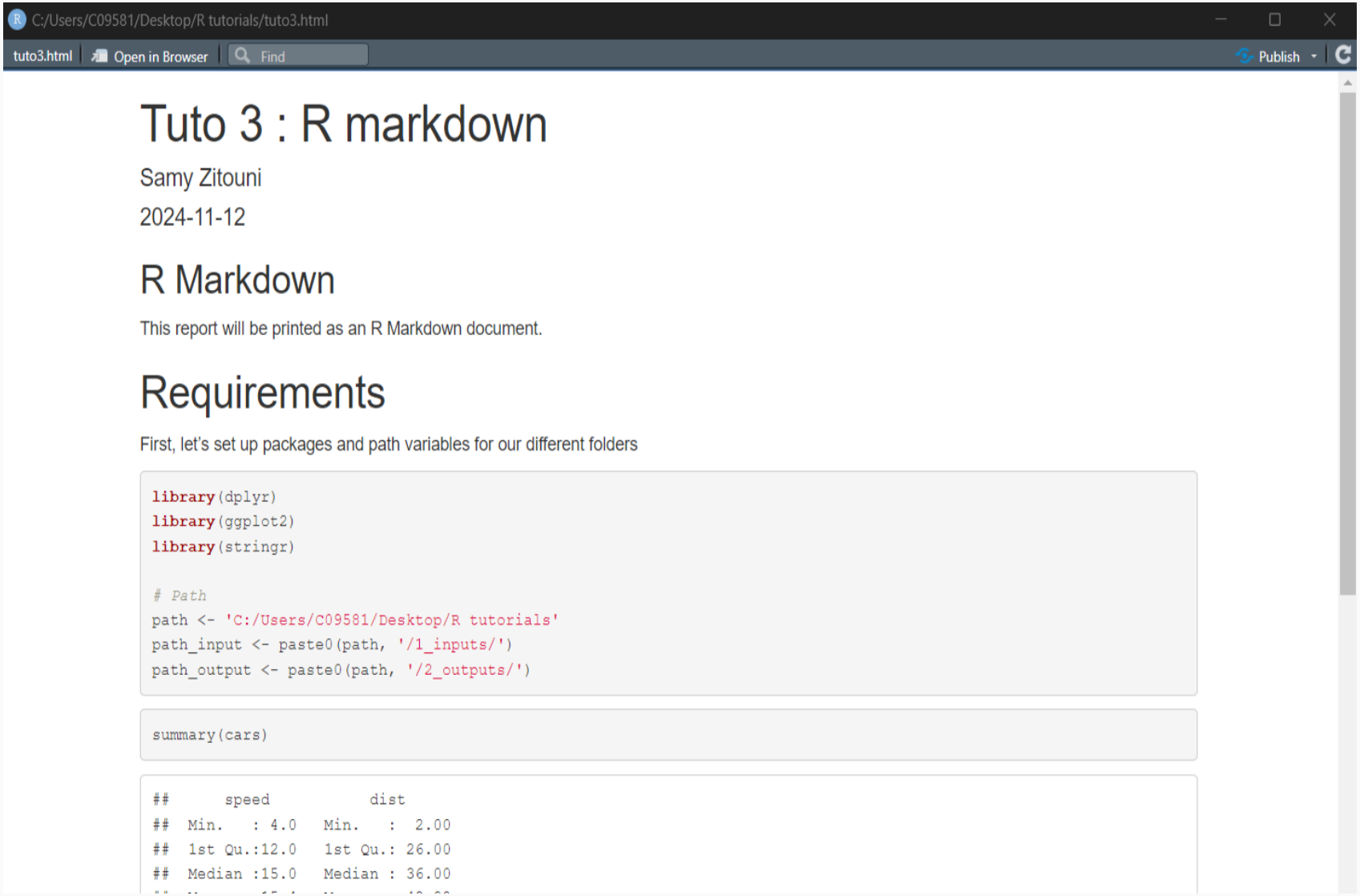
An introduction to R

R markdown basics

Samy Zitouni

October 2024

What is R Markdown ?



What is R markdown

Generates reports, in which you can include...

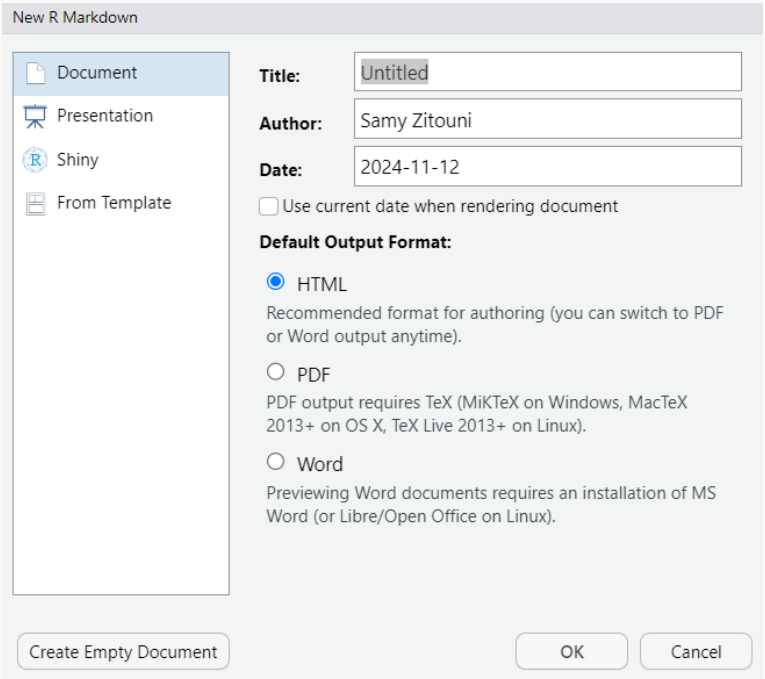
- Titles, text, global text formatting
- **Code** parts
- The results of your code, whether they are **datasets**, **plots**, regression **summaries**, etc.
- Math equations with LaTeX
- It is better that commenting code, and **more practical** than exporting your results on another text editor

You can generate...

- Reports
- Presentations
- Books, widget, dashboards, etc.
- See `here` for more examples

Getting started

- Create a new rmd file through `New document > R Markdown`
- A popup should appear
- Fill **general information** (type of document, your information, title, date)
- This defines the YAML metadata, which tells how your file is built
- Click **OK**



General structure of the code

Knit on Save | Knit | Run

Source | Visual | Outline

```
---
title: "Untitled"
author: "Samy Zitouni"
date: "2024-11-12"
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)
```

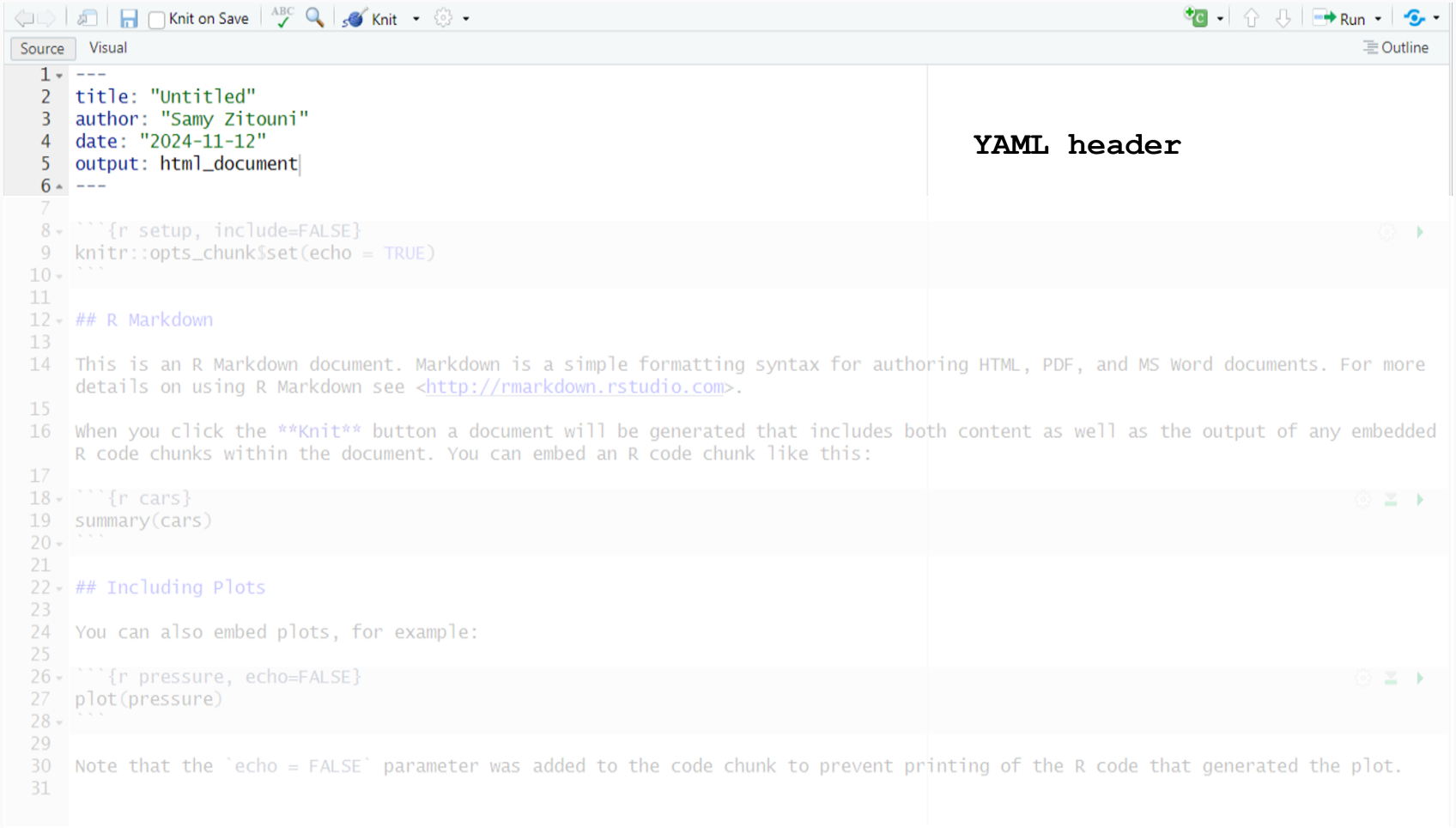
Including Plots

You can also embed plots, for example:

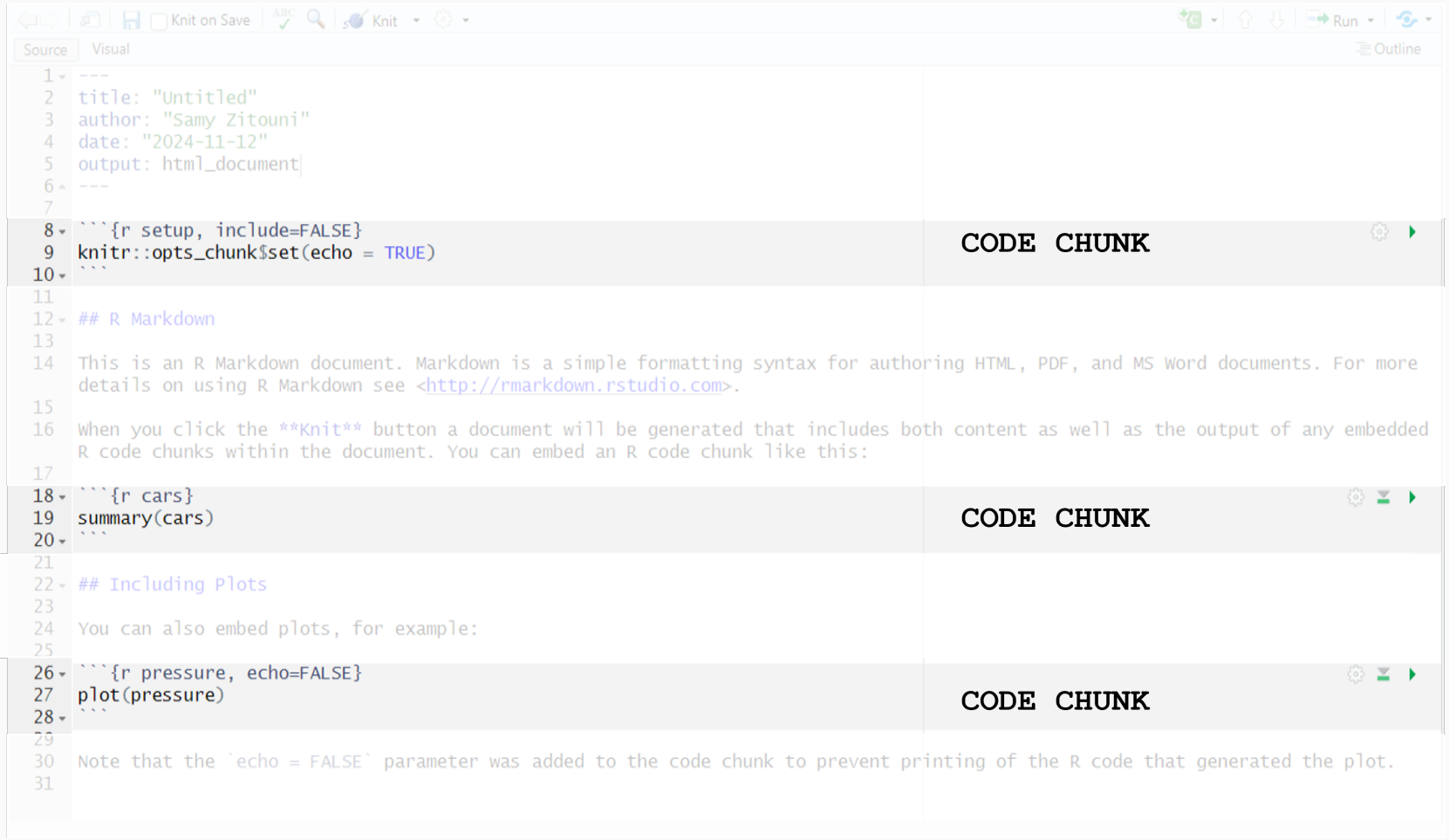
```{r pressure, echo=FALSE}
plot(pressure)
```

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

General structure of the code



General structure of the code



General structure of the code

Let’s run it !

- You can run a chunk you are into : **`Ctrl + Enter`** or clicking on the green arrow at the top right corner of the chunk
- You can run all your code chunks : **`Ctrl + Alt + R`**
- Most importantly, you can run and preview your code : **`Ctrl + Alt + K`** or pressing the ”Knit”button at the top

YAML header

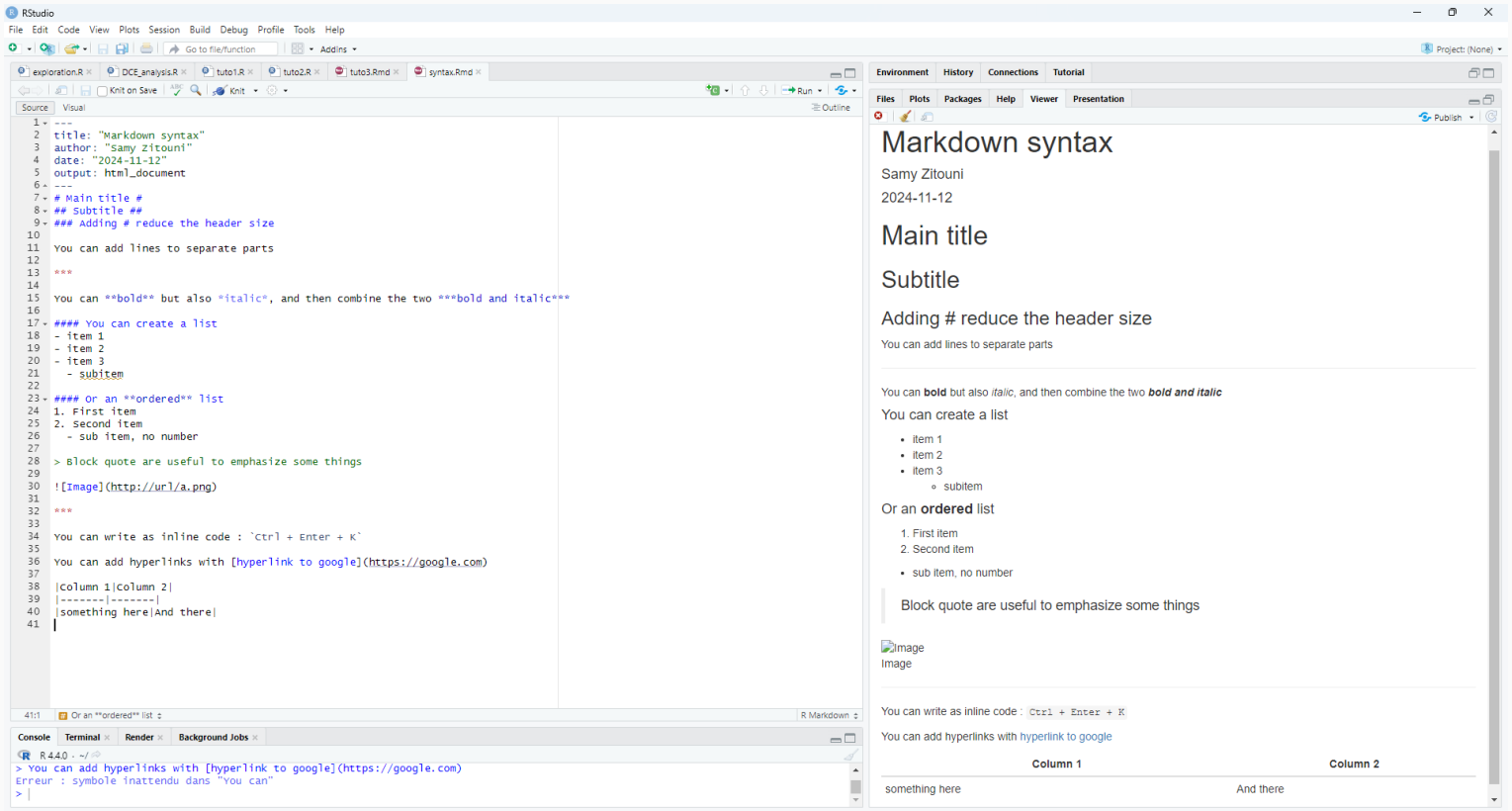
For the configuration of the header of your document

- It contains attributes which values you can set
  - Title, subtitle, author, date, the type of output (html document, word document, pdf document)
  - The yaml header is always between two sets of three dashes `---`

The text syntax

Markdown syntax

- The text parts use the markdown syntax that is common to R Markdown, Jupyter notebooks, etc.



Code chunks

- You can write the fences or simply press add cell on the top right corner

````
```{my r chunk 1}
x <- 5
x+4
```
````

- This code chunk gives :

```r
x <- 5
x+4
```

```
## [1] 9
```

Chunk options

- Every chunk has a set of options, that can be set **TRUE** or **FALSE**
- Every option allows you to **print** or **not print** different part of the code or the output

Option	Default	Effect
eval	TRUE	Evaluate the code and include its results?
echo	TRUE	Display the code and include its results?
warning	TRUE	Display warnings?
error	FALSE	Display errors?
message	TRUE	Display messages?
results	'markup'	How to render code output? 'hide' hides output
fig.width	7	Width in inches for plots created in the chunk
fig.height	7	Height in inches for plots created in the chiunk

More about inline code

- There are places in your report in which you would like to include some results in your text
- Inline code is typically what you want to look at
- It shows the code and you also can execute it directly by adding a `` `r youcodeline` ``

Example

````
```{r number of rows, echo = FALSE}
# Path
path <- 'C:/Users/C09581/Desktop/R tutorials'
path_input <- paste0(path, '/1_inputs/')
path_output <- paste0(path, '/2_outputs/')

# Import data
wid <- read.csv(paste0(path_input, 'wid.csv'))
a <- nrow(wid)
```
In this text, I am talking about a dataframe that contains `a` rows. I could also have written
  ↪    `nrow(wid)` rows directly

In this text, I am talking about a dataframe that contains `r a` rows. I could also have
  ↪    written `r nrow(wid)` rows directly
````

That gives

```
# Path
path <- 'C:/Users/C09581/Desktop/R tutorials'
path_input <- paste0(path, '/1_inputs/')
path_output <- paste0(path, '/2_outputs/')

# Import data
wid <- read.csv(paste0(path_input, 'wid.csv'))
a <- nrow(wid)
```

In this text, I am talking about a dataframe that contains `a` rows. I could also have written `nrow(wid)` rows directly

In this text, I am talking about a dataframe that contains 1610 rows. I could also have written 1610 rows directly

General tips

Showing dataframes in the report

- Use the `"DT"` library
- I don't show it here but `kable(head(data))` is neat

````
```{r my code}
    library("DT")
    datatable(mtcars, options = list(pageLength = 5))
```
````

Show 5 entries Search:

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21	6	160	110	3.9	2.62	16.46	0	1	4	4
Mazda RX4 Wag	21	6	160	110	3.9	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.32	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.44	17.02	0	0	3	2

Showing 1 to 5 of 32 entries Previous 1 2 3 4 5 6 7 Next

General tips

Header parameters

- In my header, I can add a parameter, let's say a name.
- Later in the code, I can access it through inline code `` `r params$name` ``.
- I can generate reports for each name changing one parameter only

```yaml
---
title: "Tuto 3 : R markdown"
author: "Samy Zitouni"
date: "2024-11-12"
output: html_document
params:
  name: "EDGAR"
---
```

Exercise

Objective : plot the evolution of names attribution

- Download the name dataset
- Clean the different columns
- **Summarize** the data by year and and count *the total number* of birth per year
- Choose two names and **compare** the evolution of their attribution at birth
- Style your graphs