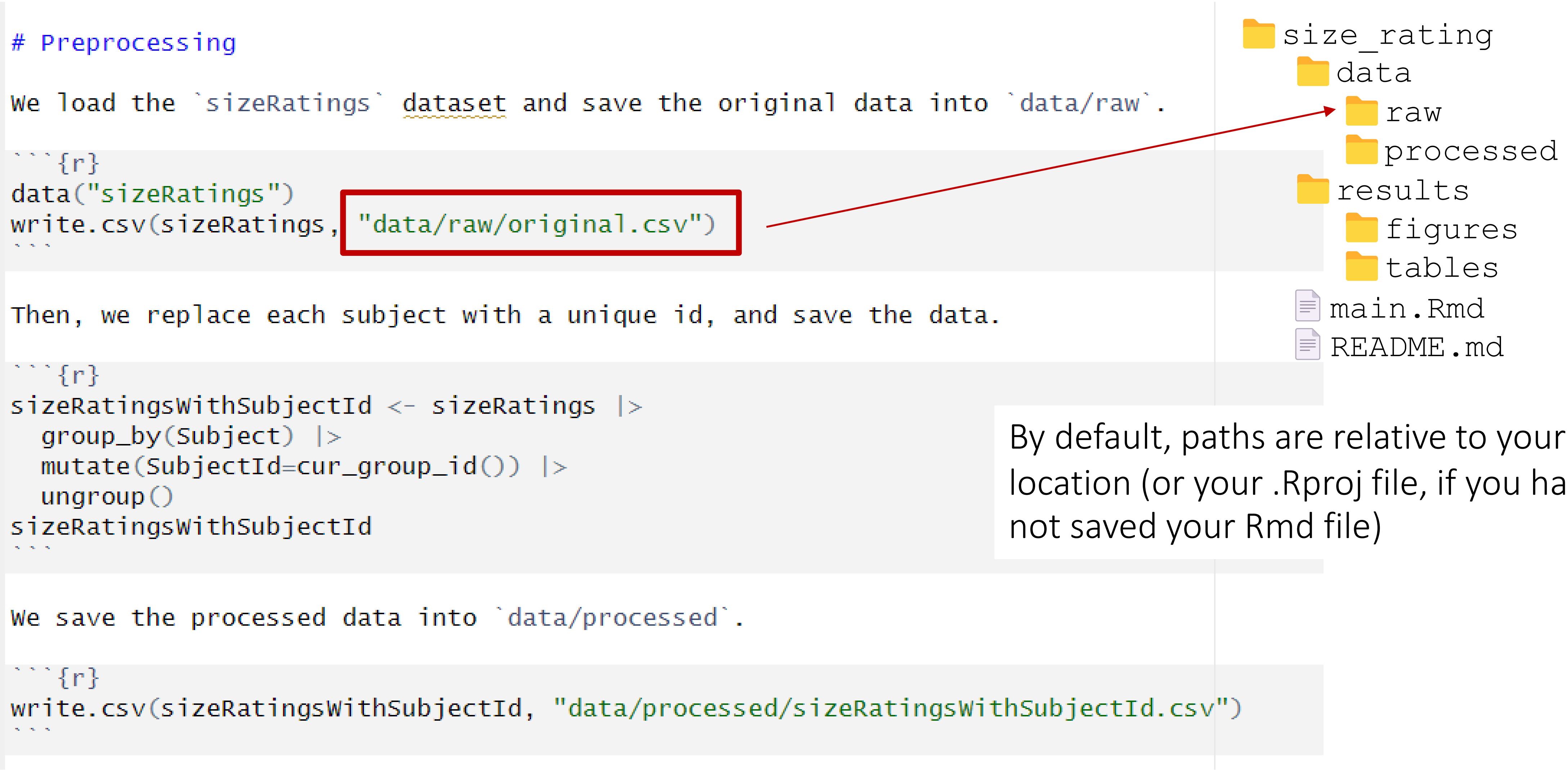
Write comments that tell the purpose of the packages

▶ on the top-right corner (Ctrl + Alt + C) runs the code and shows the results below



~~~
```{r}
library(languageR) # for the sizeRatings dataset
library(dplyr)     # for data preprocessing
library(ggplot2)   # for plotting
library(lme4)      # for model fitting
```
~~~

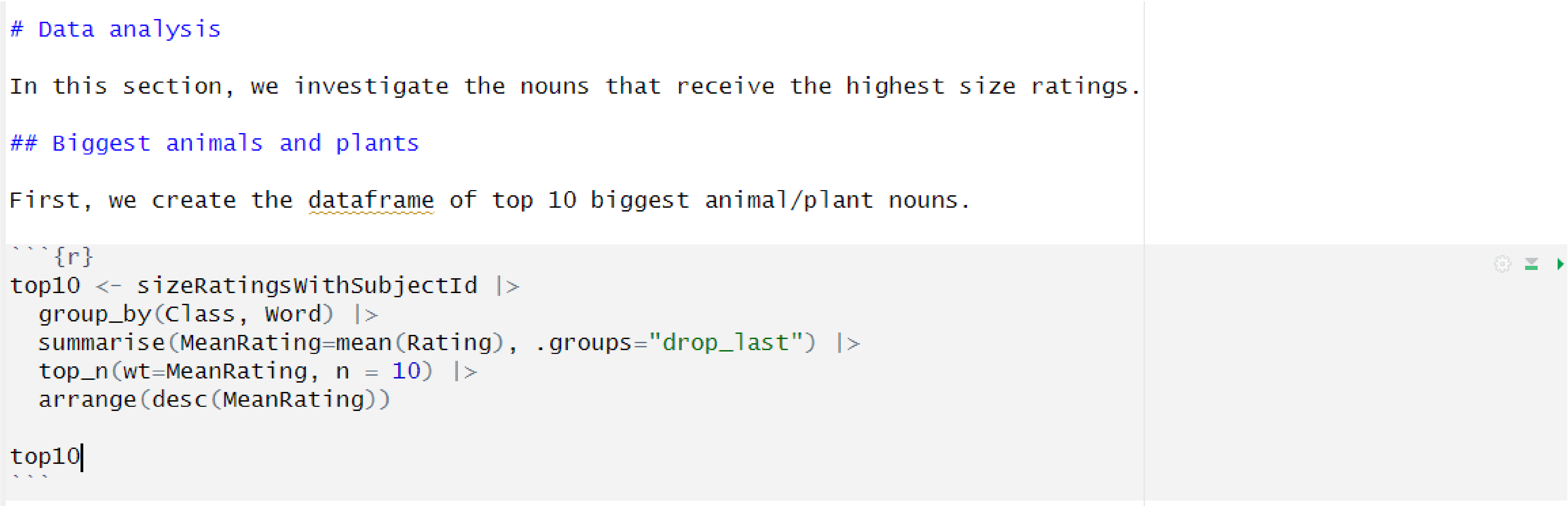
```
Attaching package: ‘dplyr’

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union

Loading required package: Matrix
```



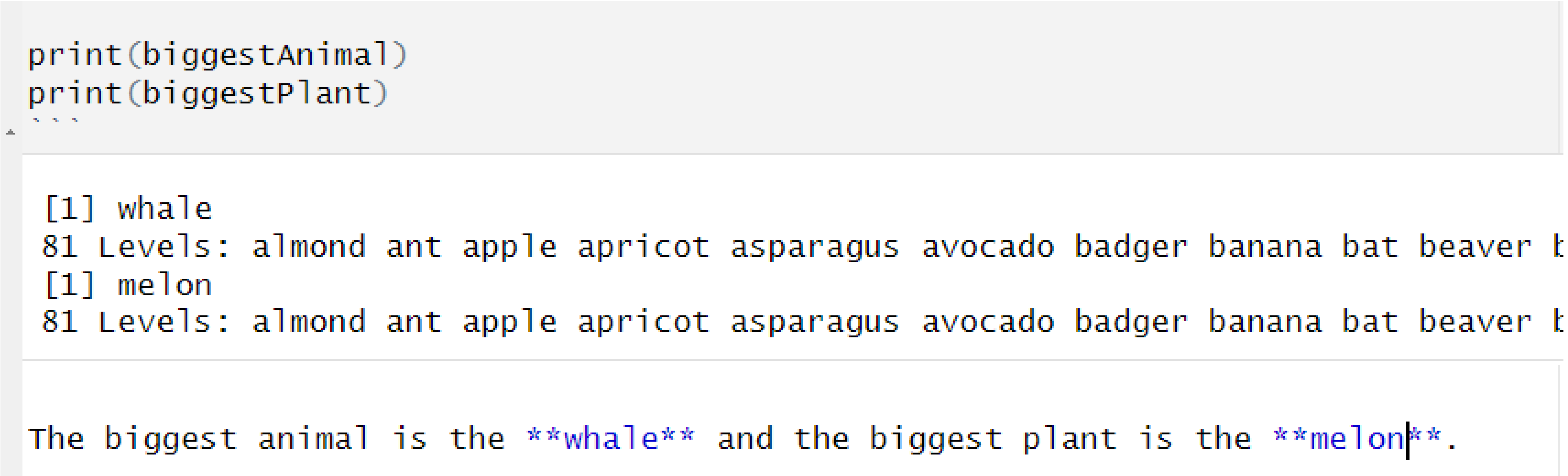
~~~
# Preprocessing

We load the `sizeRatings` dataset and save the original data into `data/raw`.

```{r}
data("sizeRatings")
write.csv(sizeRatings, "data/raw/original.csv")
```

Then, we replace each subject with a unique id, and save the data.

```{r}
sizeRatingsWithSubjectId <- sizeRatings |>
  group_by(Subject) |>
  mutate(SubjectId=cur_group_id()) |>
  ungroup()
sizeRatingsWithSubjectId
```

We save the processed data into `data/processed`.

```{r}
write.csv(sizeRatingsWithSubjectId, "data/processed/sizeRatingsWithSubjectId.csv")
```
~~~

```
size_rating
  data
    raw
    processed
  results
    figures
    tables
  main.Rmd
  README.md
```

By default, paths are relative to your Rmd location (or your .Rproj file, if you have not saved your Rmd file)



~~~
# Data analysis

In this section, we investigate the nouns that receive the highest size ratings.

## Biggest animals and plants

First, we create the dataframe of top 10 biggest animal/plant nouns.

```{r}
top10 <- sizeRatingsWithSubjectId |>
  group_by(Class, Word) |>
  summarise(MeanRating=mean(Rating), .groups="drop_last") |>
  top_n(wt=MeanRating, n = 10) |>
  arrange(desc(MeanRating))

top10
```
~~~



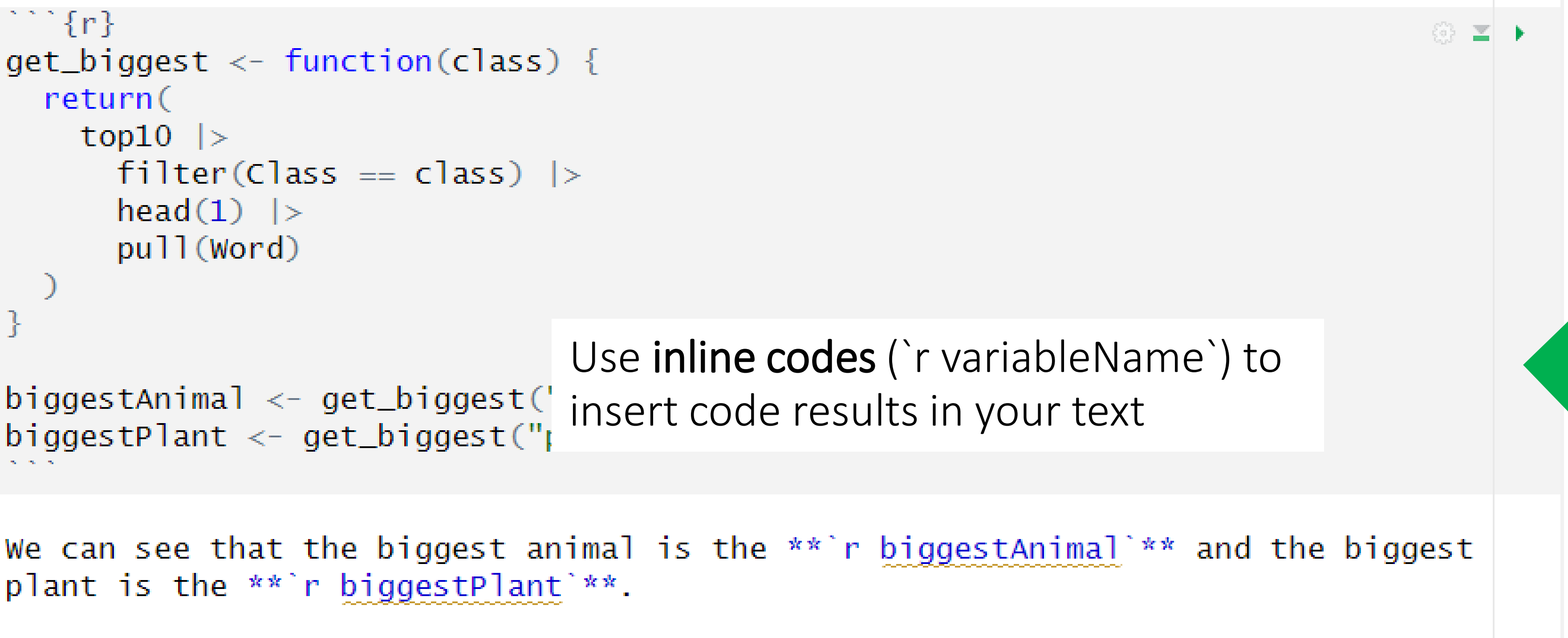
~~~
print(biggestAnimal)
print(biggestPlant)
```
~~~

```
 [1] whale
81 Levels: almond ant apple apricot asparagus avocado badger banana bat beaver b
 [1] melon
81 Levels: almond ant apple apricot asparagus avocado badger banana bat beaver b
```

```
The biggest animal is the **whale** and the biggest plant is the **melon**.
```

This is bad for reproducibility!

What if your data has errors and
when you fix it melons are not so big anymore!?



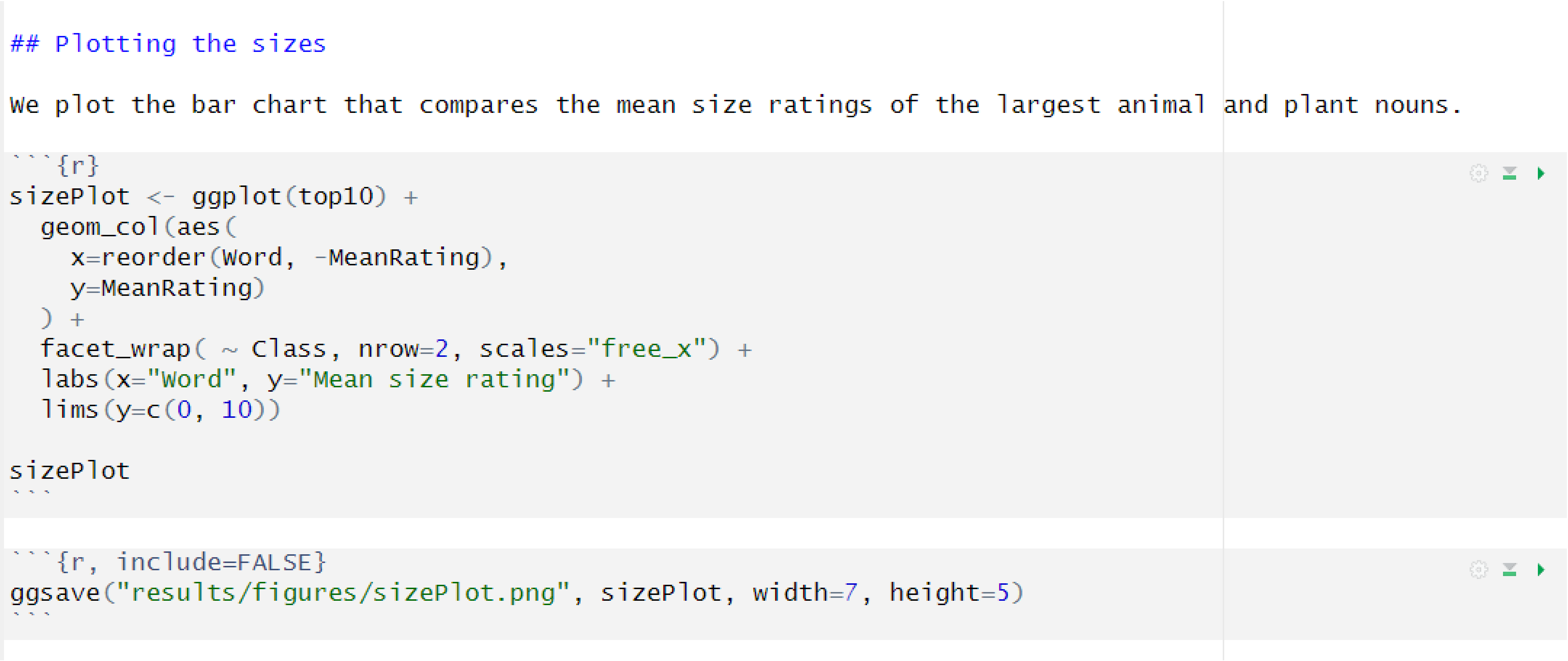
~~~
```{r}
get_biggest <- function(class) {
  return(
    top10 |>
      filter(Class == class) |>
      head(1) |>
      pull(Word)
  )
}

biggestAnimal <- get_biggest('
biggestPlant <- get_biggest("
```

We can see that the biggest animal is the **`r biggestAnimal`** and the biggest
plant is the **`r biggestPlant`**.
~~~

Use **inline codes** (`r variableName`) to insert code results in your text

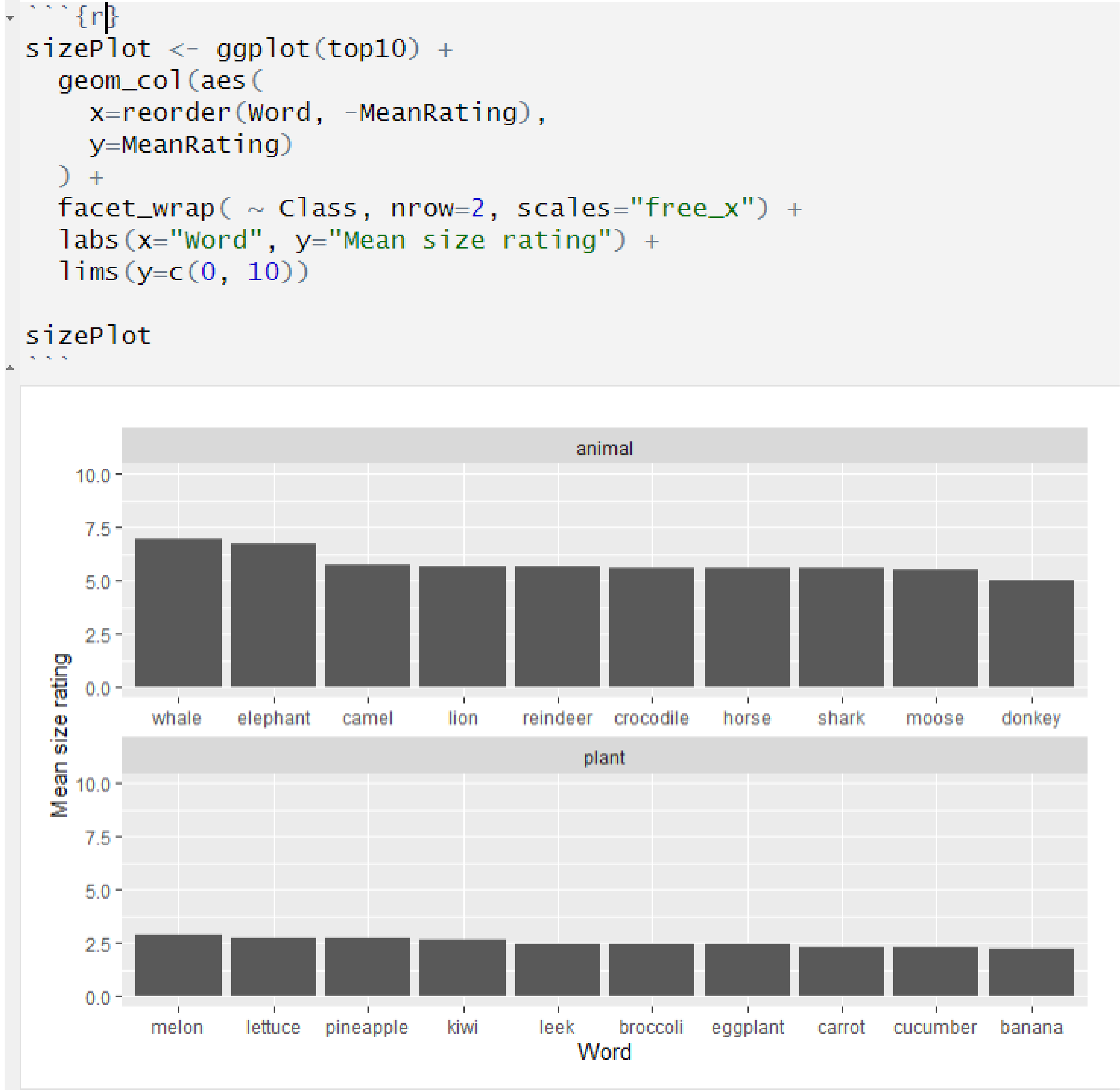
✓

These inline codes will turn into results when you turn the Rmd into a webpage or a PDF

In general, do this as much as possible

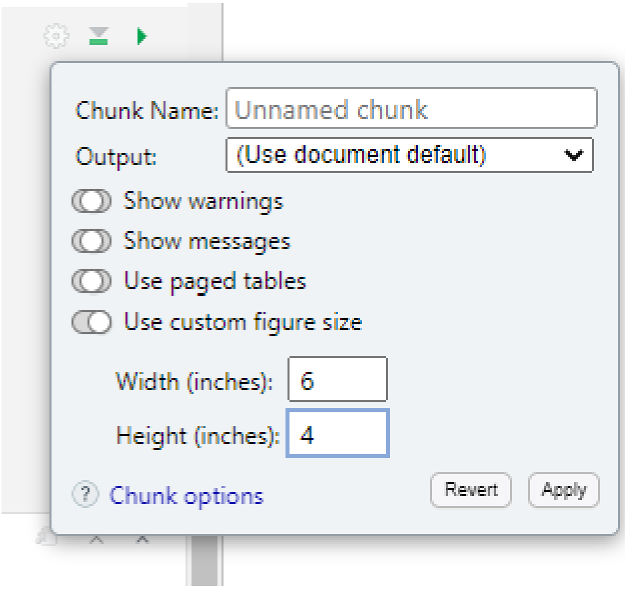


~~~
## Plotting the sizes

We plot the bar chart that compares the mean size ratings of the largest animal and plant nouns.

```{r}
sizePlot <- ggplot(top10) +
  geom_col(aes(
    x=reorder(Word, -MeanRating),
    y=MeanRating)
  ) +
  facet_wrap( ~ Class, nrow=2, scales="free_x") +
  labs(x="Word", y="Mean size rating") +
  lims(y=c(0, 10))

sizePlot
```

```{r, include=FALSE}
ggsave("results/figures/sizePlot.png", sizePlot, width=7, height=5)
```
~~~



~~~
```{r}
sizePlot <- ggplot(top10) +
  geom_col(aes(
    x=reorder(Word, -MeanRating),
    y=MeanRating)
  ) +
  facet_wrap( ~ Class, nrow=2, scales="free_x") +
  labs(x="Word", y="Mean size rating") +
  lims(y=c(0, 10))

sizePlot
```
~~~

Use chunk options to change figure dimensions etc. (Rerun to see changes)



Chunk Name: Unnamed chunk

Output: (Use document default)

○ Show warnings

○ Show messages

○ Use paged tables

○ Use custom figure size

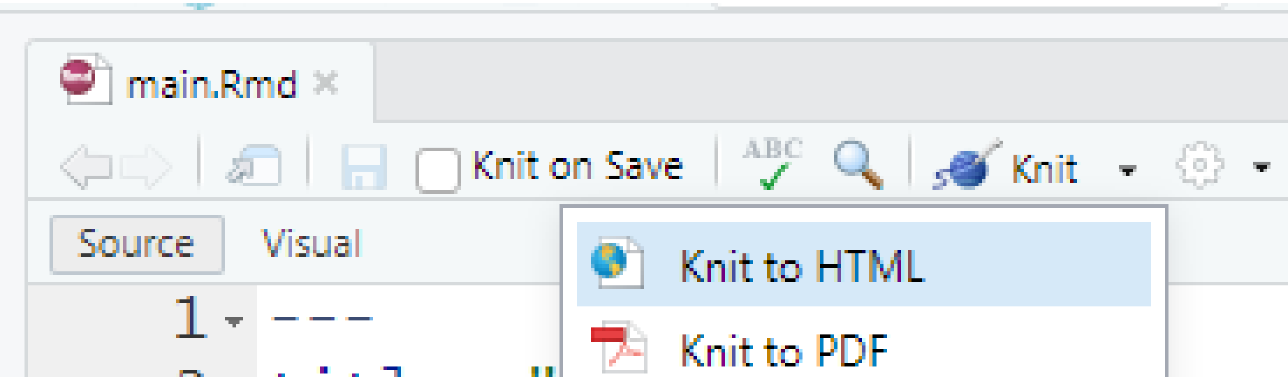
Width (inches): 6

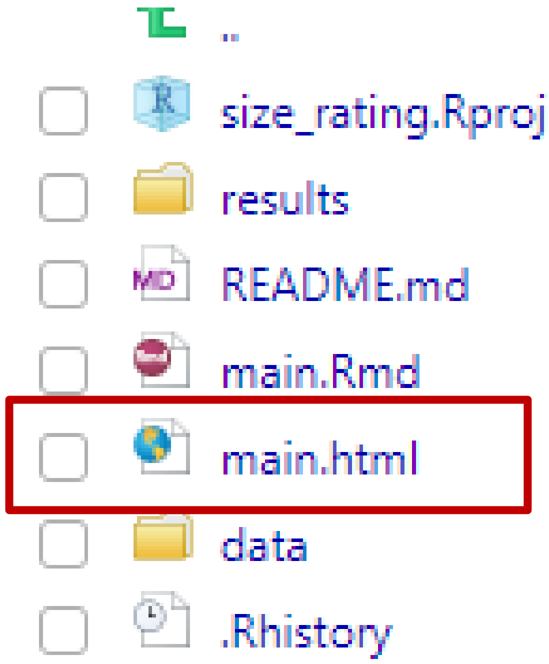
Height (inches): 4

“Knitting”

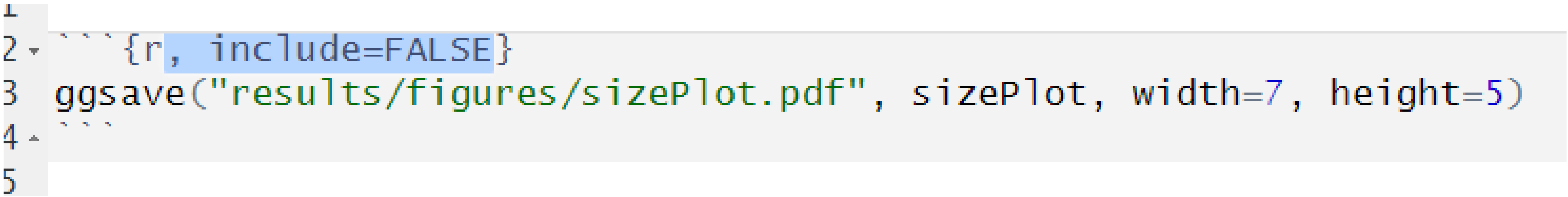
You can “knit” RMarkdown documents into formats such as webpages and PDFs.

For this demo, we will make an HTML webpage.



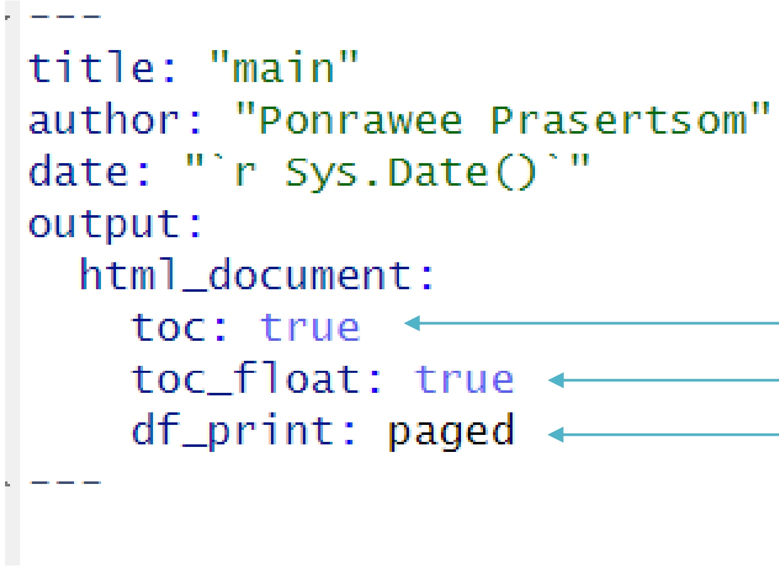


Use chunk option `include=FALSE` to hide the chunk from the knitted page



~~~
```{r, include=FALSE}
ggsave("results/figures/sizePlot.pdf", sizePlot, width=7, height=5)
```
~~~

Nice options for HTML



```
---
title: "main"
author: "Ponrawee Prasertsom"
date: "`r Sys.Date()`"
output:
  html_document:
    toc: true
    toc_float: true
    df_print: paged
---
```

Make Table of Content

Make Table of Content floating

Display tables as paged tables

Now what?

Sharing!

1. Zip the project and upload as supplemental data to your paper
2. Sharing on GitHub: Our project structure plays nicely with GitHub, with README and everything. (You may want to use `.gitignore` to ignore sensitive files)

Attend our GitHub workshop or read more at https://pplsopenresearch.github.io to learn more about GitHub

Recap

1. Reproducible workflow ensures that your research results can be reproduced
2. Folder structure should reflect your work steps and separate them clearly
3. RMarkdown can help, because it encourages documentation of the steps and helps the reproducer understand them

What's next?

You can learn more!

- Read more about reproducibility: http://www.practicereproducibleresearch.org/

- RMarkdown: Very powerful; Can "knit" your Rmd into journal articles! (Use the `rticle` package)

What's next?

Guaranteeing reproducibility is difficult.

Some more issues:

1. Software version: What if a function you use is no longer available in a different version of an R package?
2. Software availability: What if one day `ggplot` is taken off CRAN?
3. Computing environment differences: What if a different operating system gives a different result?

What’s next?

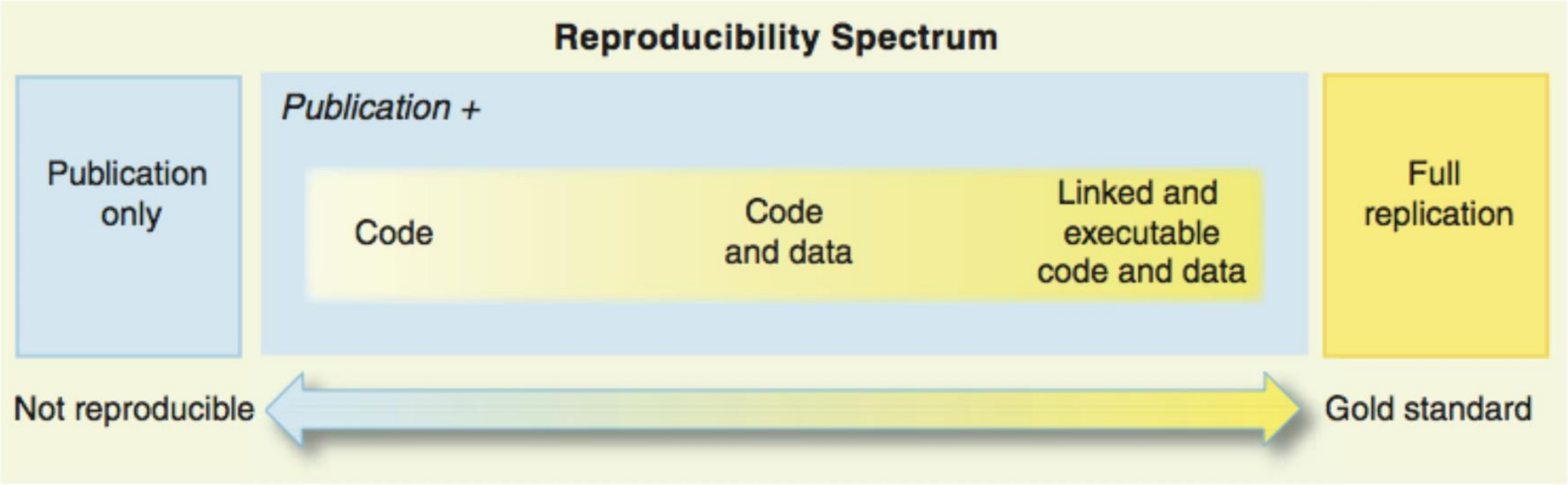
`renv` R package tracks packages you use in the project and saves version information etc. Reproducers can load & install them via `renv`.

Docker packages your whole computing environment, from the OS, to RStudio, R, R packages and your project.

Peng (2011)

What's next?

We want to try to move toward the gold standard



You can't do everything at once but you can adopt these practices gradually!