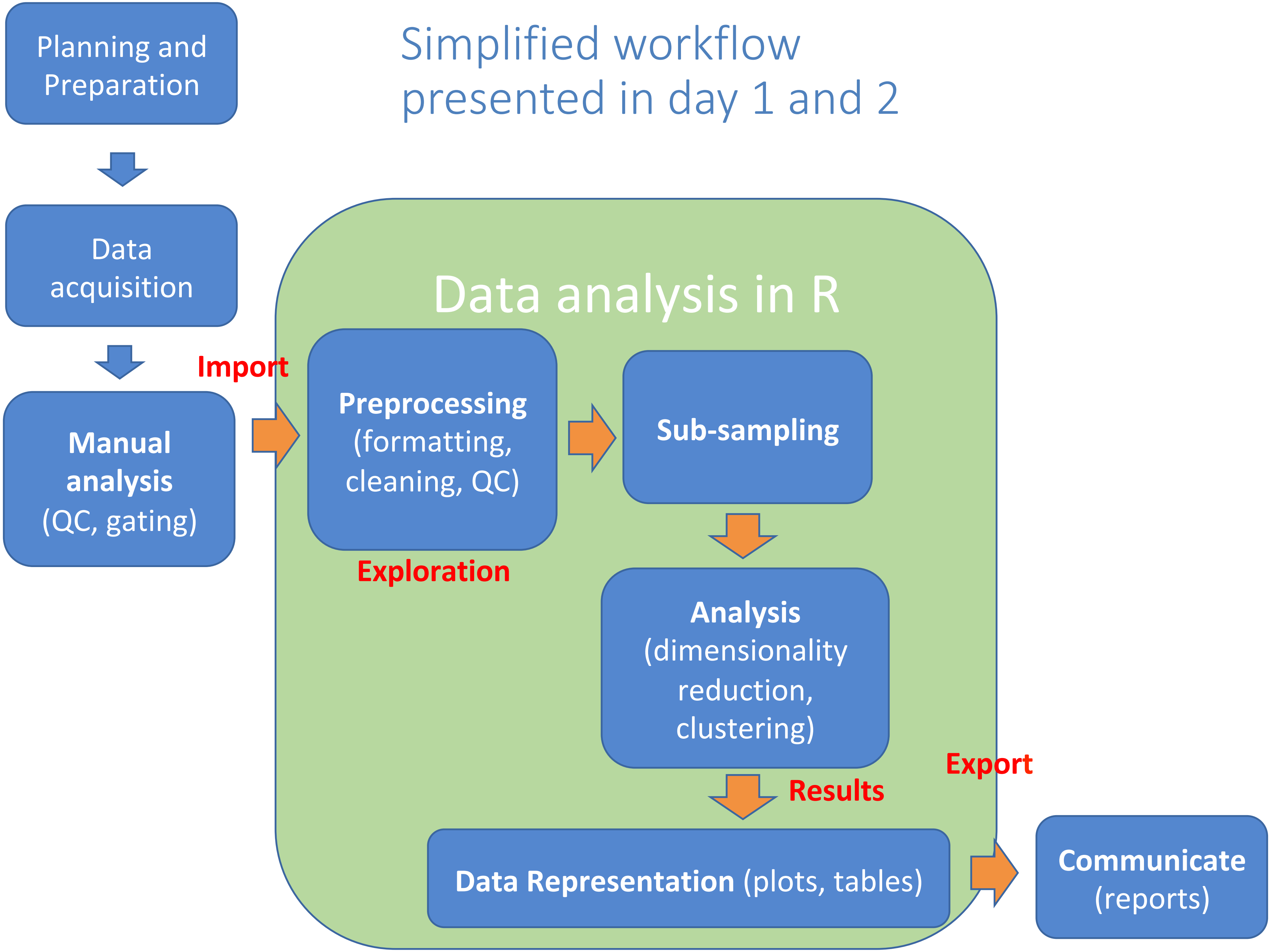
04 Generate HTML or PDF reports

Simplified workflow presented in day 1 and 2



Planning and Preparation

Data acquisition

Manual analysis (QC, gating)

Import

Data analysis in R

Preprocessing (formatting, cleaning, QC)

Exploration

Sub-sampling

Analysis (dimensionality reduction, clustering)

Results

Data Representation (plots, tables)

Export

Communicate (reports)

R Markdown

If a data analysis project involves many steps and various plots, one easy and practical way to bundle and organize all steps of analysis together is to use R markdown files to generate PDF or html reports.

Document format that **combines R (or other) code with Markdown**, a lightweight markup language.

Integrate text, code, and the output of code (such as tables and figures) in a single document.

https://rmarkdown.rstudio.com/gallery.html ← Examples of types of documents that can be created

Create:

- Documents: HTML, PDF, Word
- Interactive documents: html, shiny apps
- Presentations
- eBooks
- Websites
- Package vignettes

R Markdown

Key components:

- **Markdown**: format text using plain text characters. Use Markdown to create headers, lists, links, and other formatting elements.
- **R Code Chunks**: include executable R code chunks. These chunks are enclosed by three backticks (```) and specified with the `{r}` tag. When generating (i.e. "knitting") the document, these code chunks are executed, and the results are inserted into the final document. Code chunks can be hidden or shown in the final document
- **Output Formats**: supports various formats, including HTML and PDF. The desired output format is specified in the document's metadata. When you "knit" the document, R Markdown generates the final document in the specified format.
- **YAML Metadata**: The metadata at the beginning of the R Markdown document is typically written in YAML format. This metadata includes information such as the document title, author, date, and output format.

RStudio facilitates the creation of R Markdown documents thanks to the rmarkdown, knitr and pander packages. The "scripts" now have .Rmd extension.

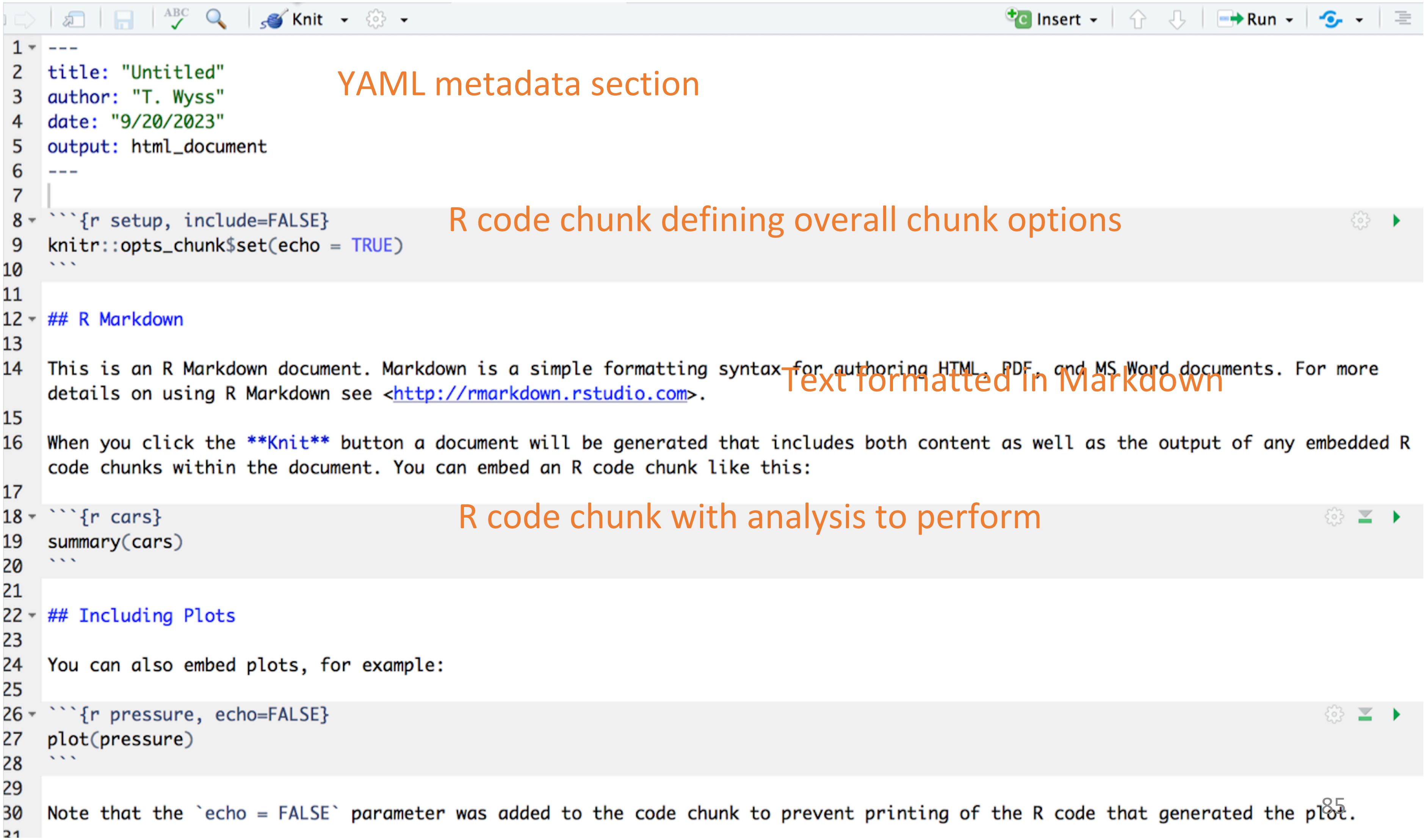
https://cran.r-project.org/web/packages/rmarkdown/index.html

https://cran.r-project.org/web/packages/knitr/index.html

https://cran.r-project.org/web/packages/pander/index.html

R Markdown

Create a .Rmd template within RStudio:
File > New File > R Markdown :



```
---
title: "Untitled"
author: "T. Wyss"
date: "9/20/2023"
output: html_document
---

```

YAML metadata section

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

R code chunk defining overall chunk options

```

## 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:

```

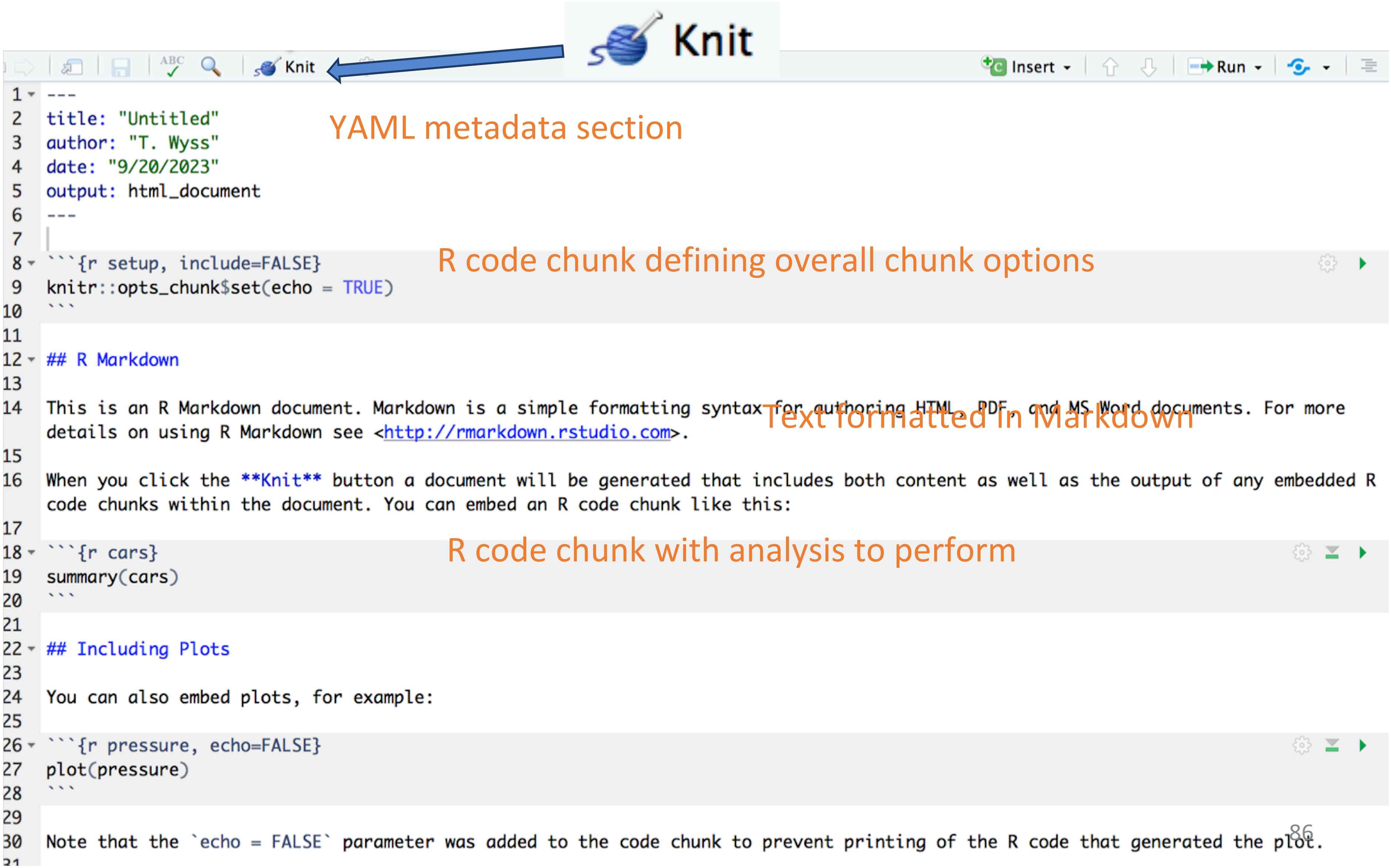
Text formatted in Markdown

~~~
```{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.
~~~

R code chunk with analysis to perform

R Markdown

Knit: generate the final report



~~~
---
title: "Untitled"
author: "T. Wyss"
date: "9/20/2023"
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

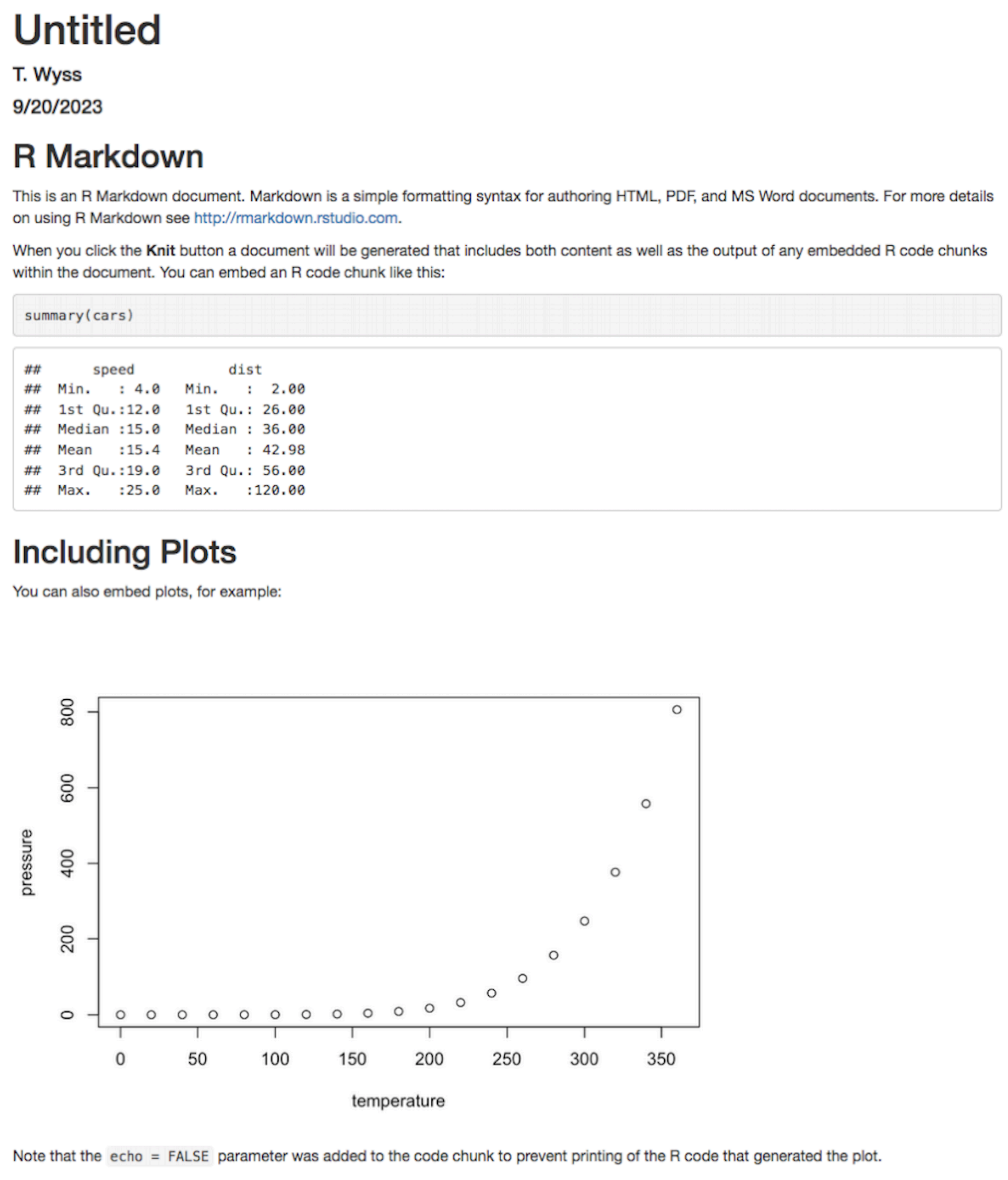
You can also embed plots, for example:

```{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 metadata section

R code chunk defining overall chunk options

Text formatted in Markdown

R code chunk with analysis to perform
```

# R Markdown

Generate a new file with .html extension saved by default in the same folder as where the .Rmd file is located.



# Untitled

T. Wyss

9/20/2023

## 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:

```
summary(cars)
```

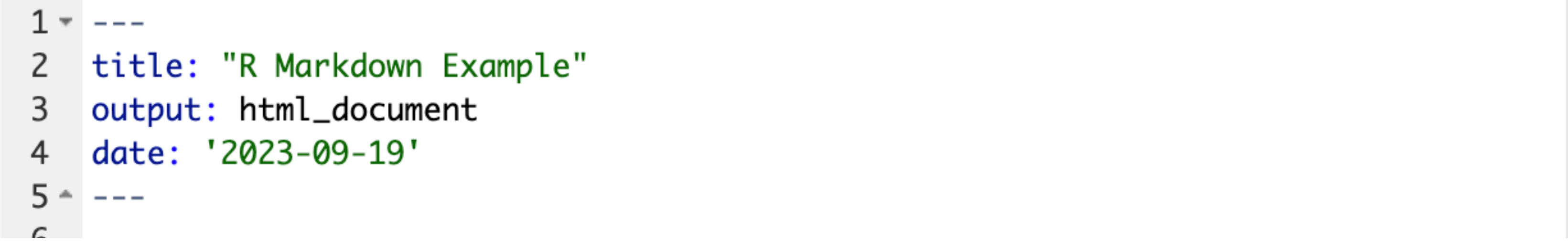
```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00  
```

Including Plots

You can also embed plots, for example:

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

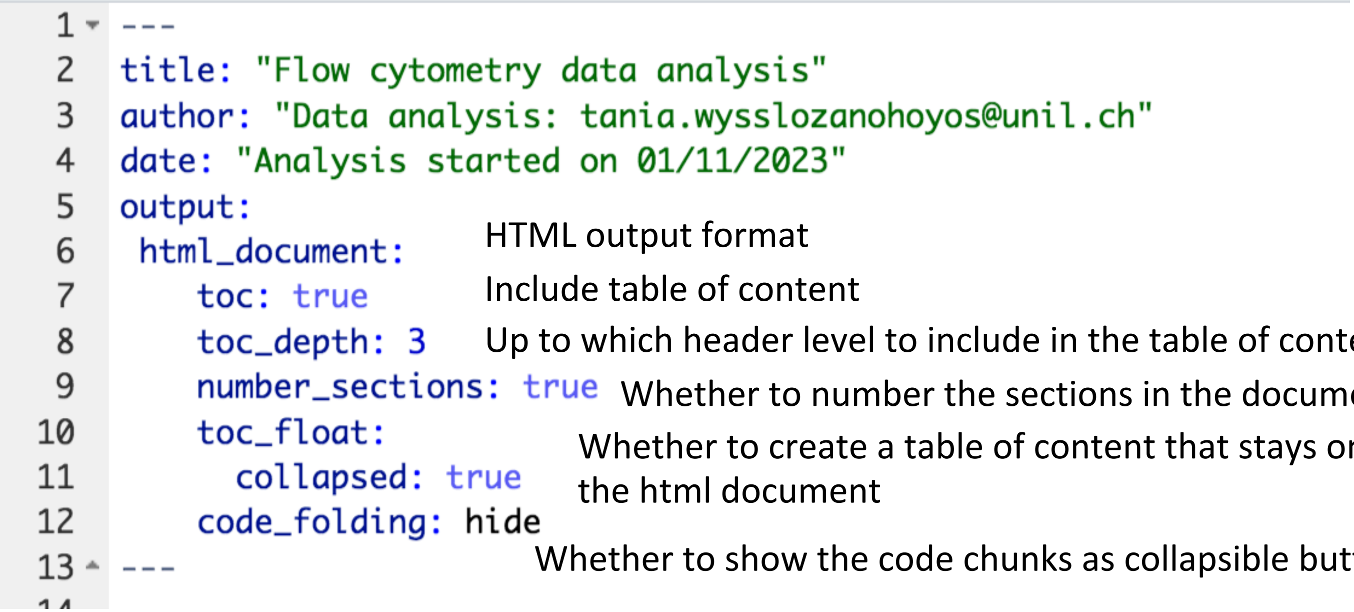
YAML metadata section



```
---
title: "R Markdown Example"
output: html_document
date: '2023-09-19'
---
```

- This section must begin with a line containing just three dashes **---** and must close with a line containing just three dashes **---**
- Parameters come in **key:value pairs**
- The key:value pairs are separated by **colons (not equals signs!**)
- The order of the parameters (if not indented) does not matter
- Basic parameters are pre-selected, but we can change them (for example, if we wish a ”**pdf_document**” instead of “**html_document**”)

Customize the YAML metadata section



```
---
title: "Flow cytometry data analysis"
author: "Data analysis: tania.wysslozanohoyos@unil.ch"
date: "Analysis started on 01/11/2023"
output:
 html_document:
    toc: true
    toc_depth: 3
    number_sections: true
    toc_float:
      collapsed: true
    code_folding: hide
---
```

HTML output format

Include table of content

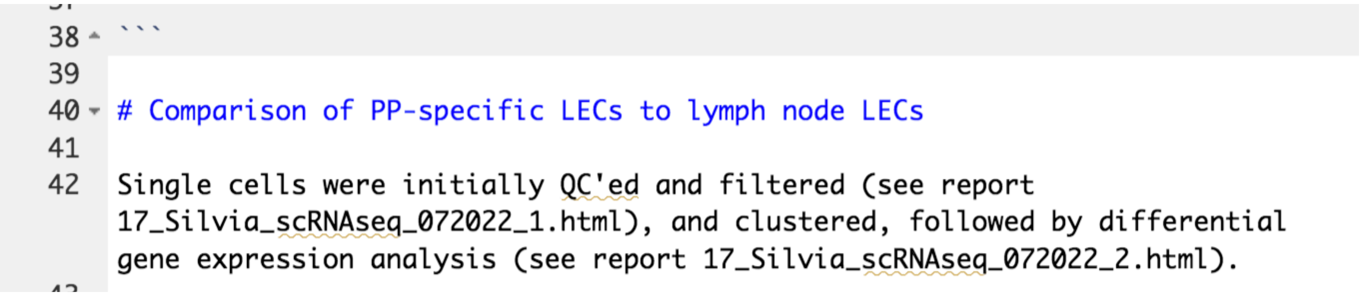
Up to which header level to include in the table of content

Whether to number the sections in the document according to header level

Whether to create a table of content that stays on the side when scrolling down the html document

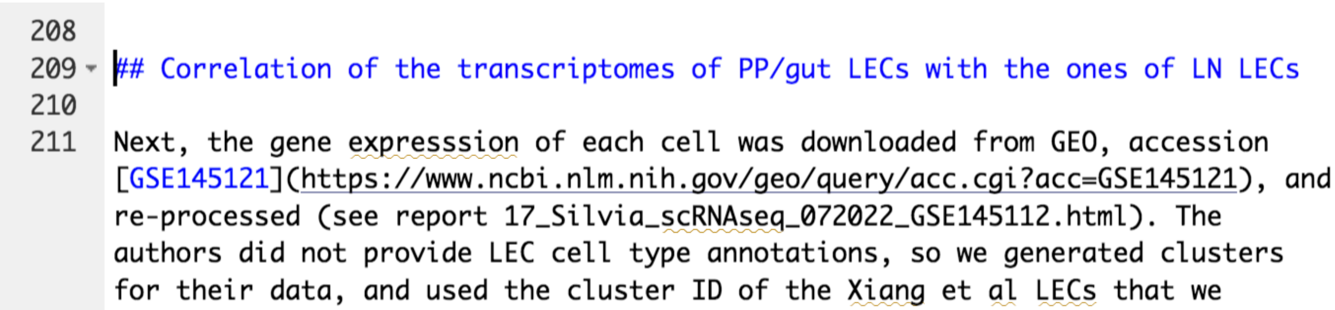
Whether to show the code chunks as collapsible buttons

…



```
```

Comparison of PP-specific LECs to lymph node LECs

Single cells were initially QC'ed and filtered (see report
17_Silvia_scRNAseq_072022_1.html), and clustered, followed by differential
gene expression analysis (see report 17_Silvia_scRNAseq_072022_2.html).
```

…



```

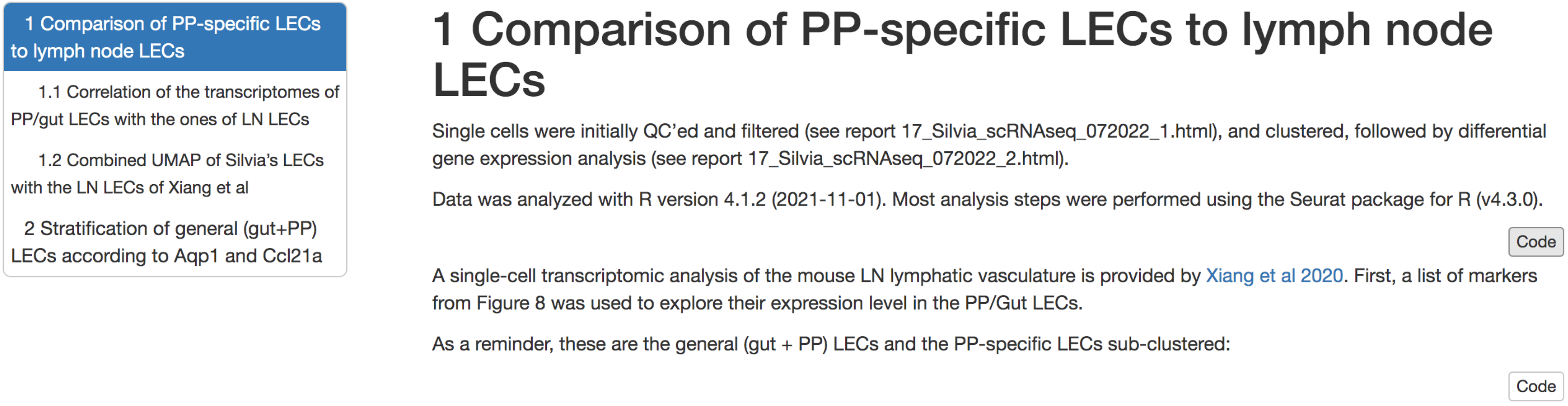
## Correlation of the transcriptomes of PP/gut LECs with the ones of LN LECs

Next, the gene expresssion of each cell was downloaded from GEO, accession
[GSE145121](https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE145121), and
re-processed (see report 17_Silvia_scRNAseq_072022_GSE145112.html). The
authors did not provide LEC cell type annotations, so we generated clusters
for their data, and used the cluster ID of the Xiang et al LECs that we
```
```

# Customize the YAML metadata section:

Floating Table Of Content (TOC) with numbered sections

Code folding buttons



## 1 Comparison of PP-specific LECs to lymph node LECs

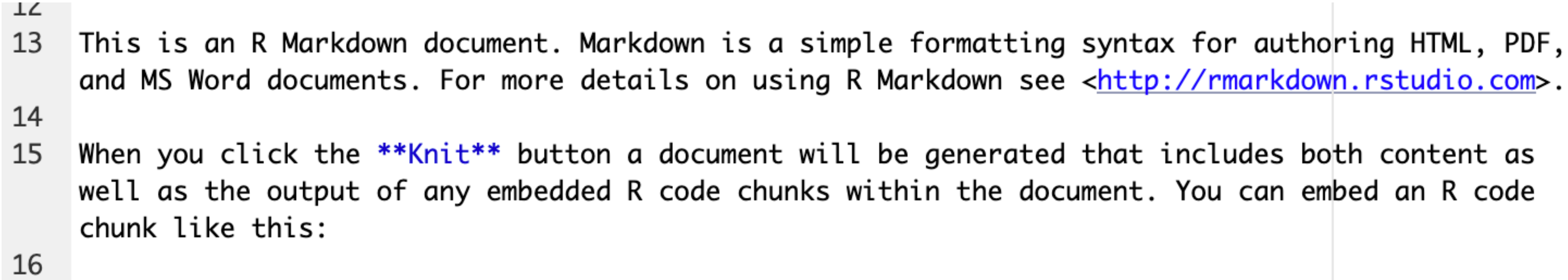
Single cells were initially QC'ed and filtered (see report 17_Silvia_scRNAseq_072022_1.html), and clustered, followed by differential gene expression analysis (see report 17_Silvia_scRNAseq_072022_2.html).

Data was analyzed with R version 4.1.2 (2021-11-01). Most analysis steps were performed using the Seurat package for R (v4.3.0).

A single-cell transcriptomic analysis of the mouse LN lymphatic vasculature is provided by Xiang et al 2020. First, a list of markers from Figure 8 was used to explore their expression level in the PP/Gut LECs.

As a reminder, these are the general (gut + PP) LECs and the PP-specific LECs sub-clustered:

# Text

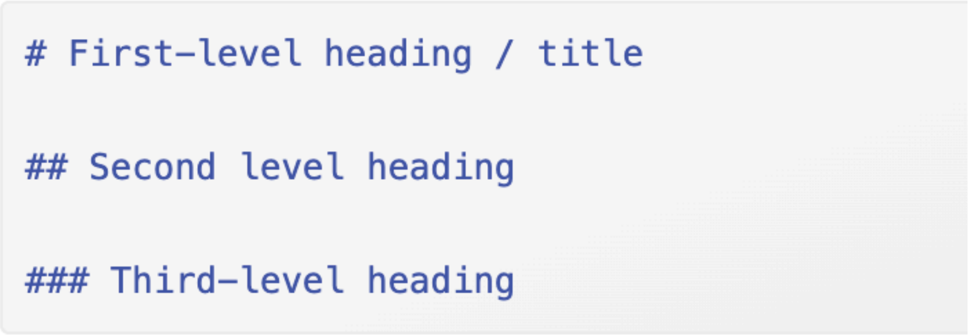


```
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF,
and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

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:

```

- Free text (like in a word document)
- **New lines**: enter two spaces*at the end of the previous line and then hit return.
- **Cases**:
  - Underscores (_text_) or single asterisk (*text*) to *italicise*
  - Double asterisks (**text**) for **bold text**
  - Back-ticks (`code`) to display text as code

# Titles and headers

Different header levels are established with different numbers of hash symbols at the start of a new line.



```
First-level heading / title

Second level heading

Third-level heading
```

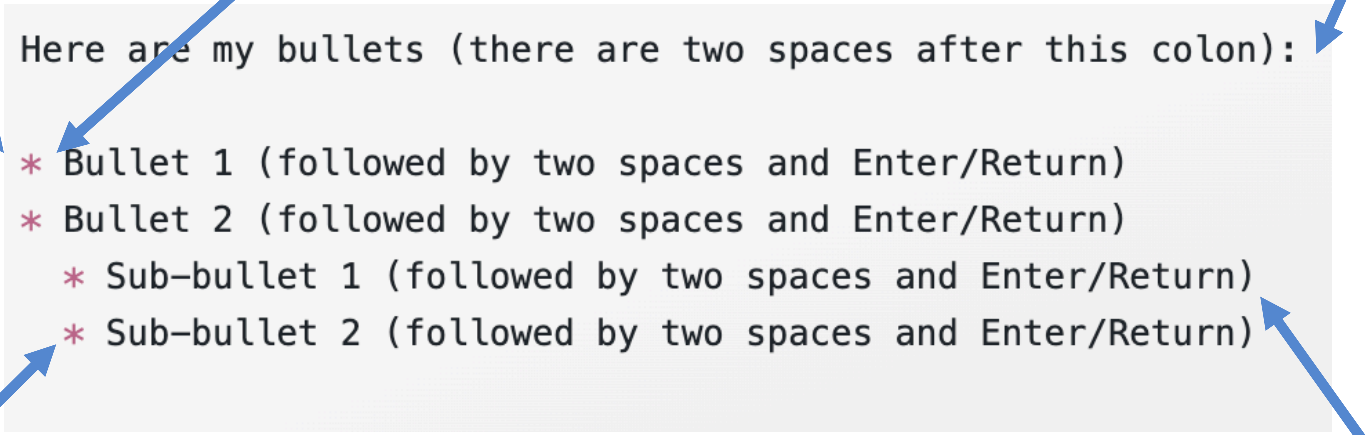
This is different than in a code chunk, in which hashes are used for commenting !

# Bullets and numbering

Use asterisks (*) to create a bullet list.

Include a space between the asterisk and your bullet text

Finish the previous sentence, enter two spaces, hit return twice, and then start the bullets



```
Here are my bullets (there are two spaces after this colon):

* Bullet 1 (followed by two spaces and Enter/Return)
* Bullet 2 (followed by two spaces and Enter/Return)
  * Sub-bullet 1 (followed by two spaces and Enter/Return)
  * Sub-bullet 2 (followed by two spaces and Enter/Return)
```

Sub-bullets work the same way but are indented (tab)

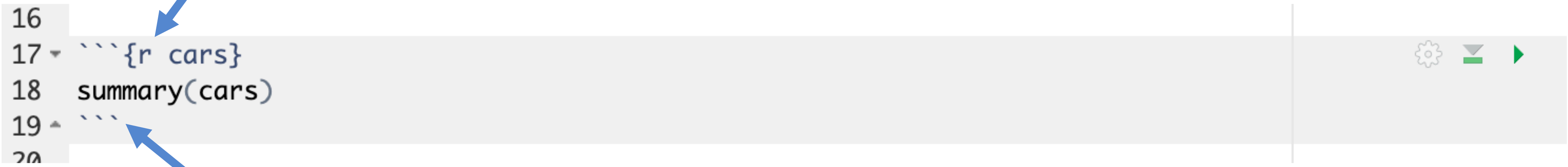
After each bullet enter two spaces and then hit return

Numbers work the same way but instead of an asterisk, write 1), 2), etc.

Code chunks

- This is where you load packages, import data, and perform the actual data processing, analysis and visualization.
- There may be many code chunks

Each chunk is opened with a line that starts with three back-ticks, and curly brackets that contain parameters for the chunk ({ }).
Here, there is a name for the chunk which is `cars`

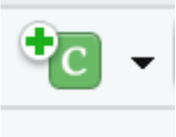


~~~
```{r cars}
summary(cars)
```
~~~

The chunk ends with three more back-ticks

Create a new chunk:

- type the starting and ending lines yourself, or
- use the keyboard shortcut “Ctrl + Alt + i” (or Cmd + Shift + r in Mac), or
- click the green ‘insert a new code chunk’ icon



Code chunk options

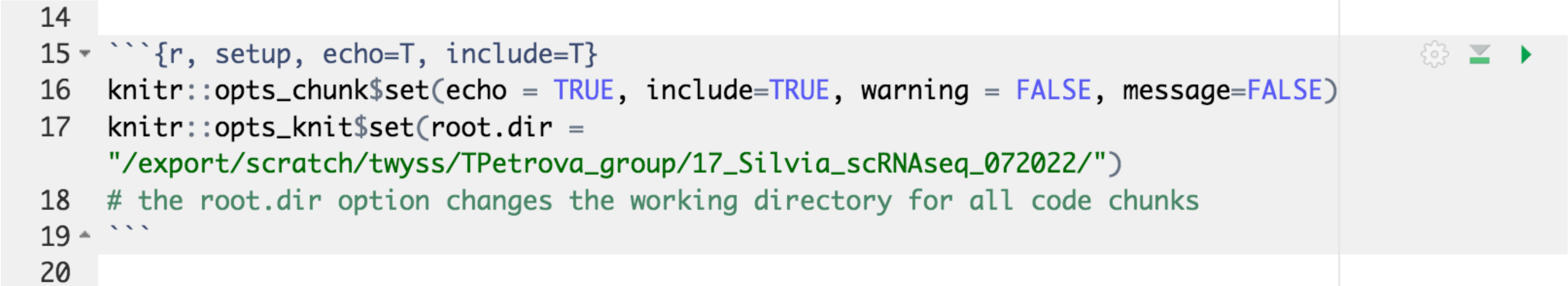
- A chunk header must be written in *one line*
- Always start with ‘r’
- After the ‘r’ you can **optionally** write a chunk “name” (**which has to be unique across the whole Rmd file**).
- Other options are written as tag=value, such as:
  - eval = FALSE to not run the R code
  - echo = FALSE to not print the chunk’s R source code
  - warning = FALSE to not print warnings produced by the R code
  - message = FALSE to not print any messages produced by the R code
  - include = either TRUE/FALSE whether to include chunk outputs (e.g. plots) or not
  - fig.align = "center" adjust how a figure is aligned across the page

More about chunk options in https://yihui.org/knitr/options/

Global options

- Applied to all chunks present in the Rmd file
- Set up within the very first R code chunk using the opts_chunk
- For instance, below, the global options set with `opts_chunk$set()` are:
    - echo=TRUE : print the R source code
    - include=TRUE : include
    - warning=FALSE: do not print warning messages
    - message=FALSE : do not print any message (eg verbose output of some functions)
- `opts_knit$set()` can be used to set the folder where to save plots or look for data.

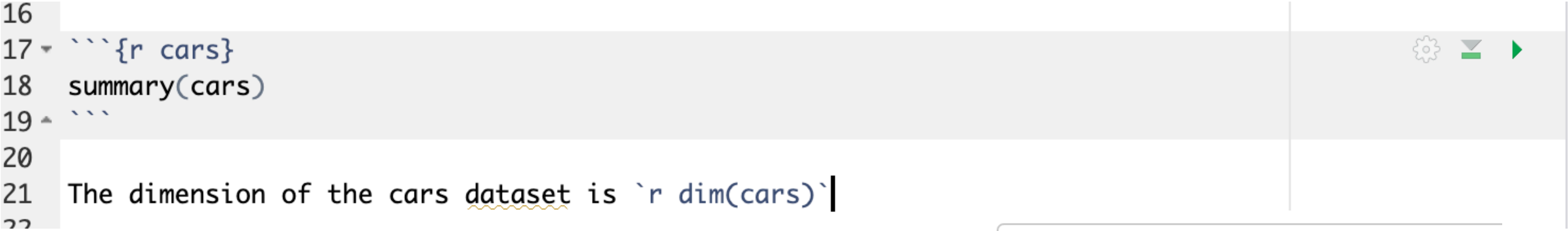
  It changes the working director for all code chunks



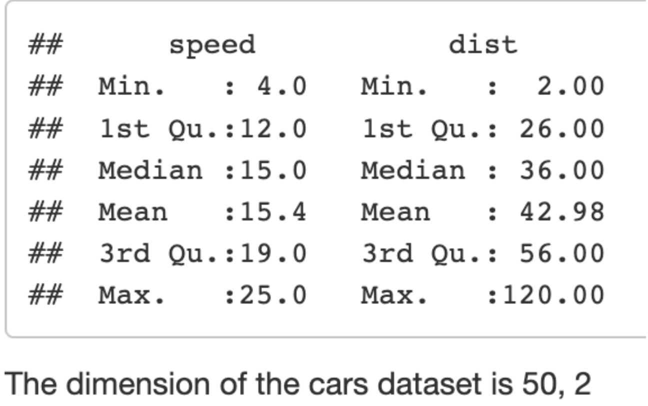
~~~
```{r, setup, echo=T, include=T}
knitr::opts_chunk$set(echo = TRUE, include=TRUE, warning = FALSE, message=FALSE)
knitr::opts_knit$set(root.dir =
"/export/scratch/twyss/TPetrova_group/17_Silvia_scRNAseq_072022/")
# the root.dir option changes the working directory for all code chunks
```
~~~

In-text R code

- Short R code commands within back-ticks
- Within the back-ticks, begin the code with “r” and a space, so RStudio knows to evaluate the text within the back-ticks as R code
- Example:



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

The dimension of the cars dataset is `r dim(cars)`
~~~

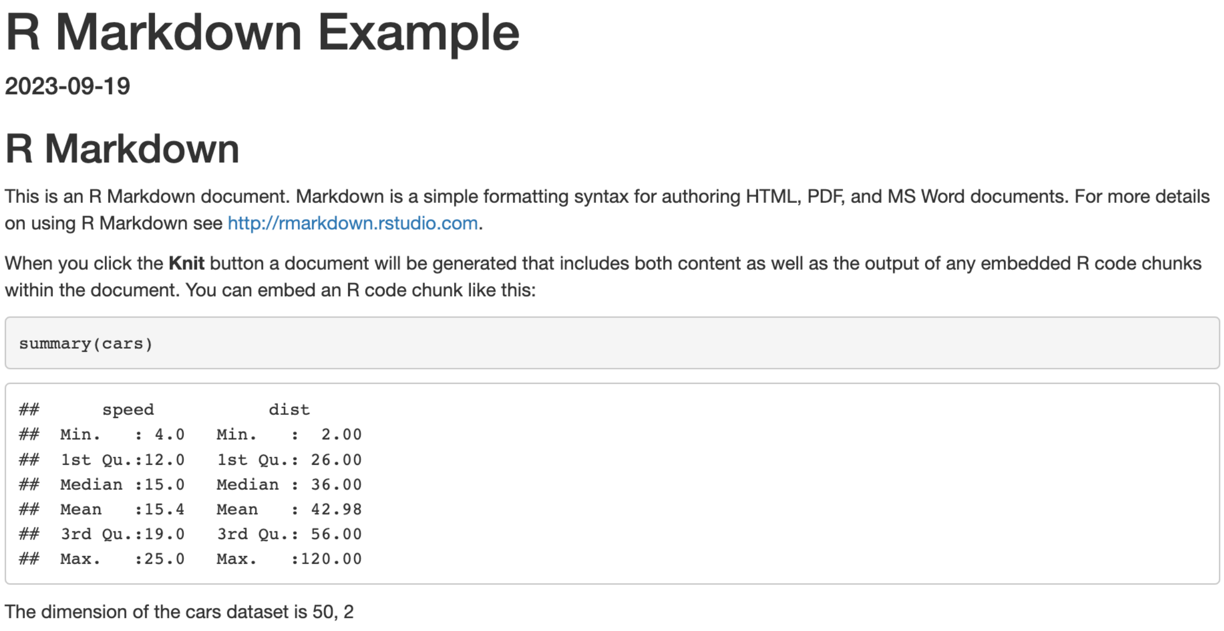


```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

The dimension of the cars dataset is 50, 2

Producing the document

- **Knitting** a file is how we generate the output file from the R markdown file
- RStudio will show you the progress within an ‘R Markdown’ tab near your R console
- The document will automatically open when complete
- **The document will be saved in the same folder as your R markdown script, and with the same file name**



R Markdown Example

2023-09-19

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:

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00  
```

The dimension of the cars dataset is 50, 2

Let’s practice – 11 : Rmarkdown

In this exercise we will create an .Rmd file, and modify it to include the code for analysis and results of dimensional reduction of the **FR_FCM_Z3WR** data, using the code of **Exercise 4**.

Create a new .Rmd in which you will:

1) Modify the YAML metadata section to include a table of content with numbered sections of 2 header levels, and buttons to hide the code
2) Write a paragraph with a level-1 header, that describes the content of the document
3) Create a chunk with global options that will print the R source code, include chunk outputs, and hide warnings and messages
4) Create a chunk that will load the required libraries: flowCore, CATALYST
5) Create a code chunk for each of the following steps, that will run and/or print the output of the following analysis:
    1) Import the cleaned flowSet generated after cleaning with flowAI during Exercise 3, as well as the panel from the csv file. **Important**: modify the path that becomes relative to the location of the .Rmd script. Eg: load("**../**course_datasets/FR_FCM_Z3WR/fcs_clean.RData"). Create a SingleCellExperiment object.

Let’s practice – 11 : Rmarkdown - continued

2) Calculate the UMAP with default parameters, as in Exercise 4.

3) Plot the UMAP, coloring the cells according to CD3 expression and faceting according to time point. Set the chunk option to center the figure, using fig.align = "center" within the chunks option.

4) Create additional chunks by playing with the plot: coloring according to other marker genes, etc. You can also try to arrange plots in grids with cowplot. Have fun!

6) Knit the document to an html report and admire it in your web-browser!

Please write wish-list for day 5 in the google doc

Thank you for your attention!

https://agora-cancer.ch/scientific-platforms/translational-data-science-facility/

Any questions? Contact us !

tds-facility@sib.swiss