

R Markdown & LaTeX

Lecture 5

Louis SIRUGUE

CPES 2 - Fall 2022



Quick reminder

The 3 core components of the ggplot() function

Component	Contribution	Implementation
Data	Underlying values	ggplot(data, data %>% ggplot(.,
Mapping	Axis assignment	aes(x = V1, y = V2, ...))
Geometry	Type of plot	+ geom_point() + geom_line() + ...

- Any **other element** should be added with a **+ sign**

```
ggplot(data, aes(x = V1, y = V2)) +  
  geom_point() + geom_line() +  
  anything_else()
```



Quick reminder

Main customization tools

Item to customize	Main functions
Axes	scale_[x/y]_[continuous/discrete]
Baseline theme	theme_[void/minimal/.../dark]()
Annotations	geom_--[[h/v]line/text](), annotate()
Theme	theme(axis.[line/ticks].[x/y] = ...,

Main types of geometry

Geometry	Function
Bar plot	geom_bar()
Histogram	geom_histogram()
Area	geom_area()
Line	geom_line()
Density	geom_density()
Boxplot	geom_boxplot()
Violin	geom_violin()
Scatter plot	geom_point()



Quick reminder

Main types of aesthetics

Argument	Meaning
alpha	opacity from 0 to 1
color	color of the geometry
fill	fill color of the geometry
size	size of the geometry
shape	shape for geometries like points
linetype	solid, dashed, dotted, etc.

- If specified **in the geometry**
 - It will apply uniformly to **all the geometry**
- If assigned to a variable **in aes**
 - It will **vary with the variable** according to a scale documented in legend

```
ggplot(data, aes(x = V1, y = V2, size = V3)) +  
  geom_point(color = "steelblue", alpha = .6)
```

Warm up practice

- Today we're going to use the "*Fichier des prénoms*"
 - This is where the INSEE reports the **birth count** associated with **each first name in France**
 - It is **virtually exhaustive from 1946**, when the INSEE was founded

```
names <- read.csv("C:/User/Documents/fichier_prenoms.csv", sep = ";", encoding = "UTF-8")
str(names)
```

```
## 'data.frame':    686538 obs. of  4 variables:
## $ sexe      : int  1 1 1 1 1 1 1 1 1 ...
## $ preusuel: chr  "_PRENOMS_RARES" "_PRENOMS_RARES" "_PRENOMS_RARES" "_PRENOMS_RARES" ...
## $ annais   : chr  "1900" "1901" "1902" "1903" ...
## $ nombre   : int  1249 1342 1330 1286 1430 1472 1451 1514 1509 1526 ...
```

- **sex** 1 for Male and 2 for Female
- **preusuel** first name (*_PRENOMS_RARES gathers rare first names for anonymity considerations*)
- **annais** birth year (*XXXX groups unknown birth years*)
- **nombre** number of newborns for the corresponding sex/name/year

Warm up practice

10 : 00

- 1) Recode the `sex` variable with `Male` and `Female` instead of `1` and `2`
- 2) Filter out observations for which `annais` is `XXXX` and convert `annais` to numeric
- 3) Summarise your data into the total number of births per year
- 4) Plot the evolution of the number of births over time using a line geometry

You've got 10 minutes!

Solution

Load the necessary packages

```
library(dplyr)  
library(ggplot2)
```

1) Recode the `sex` variable with `Male` and `Female` instead of `1` and `2`

```
names %>%  
  mutate(sex = ifelse(sex == 1, "Male", "Female"))
```

2) Filter out observations for which `annais` is `XXXX` and convert `annais` to numeric

```
names %>%  
  mutate(sex = ifelse(sex == 1, "Male", "Female")) %>%  
  filter(annais != "XXXX") %>%  
  mutate(annais = as.numeric(annais))
```

Solution

3) Summarise your data into the total number of births per year

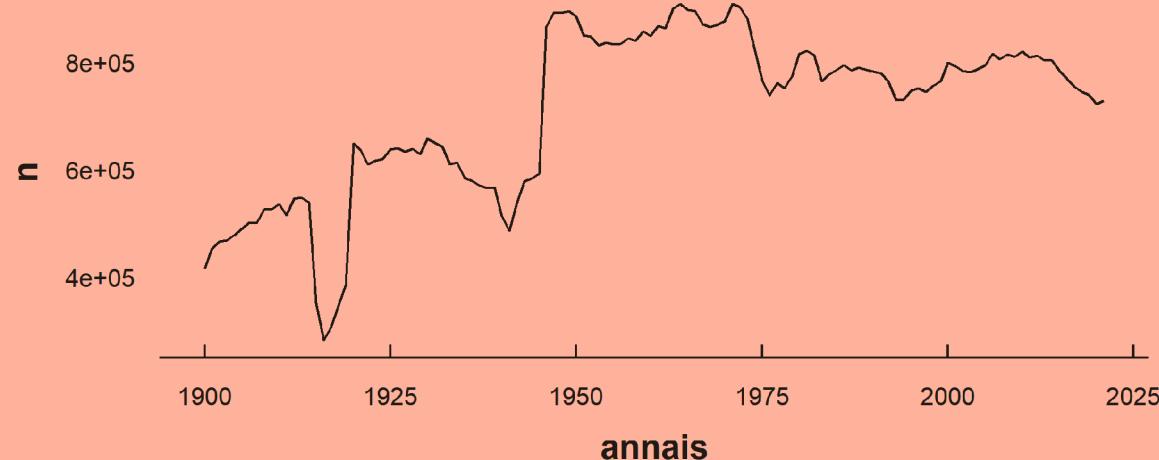
```
names %>%
  mutate(sexe = ifelse(sexe == 1, "Male", "Female")) %>%
  filter(annais != "XXXX") %>%
  mutate(annais = as.numeric(annais)) %>%
  group_by(annais) %>%
  summarise(n = sum(nombre))
```

```
## # A tibble: 8 x 2
##   annais     n
##   <dbl>   <int>
## 1 1900  415040
## 2 1901  453456
## 3 1902  465791
## 4 1903  468810
## 5 1904  478962
## 6 1905  489697
## 7 1906  501745
## 8 1907  501025
```

Solution

4) Plot the evolution of the number of births over time using a line geometry

```
names %>%
  mutate(sexe = ifelse(sexe == 1, "Male", "Female")) %>%
  filter(annais != "XXXX") %>%
  mutate(annais = as.numeric(annais)) %>%
  group_by(annais) %>%
  summarise(n = sum(nombre)) %>%
  ggplot(aes(x = annais, y = n)) + geom_line()
```





Today we learn how to make reports with R Markdown!

1. Basic principles

- 1.1. What is R Markdown?
- 1.2. YAML header
- 1.3. Code chunks
- 1.4. Text formatting
- 1.5. Run and knit your code

3. LaTeX for equations

- 3.1. What is LaTeX?
- 3.2. LaTeX syntax
- 3.3. Large equations

4. Wrap up!

2. Useful features

- 2.1. Inline code
- 2.2. Tables
- 2.3. Preset themes
- 2.4. Report parameters



Today we learn how to make reports with R Markdown!

1. Basic principles

- 1.1. What is R Markdown?
- 1.2. YAML header
- 1.3. Code chunks
- 1.4. Text formatting
- 1.5. Run and knit your code



1. Basic principles

1.1. What is R Markdown?

- **R Markdown** is a type of document in which you can both **write/run R code** and **edit text**
- Here are some examples of R Markdown reports (*from my other course*)
 - Last year homework
 - Example of research project
 - Supplementary material
 - Course webpage and material
- It is structured around **3 types of content**:
 - **Code chunks** to run and render the output
 - **Editable text** to display
 - **YAML metadata** for the R Markdown build process

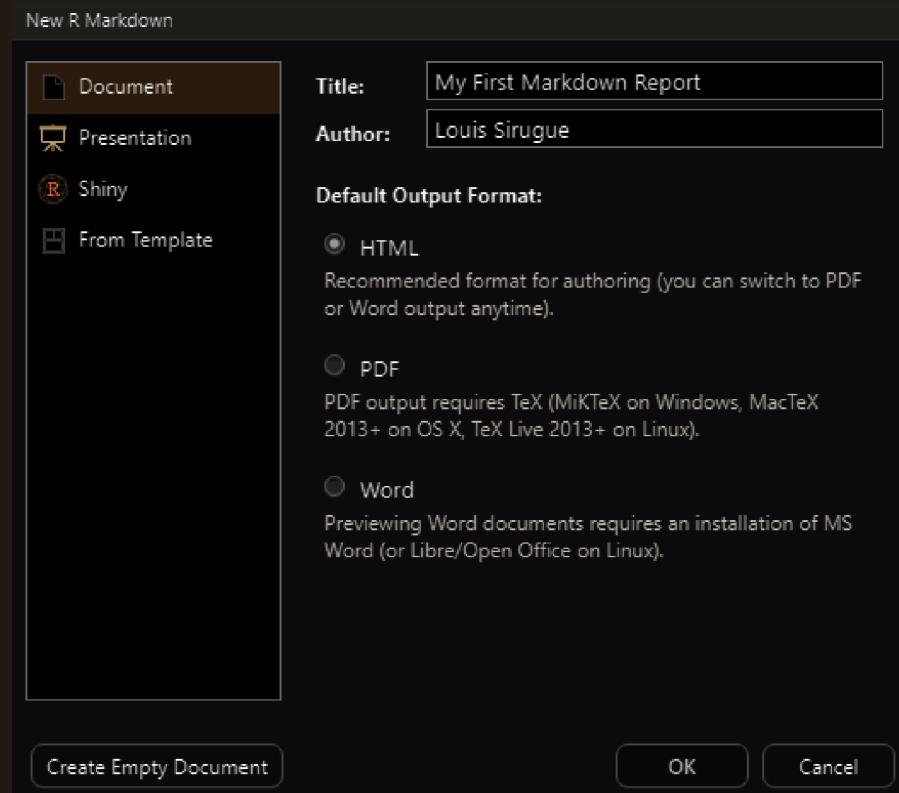
→ *Let's go through them by creating our first R Markdown!*



1. Basic principles

1.1. What is R Markdown?

→ Click on **File > New File > Rmarkdown**



1. Fill out the information and select **HTML**

2. Click on **OK**



1. Basic principles

1.1. What is R Markdown?

- It creates a **template** containing the **3 types of content**:

YAML header →

Code chunks →

Text →

*Let's go through them
one by one!*

```
1 ---  
2 title: "My First Markdown Report"  
3 author: "Louis Sirugue"  
4 date: "24/09/2021"  
5 output: html_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10  
11  
12 ## R Markdown  
13  
14 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>.  
15  
16 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:  
17  
18 ```{r cars}  
19 summary(cars)  
20  
21  
22 ## Including Plots  
23  
24 You can also embed plots, for example:  
25  
26 ```{r pressure, echo=FALSE}  
27 plot(pressure)  
28  
29  
30 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code  
that generated the plot.  
31
```



1. Basic principles

1.2. YAML header

- The **YAML header** contains general information related to the **file configuration**:
 - Title/subtitle (in quotes)
 - Author (in quotes)
 - Date (in quotes)
 - Output type (html_document/pdf_document)
 - ...
- It should be specified at the **very beginning** of the document and surrounded by **three dashes** like so:

```
---
```

```
title: "My First Markdown Report"
author: "Louis Sirugue"
date: "24/09/2021"
output: html_document
---
```



1. Basic principles

1.3. Code chunks

- **Code chunks** are **blocks of R code** that can be run when working on and rendering the .Rmd file
- You can insert a code chunk using `Ctrl + Alt + i` or by typing the **backticks chunk delimiters** as follows

```
```{r}
1+1
```
```

- When **rendering** the document, R will **execute** the code
 - Both the **code** and the **output** will appear in the document like so:

```
1+1
```

```
## [1] 2
```



1. Basic principles

1.3. Code chunks

- The **content** to be **displayed** from the code chunk can be specified in **chunk options**
 - For instance, to display only the output and not the code chunk, you can set echo to FALSE

```
```{r, echo = F}
1+1
```
```

- And the output will only be

```
## [1] 2
```

- Instead of

```
1+1
```

```
## [1] 2
```



1. Basic principles

1.3. Code chunks

Chunk options to know

| Option | Default | Effect |
|------------|----------|--|
| eval | TRUE | Whether to evaluate the code and include its results |
| echo | TRUE | Whether to display code along with its results |
| warning | TRUE | Whether to display warnings |
| error | TRUE | Whether to display errors |
| message | TRUE | Whether to display messages |
| results | 'markup' | 'hide' to hide the output |
| fig.width | 7 | Width in inches for plots created in chunk |
| fig.height | 7 | Height in inches for plots created in chunk |



1. Basic principles

1.4. Text formatting

- R Markdown is not only about rendering code but also about **writing** actual **text**
 - You can write **paragraphs** as you would normally do on a typical report
 - And R Markdown provides convenient ways to **format** your text
- Basic formatting includes:
 - Italics
 - Bold
 - Hyperlinks
 - Headers
 - Block quotes
 - Un/ordered lists
 - ...
- Unlike most text editing software, in R Markdown **text formatting** isn't about clicking on dedicated buttons
 - It **relies on symbols** that should be written along with the text



1. Basic principles

1.4. Text formatting

Syntax

```
Plain text
```

```
End a line with two spaces for line break
```

```
*italics*
```

```
**bold**
```

```
# Header 1
```

```
## Header 2
```

```
...
```

```
##### Header 6
```

```
[link] (https://www.rstudio.com)
```

Output

Plain text

End a line with two spaces for line break

italics

bold

Header 1

Header 2

...

Header 6

link



1. Basic principles

1.4. Text formatting

Syntax

```
> block quote
```

Horizontal rule:

```
***
```

```
* unordered list
* item 2
  + sub-item 1
  + sub-item 2
```

```
1. ordered list
2. item 2
  + sub-item 1
  + sub-item 2
```

Output

```
| block quote
```

Horizontal rule:

- unordered list
- item 2
 - sub-item 1
 - sub-item 2

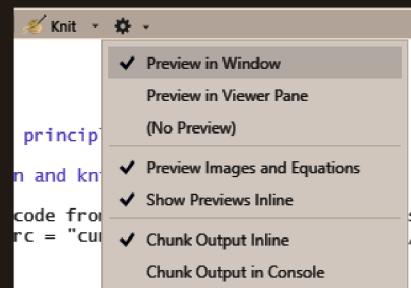
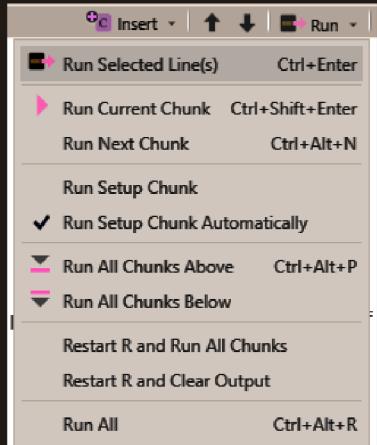
1. ordered list
2. item 2
 - sub-item 1
 - sub-item 2



1. Basic principles

1.5. Run and knit your code

- To **execute** the content of a **code** chunk in R Markdown
 - Click on the **green play button** at the top right of the chunk 
- You can also:
 - **Run all chunks above** the current chunk 
 - **Run all chunks** from the Run drop down menu at the top right (or Ctrl+Alt+R)
- To choose where the **output** must be **displayed**, click on the "*Options*" button
 - **Chunk output inline:** output displayed right below the chunk in the **source panel**
 - **Chunk output in console:** output displayed in **console panel**





1. Basic principles

1.5. Run and knit your code

- To **render** an R Markdown file, click on the **knit button** (ctrl + shift + k)

```
1 ---  
2 title: "My First Markdown Report"  
3 author: "Louis Sirugue"  
4 date: "24/09/2021"  
5 output: html_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10   
11  
12 ## R Markdown  
13  
14 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>.  
15  
16 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:  
17  
18 ```{r cars}  
19 summary(cars)  
20   
21  
22 ## Including Plots  
23  
24 You can also embed plots, for example:  
25  
26 ```{r pressure, echo=FALSE}  
27 plot(pressure)  
28   
29 Note that the `echo = FALSE` parameter was added to the code chunk  
30
```

My First Markdown Report

Louis Sirugue

24/09/2021

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
```



1. Basic principles

1.5. Run and knit your code

- To **render** an R Markdown file, click on the **knit button** (ctrl + shift + k)

The image shows a comparison between the RStudio code editor and the resulting knitted document.

Code Editor (Left):

```
21
22 ## Including Plots
23
24 You can also embed plots, for example:
25
26 ```{r pressure, echo=FALSE}
27 plot(pressure)
28
29 Note that the `echo = FALSE` parameter was added to the code chunk
30 to prevent printing of the R code that generated the plot.
31
```

Knitted Document (Right):

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.



Overview

1. Basic principles ✓

- 1.1. What is R Markdown?
- 1.2. YAML header
- 1.3. Code chunks
- 1.4. Text formatting
- 1.5. Run and knit your code

3. LaTeX for equations

- 3.1. What is LaTeX?
- 3.2. LaTeX syntax
- 3.3. Large equations

4. Wrap up!

2. Useful features

- 2.1. Inline code
- 2.2. Tables
- 2.3. Preset themes
- 2.4. Report parameters



Overview

1. Basic principles ✓

- 1.1. What is R Markdown?
- 1.2. YAML header
- 1.3. Code chunks
- 1.4. Text formatting
- 1.5. Run and knit your code

2. Useful features

- 2.1. Inline code
- 2.2. Tables
- 2.3. Preset themes
- 2.4. Report parameters



2. Useful features

2.1. Inline code

- A big advantage of R Markdown is that you can **automate** your **reports**

Why is it useful?

- You might figure out quite late in the process that you need to **make a change** at the beginning of the analysis
 - A change that potentially **impacts everything** that comes after in the report
- Imagine that you forgot to filter out an irrelevant group of observations at the beginning
 - If you simply filter your data at the beginning in a code chunk
 - All your tables and figures will **update automatically**
- But what if you wrote some of your results **within paragraphs**?
 - In a usual text formatting software you would have to update everything manually
 - But here you can also make it **update automatically!**



2. Useful features

2.1. Inline code

- Consider the following report :

```
1 ---  
2 title: "Report example"  
3 author: "Louis Sirugue"  
4 date: "26/09/2021"  
5 output: html_document  
---  
7  
8 ## overview of the data  
9  
10 ```{r cars}  
11 names(cars)  
12 dim(cars)  
13 c(mean(cars$speed), mean(cars$dist))  
14 ````  
15  
16 The dataset we consider contains two variables, speed and distance, and has 50 observations. The average speed value is 15.4 and the average distance value is 42.98.
```

Report example

Louis Sirugue

26/09/2021

Overview of the data

```
names(cars)
```

```
## [1] "speed" "dist"
```

```
dim(cars)
```

```
## [1] 50  2
```

```
c(mean(cars$speed), mean(cars$dist))
```

```
## [1] 15.40 42.98
```

The dataset we consider contains two variables, speed and distance, and has 50 observations. The average speed value is 15.4 and the average distance value is 42.98.



2. Useful features

2.1. Inline code

- Imagine that there is a problem with the observation for which `dist > 100` and that you should discard it

The screenshot shows a dark-themed R Markdown report. On the left, the R code is displayed with line numbers:

```
1 ---  
2 title: "Report example"  
3 author: "Louis Sirugue"  
4 date: "26/09/2021"  
5 output: html_document  
6 ---  
7  
8 ## Overview of the data  
9  
10 ````{r cars}  
11 # Omit if distance >= 100  
12 cars <- cars[cars$dist < 100, ]  
13 names(cars)  
14 dim(cars)  
15 c(mean(cars$speed), mean(cars$dist))  
16  
17  
18 The dataset we consider contains two variables, speed and distance, and has 50 observations. The average speed value is 15.4 and the average distance value is 42.98.
```

The right side of the interface displays the generated report content. It includes a title, author, date, and an "Overview of the data" section. The "Overview of the data" section contains the R code and its output, which is identical to the code shown on the left. Below this, a summary sentence is displayed.

Report example
Louis Sirugue
26/09/2021

Overview of the data

```
# Omit if distance >= 100  
cars <- cars[cars$dist < 100, ]  
names(cars)
```

```
## [1] "speed" "dist"
```

```
dim(cars)
```

```
## [1] 49  2
```

```
c(mean(cars$speed), mean(cars$dist))
```

```
## [1] 15.22449 41.40816
```

The dataset we consider contains two variables, speed and distance, and has 50 observations. The average speed value is 15.4 and the average distance value is 42.98.



2. Useful features

2.1. Inline code

- All the results were updated automatically but not the text
 - That's where **inline code** comes in!

→ **Inline code** allows to include the output of some **R code within text areas** of your report

- R code outside code chunks should be included between backticks:
 - Surrounding code with **backticks** in a text area will **change** the **format** to that of the code chunk
 - **Adding** the **r** letter right after the first backtick will **show** the **output** of the code instead of the code

Syntax

```
`paste("a", "b", sep = "-")`
```

Output

```
paste("a", "b", sep = "-")
```

```
`r paste("a", "b", sep = "-")`
```

a-b



2. Useful features

2.1. Inline code

- With inline code, **paragraphs** also do **update automatically**:

```
1 v ---
2 title: "Report example"
3 author: "Louis Sirugue"
4 date: "26/09/2021"
5 output: html_document
6 ^
7
8 ## overview of the data
9
10 {r cars}
11 # Omit if distance >= 100
12 cars <- cars[cars$dist < 100, ]
13 names(cars)
14 dim(cars)
15 c(mean(cars$speed), mean(cars$dist))
16
17
18 The dataset we consider contains two variables, speed and distance, and has `r dim(cars)[1]` observations. The average speed value is `r mean(cars$speed)` and the average distance value is `r mean(cars$dist)`.
```

Report example

Louis Sirugue

26/09/2021

Overview of the data

```
# Omit if distance >= 100
cars <- cars[cars$dist < 100, ]
names(cars)

## [1] "speed" "dist"

dim(cars)

## [1] 49  2

c(mean(cars$speed), mean(cars$dist))

## [1] 15.22449 41.40816
```

The dataset we consider contains two variables, speed and distance, and has 49 observations. The average speed value is 15.2244898 and the average distance value is 41.4081633.



2. Useful features

2.2. Tables

- Displaying a table as a raw output can be unpleasant to read

```
head(mtcars)
```

```
##          mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4     21.0   6 160 110 3.90 2.620 16.46  0  1     4     4
## Mazda RX4 Wag 21.0   6 160 110 3.90 2.875 17.02  0  1     4     4
## Datsun 710    22.8   4 108  93 3.85 2.320 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 3.440 17.02  0  0     3     2
## Valiant       18.1   6 225 105 2.76 3.460 20.22  1  0     3     1
```

- The `kable()` function from the `knitr` package allows to display tables in a nice way

```
library("knitr")
```



2. Useful features

2.2. Tables

- You just need to put the table you want to display inside the `kable()` function

```
kable(head(mtcars), caption = "First rows of the dataset")
```

| | mpg | cyl | disp | hp | drat | wt | qsec | vs | am | gear | carb |
|-------------------|------|-----|------|-----|------|------|-------|----|----|------|------|
| Mazda RX4 | 21.0 | 6 | 160 | 110 | 3.90 | 2.62 | 16.46 | 0 | 1 | 4 | 4 |
| Mazda RX4 Wag | 21.0 | 6 | 160 | 110 | 3.90 | 2.88 | 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 | 3.21 | 19.44 | 1 | 0 | 3 | 1 |
| Hornet Sportabout | 18.7 | 8 | 360 | 175 | 3.15 | 3.44 | 17.02 | 0 | 0 | 3 | 2 |
| Valiant | 18.1 | 6 | 225 | 105 | 2.76 | 3.46 | 20.22 | 1 | 0 | 3 | 1 |



2. Useful features

2.2. Tables

- For **big tables**, one solution is the `datatable()` function from the `DT` package
- As with `kable()`, you just need to put the table you want to display inside the `datatable()` function

```
library("DT")
datatable(mtcars)
```

- The output will be an **interactive table** which allows to:
 - Navigate in the table by displaying a limited number of rows at a time
 - Choose the number of rows to display
 - Search for a given element in the table
- You can select the default number of rows to display as follows

```
datatable(mtcars, options = list(pageLength = 5))
```



2. Useful features

2.2. Tables

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

→ Try to search for "Toyota" for instance



2. Useful features

2.3. Preset themes

- The **default theme** of R Markdown might seem **a bit dull**
 - The look of your reports can easily be **enhanced** using a variety of **preset** themes
 - The preset theme to use should be specified in the **YAML header**
 - Add a theme argument to the `html_document` format specified as output

```
---
```

```
title: "My First Markdown Report"
author: "Louis Sirugue"
date: "24/09/2021"
output:
  html_document:
    theme: cosmo
---
```

- When using themes from downloaded packages, the way you set the theme can be slightly different
 - Check the online documentation



2. Useful features

2.3. Preset themes

cosmo

The screenshot shows a dark-themed web page titled "Document Title". The header includes "Author Name" and the date "4/11/2018". A "Level-1 Header" section contains a paragraph about blind texts. Below it, a "Level-2 Header" section contains a paragraph from Kafka's "Metamorphosis".

Document Title

Author Name
4/11/2018

Level-1 Header

Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by their place and supplies it with the necessary regelialia. It is a paradesemic country, in which roasted parts of sentences fly into your mouth. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas, wild Question Marks and devious Semikoli, but the Little Blind Text didn't listen. She packed her seven versalia, put her initial into the belt and made herself on the way. When she reached the first hills of the Iitalic Mountains, she had a last view back on the skyline of her hometown Bookmarksgrove, the headline of Alphabet Village and the subline of her own road, the Line Lane.

Level-2 Header

One morning, when Gregor Samsa woke from troubled dreams, he found himself transformed in his bed into a horrible vermin. He lay on his armour-like back, and if he lifted his head a little he could see his brown belly, slightly domed and divided by arches into stiff sections. The bedding was hardly able to cover it and seemed ready to slide off any moment. His many legs, pitifully thin compared with the size of the rest of him, waved about helplessly as he looked. "What's happened to me?" he thought. It wasn't a dream. His room, a proper human room although a little too small, lay peacefully between its four familiar walls. A collection of textile samples lay spread out on the table - Samsa was a travelling salesman - and above it there hung a picture that he had recently cut out of an illustrated magazine and housed in a nice, gilded frame. It showed a lady fitted out with a fur hat and fur boa who sat upright, raising a heavy fur muff that covered the whole of her lower arm towards the viewer. Gregor then turned to look out the window at the dull weather.

cerulean

The screenshot shows a dark-themed web page titled "Document Title". The header includes "Author Name" and the date "4/11/2018". A "Level-1 Header" section contains a paragraph about blind texts. Below it, a "Level-2 Header" section contains a paragraph from Kafka's "Metamorphosis".

Document Title

Author Name
4/11/2018

Level-1 Header

Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by their place and supplies it with the necessary regelialia. It is a paradesemic country, in which roasted parts of sentences fly into your mouth. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas, wild Question Marks and devious Semikoli, but the Little Blind Text didn't listen. She packed her seven versalia, put her initial into the belt and made herself on the way. When she reached the first hills of the Iitalic Mountains, she had a last view back on the skyline of her hometown Bookmarksgrove, the headline of Alphabet Village and the subline of her own road, the Line Lane.

Level-2 Header

One morning, when Gregor Samsa woke from troubled dreams, he found himself transformed in his bed into a horrible vermin. He lay on his armour-like back, and if he lifted his head a little he could see his brown belly, slightly domed and divided by arches into stiff sections. The bedding was hardly able to cover it and seemed ready to slide off any moment. His many legs, pitifully thin compared with the size of the rest of him, waved about helplessly as he looked. "What's happened to me?" he thought. It wasn't a dream. His room, a proper human room although a little too small, lay peacefully between its four familiar walls. A collection of textile samples lay spread out on the table - Samsa was a travelling salesman - and above it there hung a picture that he had recently cut out of an illustrated magazine and housed in a nice, gilded frame. It showed a lady fitted out with a fur hat and fur boa who sat upright, raising a heavy fur muff that covered the whole of her lower arm towards the viewer. Gregor then turned to look out the window at the dull weather.



2. Useful features

2.3. Preset themes

cayman (from prettydoc)

Creating Pretty Documents From R Markdown

The Cayman Theme

The `prettydoc` package provides an alternative engine, `html_pretty`, to knit your R Markdown document into pretty HTML pages. Its usage is extremely easy: simply replace the `rmarkdown::html_document` or `rmarkdown::html_vignette` output engine by `prettydoc::html_pretty` in your R Markdown header, and use one of the built-in themes and syntax highlighters.

Elements

We demonstrate some commonly used HTML elements here to show the appearance of themes.

Tables

| | Df | Sum Sq | Mean Sq | F value | Pr(>F) | |
|-----------|----|--------|---------|---------|---------|----|
| Block | 5 | 343.3 | 68.66 | 4.288 | 0.01272 | * |
| N | 1 | 189.3 | 189.28 | 11.821 | 0.00366 | ** |
| P | 1 | 8.4 | 8.40 | 0.525 | 0.47999 | |
| K | 1 | 95.2 | 95.20 | 5.946 | 0.02767 | * |
| Residuals | 15 | 240.2 | 16.01 | | | |

Code

Familiar `knitr` R code and plots:

```
set.seed(123)
n <- 1000
x1 <- matrix(rnorm(n), ncol = 2)
x2 <- matrix(rnorm(n, mean = 3, sd = 1.5), ncol = 2)
x <- rbind(x1, x2)
par(mar = c(4, 4, 1, 2))
smoothScatter(x, xlab = "x1", ylab = "x2")
```

tactile (from prettydoc)

Creating Pretty Documents From R Markdown

The Tactile Theme

The `prettydoc` package provides an alternative engine, `html_pretty`, to knit your R Markdown document into pretty HTML pages. Its usage is extremely easy: simply replace the `rmarkdown::html_document` or `rmarkdown::html_vignette` output engine by `prettydoc::html_pretty` in your R Markdown header, and use one of the built-in themes and syntax highlighters.

Elements

We demonstrate some commonly used HTML elements here to show the appearance of themes.

Tables

| | Df | Sum Sq | Mean Sq | F value | Pr(>F) | |
|-----------|----|--------|---------|---------|---------|----|
| Block | 5 | 343.3 | 68.66 | 4.288 | 0.01272 | * |
| N | 1 | 189.3 | 189.28 | 11.821 | 0.00366 | ** |
| P | 1 | 8.4 | 8.40 | 0.525 | 0.47999 | |
| K | 1 | 95.2 | 95.20 | 5.946 | 0.02767 | * |
| Residuals | 15 | 240.2 | 16.01 | | | |

Code

Familiar `knitr` R code and plots:

```
set.seed(123)
n <- 1000
x1 <- matrix(rnorm(n), ncol = 2)
x2 <- matrix(rnorm(n, mean = 3, sd = 1.5), ncol = 2)
x <- rbind(x1, x2)
par(mar = c(4, 4, 1, 2))
smoothScatter(x, xlab = "x1", ylab = "x2")
```



2. Useful features

2.3. Preset themes

leonids (from prettydoc)

The `prettydoc` package provides an alternative engine, `html_pretty`, to knit your R Markdown document into pretty HTML pages. Its usage is extremely easy: simply replace the `rmarkdown::html_document` or `rmarkdown::html_vignette` output engine by `prettydoc::html_pretty` in your R Markdown header; and use one of the built-in themes and syntax highlighters.

Elements

We demonstrate some commonly used HTML elements here to show the appearance of themes.

Tables

| | Df | Sum Sq | Mean Sq | F value | Pr(>F) | |
|-----------|----|--------|---------|---------|---------|----|
| Block | 5 | 343.3 | 68.66 | 4.447 | 0.01594 | * |
| N | 1 | 189.3 | 189.28 | 12.259 | 0.00437 | ** |
| P | 1 | 8.4 | 8.40 | 0.544 | 0.47490 | |
| K | 1 | 95.2 | 95.20 | 6.166 | 0.02880 | * |
| N:P | 1 | 21.3 | 21.28 | 1.378 | 0.26317 | |
| N:K | 1 | 33.1 | 33.14 | 2.146 | 0.16865 | |
| P:K | 1 | 0.5 | 0.48 | 0.031 | 0.86275 | |
| Residuals | 12 | 185.3 | 15.44 | | | |



2. Useful features

2.3. Preset themes

downcute (from rmdformats)

Code and tables

Syntax highlighting

Here is a sample code chunk, just to show that syntax highlighting works as expected.

```
library(palmerpenguins)
library(glue)

say_hello <- function (name) {
  glue("Hello, {name}!")
}

say_hello("world")
```

Hello, world !

Verbatim

Here is the structure of the `penguins` dataset.

```
str(penguins)
```

downcute chaos (from rmdformats)

Code and tables

Syntax highlighting

Here is a sample code chunk, just to show that syntax highlighting works as expected.

```
say_hello <- function (name) {
  paste("Hello,", name, "!")
}

say_hello("world")
```

[1] "Hello, world !"

Verbatim

Here is the structure of the `penguins` dataset.



2. Useful features

2.3. Preset themes

readthedown (from rmdformats)

[readthedown template example](#)

[Code](#)

Code and tables

Styling

Figures

readthedown template example

Code and tables

Syntax highlighting

Here is a sample code chunk, just to show that syntax highlighting works as expected.

[Code](#)

```
Hello, world!
```

Verbatim

Here is the structure of the `penguins` dataset.

[Code](#)

```
tibble [344 x 8] (S3:tbl_df/tbl/data.frame)
$ species      : Factor w/ 3 levels "Adelie","Chinstrap","Gentoo": 1 1 1 1 1 1 1 1 1 ...
$ island       : Factor w/ 3 levels "Biscoe","Dream","Gentoo": 3 3 3 3 3 3 3 3 3 ...
$ bill_length_mm: num [1:344] 18.7 19.5 40.0 NA 36.7 39.8 34.9 39.2 34.1 42 ...
$ bill_depth_mm: num [1:344] 18.7 17.4 18.0 NA 19.3 20.6 17.8 19.6 18.1 20.2 ...
$ flipper_length_mm: int [1:344] 181 186 195 NA 193 198 181 195 193 190 ...
$ body_mass_g   : int [1:344] 3750 3800 3250 NA 3450 3650 3625 4675 3475 4250 ...
$ sex          : Factor w/ 2 levels "Female","Male": 2 1 1 NA 1 2 1 2 NA NA ...
$ year         : int [1:344] 2007 2007 2007 2007 2007 2007 2007 2007 2007 2007 ...
```

Table

Sample table output.

robobook (from rmdformats)

robobook template example

Code and tables

Mathjax

Figures

robobook template example

Code and tables

Syntax highlighting

Here is a sample code chunk, just to show that syntax highlighting works as expected.

library(palmerpenguins)
library(glue)

say_hello <- function (name) {
 glue("Hello, {name}!")
}

say_hello("world")

Hello, world!

Verbatim

Here is the structure of the `penguins` dataset.

str(penguins)

Practice

Reproduce the following html using R markdown

Copy raw output

You've got 15 minutes!

15 : 00

Report on the first name LOUIS

Your name

Fall 2022

1. Setup

The packages needed in an Rmd must *always* be loaded in a code chunk at the beginning of the file.

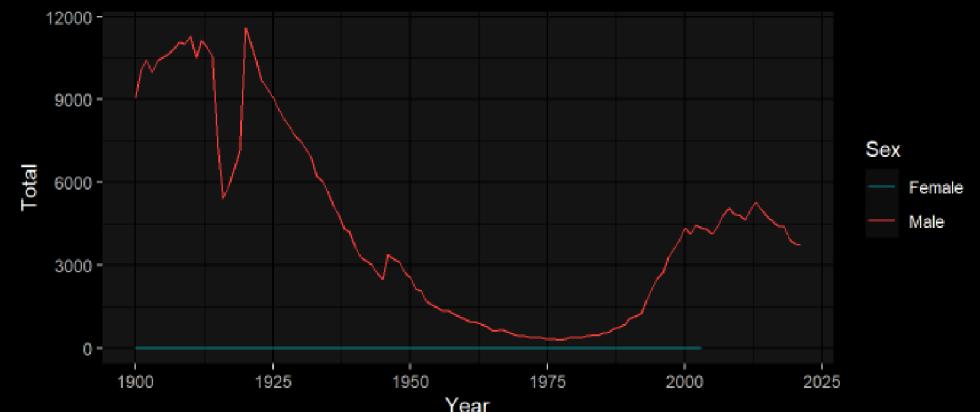
```
library(dplyr)  
library(ggplot2)
```

However, the command `install.packages()` must **not** be written in an R markdown. It should be run only once in the console.

2. Data cleaning

```
names <- read.csv("fichier_prenoms.csv", encoding = "UTF-8", sep = ";") %>%  
  mutate(sexe = ifelse(sexe == 1, "Male", "Female")) %>%  
  rename(Sex = sexe, Year = annais) %>%  
  filter(Year != "XXXX") %>%  
  mutate(Year = as.numeric(Year))
```

3. Evolution of the first name LOUIS over time



3715 children were born under the name LOUIS in 2021. This statistic is written in **inline code** such that it updates automatically.

Solution

```
---
```

```
title: "Report on the first name LOUIS"
author: "Your name"
date: "Fall 2022"
output:
  html_document:
    theme: cosmo
---
```

1) Setup

The packages needed in an Rmd must **always** be loaded in a code chunk at the beginning of the file.

```
```{r, message = F, warning = F}
library(dplyr)
library(ggplot2)
```
```

However, the command `install.packages()` must ****not**** be written in an R markdown.
It should be run only once in the console.

Solution

```
### 2) Data cleaning

```{r}
names <- read.csv("fichier_prenoms.csv", encoding = "UTF-8", sep = ";") %>%
 mutate(sexe = ifelse(sexe == 1, "Male", "Female")) %>%
 rename(Sex = sexe, Year = annais) %>%
 filter(Year != "XXXX") %>%
 mutate(Year = as.numeric(Year))
```

### 3) Evolution of the first name LOUIS over time

```{r, echo = F, message = F, fig.height = 3}
names %>%
 filter(preusuel == "LOUIS") %>%
 group_by(Sex, Year) %>%
 summarise(Total = sum(nombre)) %>%
 ggplot(aes(x = Year, y = Total, color = Sex)) +
 geom_line()
```

`r sum(names[names$preusuel=='LOUIS' & names$Year==2021, 'nombre'])` children were born under the
name LOUIS in 2021. This statistic is written in **inline code** such that it updates automatically
```



2. Useful features

2.4. Report parameters

- It may sometimes be useful to produce **separate html reports for different groups** in your data
 - Country/state-specific reports
 - Here, a different report for each first name
- **YAML parameters** are very useful for that
 - They are accessible **like any object** in your environment
 - They must be specified as follows

```
title: "Report on the first name `r params$name`"
author: "Your name"
date: "Fall 2022"
output:
  html_document:
    theme: cosmo
params:
  name: "LOUIS"
```



2. Useful features

2.4. Report parameters

- You simply have to call that object in your code chunks or inline code when needed

```
### 3) Evolution of the first name `r params$name` over time
```

```
```{r, echo = F, message = F, fig.height = 3}
names %>%
 filter(preusuel == params$name) %>%
 group_by(Sex, Year) %>%
 summarise(Total = sum(nombre)) %>%
 ggplot(aes(x = Year, y = Total, color = Sex)) +
 geom_line()
```

```

`r sum(names[names\$preusuel == params\$name & names\$Year == 2021, 'nombre'])` children were born under the name `r params\$name` in 2021. This statistic is written in ****inline code**** such that it updates automatically.

→ Let's knit our .Rmd with different values of that parameter!

Report on the first name LOUIS

Your name

Fall 2022

1. Setup

The packages needed in an Rmd must *always* be loaded in a code chunk at the beginning of the file.

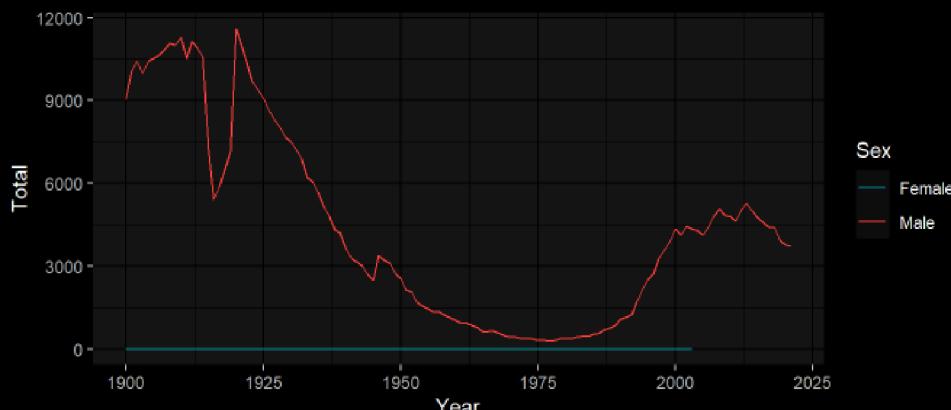
```
library(dplyr)  
library(ggplot2)
```

However, the command `install.packages()` must **not** be written in an R markdown.
It should be run only once in the console.

2. Data cleaning

```
names <- read.csv("fichier_prenoms.csv", encoding = "UTF-8", sep = ";") %>%  
  mutate(sexe = ifelse(sexe == 1, "Male", "Female")) %>%  
  rename(Sex = sexe, Year = annais) %>%  
  filter(Year != "XXXX") %>%  
  mutate(Year = as.numeric(Year))
```

3. Evolution of the first name LOUIS over time



3715 children were born under the name LOUIS in 2021. This statistic is written in **inline code** such that it updates automatically.

Report on the first name DIDIER

Your name

Fall 2022

1. Setup

The packages needed in an Rmd must *always* be loaded in a code chunk at the beginning of the file.

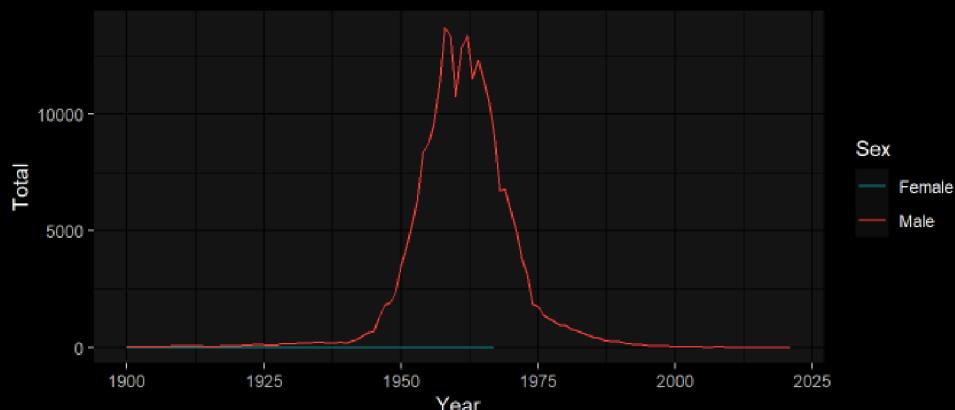
```
library(dplyr)  
library(ggplot2)
```

However, the command `install.packages()` must **not** be written in an R markdown.
It should be run only once in the console.

2. Data cleaning

```
names <- read.csv("fichier_prenoms.csv", encoding = "UTF-8", sep = ";") %>%  
  mutate(sexe = ifelse(sexe == 1, "Male", "Female")) %>%  
  rename(Sex = sexe, Year = annais) %>%  
  filter(Year != "XXXX") %>%  
  mutate(Year = as.numeric(Year))
```

3. Evolution of the first name DIDIER over time



3 children were born under the name DIDIER in 2021. This statistic is written in **inline code** such that it updates automatically.

Report on the first name PAULINE

Your name

Fall 2022

1. Setup

The packages needed in an Rmd must *always* be loaded in a code chunk at the beginning of the file.

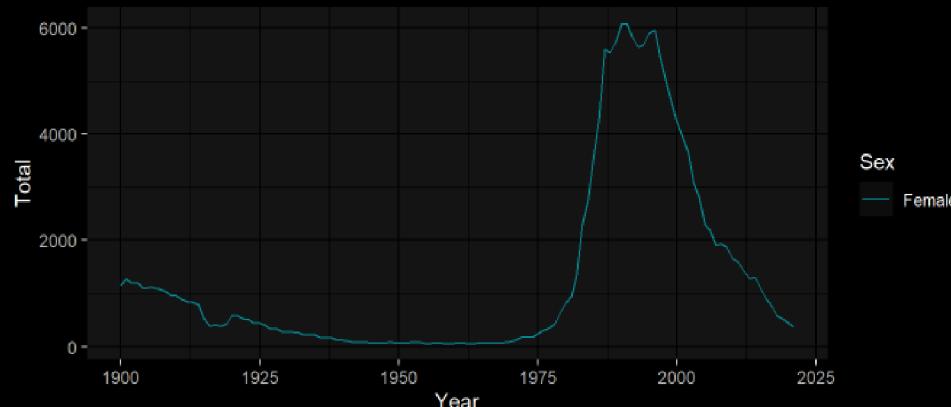
```
library(dplyr)  
library(ggplot2)
```

However, the command `install.packages()` must **not** be written in an R markdown.
It should be run only once in the console.

2. Data cleaning

```
names <- read.csv("fichier_prenoms.csv", encoding = "UTF-8", sep = ";") %>%  
  mutate(sexe = ifelse(sexe == 1, "Male", "Female")) %>%  
  rename(Sex = sexe, Year = annais) %>%  
  filter(Year != "XXXX") %>%  
  mutate(Year = as.numeric(Year))
```

3. Evolution of the first name PAULINE over time



366 children were born under the name PAULINE in 2021. This statistic is written in **inline code** such that it updates automatically.

Report on the first name CAMILLE

Your name

Fall 2022

1. Setup

The packages needed in an Rmd must *always* be loaded in a code chunk at the beginning of the file.

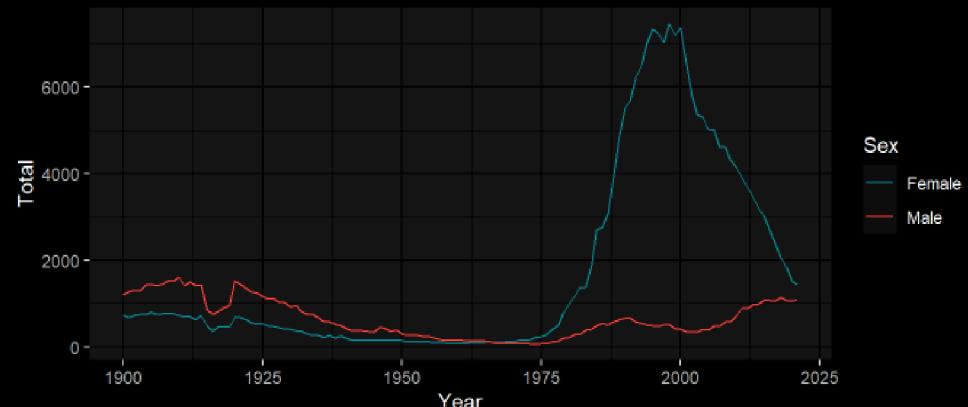
```
library(dplyr)  
library(ggplot2)
```

However, the command `install.packages()` must **not** be written in an R markdown.
It should be run only once in the console.

2. Data cleaning

```
names <- read.csv("fichier_prenoms.csv", encoding = "UTF-8", sep = ";") %>%  
  mutate(sexe = ifelse(sexe == 1, "Male", "Female")) %>%  
  rename(Sex = sexe, Year = annais) %>%  
  filter(Year != "XXXX") %>%  
  mutate(Year = as.numeric(Year))
```

3. Evolution of the first name CAMILLE over time



2524 children were born under the name CAMILLE in 2021. This statistic is written in **inline code** such that it updates automatically.



2. Useful features

2.4. Report parameters

- But by **default** the **name of the .html** output will be the name of your .Rmd
 - So if you **knit report.Rmd** for the first name Louis it will save the report under **report.html**
 - And if you **knit it a second time** for the first name Didier it will **override** the first .html
- The **solution** is to **knit** your .Rmd **externally**
 - You can do that with the **render()** function of the rmarkdown package
 - Save your .Rmd and **open a new .R script** to try it out

```
library(rmarkdown)  
  
render(  
  input = "C:/User/Documents/prenom.Rmd",                      # Specify the input .Rmd  
  output_file = "C:/User/Documents/LOUIS.html",                  # Specify the output file  
  params = list(name = "LOUIS")                                    # Specify the YAML parameter(s)  
)
```



2. Useful features

2.4. Report parameters

- To **avoid copy-pasting** this command for each name we want a report on, we must **use a loop**

- 1.
- 2.
- 3.
- 4.

```
for ( in ) {  
}  
}
```



2. Useful features

2.4. Report parameters

- To **avoid copy-pasting** this command for each name we want a report on, we must **use a loop**
 1. First we should name the object that will successively take the value of each first name
 - 2.
 - 3.
 - 4.

```
for (i in ) {  
}  
}
```



2. Useful features

2.4. Report parameters

- To **avoid copy-pasting** this command for each name we want a report on, we must **use a loop**
 1. First we should name the object that will successively take the value of each first name
 2. Then indicate which values this object must successively take
 - 3.
 - 4.

```
for (i in c("LOUIS", "DIDER", "PAULINE", "CAMILLE")) {  
}  
}
```



2. Useful features

2.4. Report parameters

- To **avoid copy-pasting** this command for each name we want a report on, we must **use a loop**
 1. First we should name the object that will successively take the value of each first name
 2. Then indicate which values this object must successively take
 3. Then indicate what to do at each iteration
 - 4.

```
for (i in c("LOUIS", "DIDER", "PAULINE", "CAMILLE")) {  
  
  render(  
    input = "C:/User/Documents/prenom.Rmd",  
    output_file = "C:/User/Documents/LOUIS.html",  
    params = list(name = "LOUIS")  
  )  
}
```



2. Useful features

2.4. Report parameters

- To **avoid copy-pasting** this command for each name we want a report on, we must **use a loop**
 1. First we should name the object that will successively take the value of each first name
 2. Then indicate which values this object must successively take
 3. Then indicate what to do at each iteration
 4. And this should depend on the object that successively take each value

```
for (i in c("LOUIS", "DIDER", "PAULINE", "CAMILLE")) {  
  
  render(  
    input = "C:/User/Documents/prenom.Rmd",  
    output_file = paste0("C:/User/Documents/", i, ".html"),  
    params = list(name = i)  
  )  
}
```



Overview

1. Basic principles ✓

- 1.1. What is R Markdown?
- 1.2. YAML header
- 1.3. Code chunks
- 1.4. Text formatting
- 1.5. Run and knit your code

3. LaTeX for equations

- 3.1. What is LaTeX?
- 3.2. LaTeX syntax
- 3.3. Large equations

4. Wrap up!

2. Useful features ✓

- 2.1. Inline code
- 2.2. Tables
- 2.3. Preset themes
- 2.4. Report parameters



Overview

1. Basic principles ✓

- 1.1. What is R Markdown?
- 1.2. YAML header
- 1.3. Code chunks
- 1.4. Text formatting
- 1.5. Run and knit your code

3. LaTeX for equations

- 3.1. What is LaTeX?
- 3.2. LaTeX syntax
- 3.3. Large equations

2. Useful features ✓

- 2.1. Inline code
- 2.2. Tables
- 2.3. Preset themes
- 2.4. Report parameters



3. LaTeX for equations

3.1. What is LaTeX?

- *LaTeX* is a document preparation system
- But LaTeX is not a "*what you see is what you get*" system
 - In Microsoft Word or Google doc, you work directly on the "output document"
 - **LaTeX** works more like R Markdown: **Edit** your text **in a script using commands and symbols**
Compile the script to **get the output**
- LaTeX is the **preferred** typesetting system for most **academic** fields mainly because:
 - Many things can be **automated** in LaTeX
 - It has a good way to typeset **mathematical formulas**
- We're not gonna learn how to make *LaTeX* documents (do it in 30mn), but just how to make equations

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$$



3. LaTeX for equations

3.2. LaTeX syntax

- To include a **LaTeX equation** in R Markdown, you simply have to surround it with the **\$ sign**:

| Syntax | Output |
|-----------|--------|
| 1 + 1 | 1 + 1 |
| \$1 + 1\$ | 1 + 1 |

- LaTeX is a convenient way to display **mathematical symbols** and to **structure equations**
 - The **syntax** is mainly based on **backslashes ** and **braces {}**

Example:

- What you type in the text area: `$x \neq \frac{\alpha}{\beta} \times \beta$`
- What is rendered when knitting the document: $x \neq \frac{\alpha \times \beta}{2}$



3. LaTeX for equations

3.2. LaTeX syntax

→ Common greek letters

| Syntax | Output |
|---|------------------------|
| <code>\$\alpha\$</code> | α |
| <code>\$\beta\$</code> | β |
| <code>\$\gamma\$ \$\\Gamma\$</code> | $\gamma \Gamma$ |
| <code>\$\\delta\$ \$\\Delta\$</code> | $\delta \Delta$ |
| <code>\$\\epsilon\$ \$\\varepsilon\$</code> | $\epsilon \varepsilon$ |
| <code>\$\\lambda\$ \$\\Lambda\$</code> | $\lambda \Lambda$ |
| <code>\$\\phi\$ \$\\Phi\$</code> | $\phi \Phi$ |
| <code>\$\\pi\$ \$\\Pi\$</code> | $\pi \Pi$ |
| <code>\$\\psi\$ \$\\Psi\$</code> | $\psi \Psi$ |
| <code>\$\\theta\$ \$\\Theta\$</code> | $\theta \Theta$ |
| <code>\$\\sigma\$ \$\\Sigma\$</code> | $\sigma \Sigma$ |
| ... | ... |



3. LaTeX for equations

3.2. LaTeX syntax

→ Common symbols

| Syntax | Output |
|---|--|
| <code>\$+ - \pm\$</code> | $+$ $-$ \pm |
| <code>\$\times \div\$</code> | \times \div |
| <code>\$= \neq \equiv \approx\$</code> | $=$ \neq \equiv \approx |
| <code>\$> < \geq \leq \lessgt\$</code> | $>$ $<$ \geq \leq \lessgt |
| <code>\$\rightarrowtail \leftarrowtail \Leftrightarrowtail\$</code> | \rightarrowtail \leftarrowtail \Leftrightarrowtail |
| <code>\$\in \notin\$</code> | \in \notin |
| <code>\$\forall \exists \nexists\$</code> | \forall \exists \nexists |
| <code>\$\infty\$</code> | ∞ |
| <code>\$\sum \prod \int\$</code> | \sum \prod \int |
| ... | ... |



3. LaTeX for equations

3.2. LaTeX syntax

→ Exponents and accentuation

| Syntax | Output |
|----------------------|----------------------|
| x^a | x^a |
| x_b | x_b |
| x^{a_b} | x_b^a |
| $x^{a, i}_{b, j}$ | $x_{b,j}^{a,i}$ |
| $\hat{\beta}$ | $\widehat{\beta}$ |
| $\tilde{\beta}$ | $\widetilde{\beta}$ |
| \overline{x} | \overline{x} |
| \overrightarrow{x} | \overrightarrow{x} |
| \overleftarrow{x} | \overleftarrow{x} |
| ... | ... |



3. LaTeX for equations

3.2. LaTeX syntax

→ Math constructs and variable sized symbols

| Syntax | Output |
|---|---|
| $\frac{a \times b}{c}$ | $\frac{a \times b}{c}$ |
| $\sqrt{x} \quad \sqrt[n]{x}$ | $\sqrt{x} \quad \sqrt[n]{x}$ |
| $\sum_{i=1}^N$ | $\sum_{i=1}^N$ |
| $\prod_{i=1}^N$ | $\prod_{i=1}^N$ |
| \int_a^b | \int_a^b |
| $\overline{x} = \frac{1}{N} \sum_{i=1}^N x_i$ | $\overline{x} = \frac{1}{N} \sum_{i=1}^N x_i$ |
| ... | ... |



3. LaTeX for equations

3.3. Large equations

- Surrounding a LaTeX input with **one \$** on each side is suitable for **inline equation**
- You can also surround a LaTeX input with **two \$** on each side
 - It puts the equation at the **center of a new line**
 - And gives **more vertical space** to the equation
- Surrounding a LaTeX input with two \$ is usually good for:
 - Large equations
 - Equations that should be emphasized

The mean formula with one \$ on each side

→ For inline equations

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$$

The mean formula with two \$ on each side

→ For large/emphasized equations

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$$



3. LaTeX for equations

3.3. Large equations

- Sometimes you do not want two **consecutive lines** of equations to be centered
 - You may want to **align** them based on a **common part** within the equations
- This should be done in an **aligned environment** (`\begin{aligned} ... \end{aligned}`)
 - Place the "**&**" symbol where the equations should be aligned
 - And break a line using "**\\"**"

```
$$
\begin{aligned}
x &= (a + b) \times c \\
&= (a \times c) + (b \times c)
\end{aligned}
$$
```

$$\begin{aligned} x &= (a + b) \times c \\ &= (a \times c) + (b \times c) \end{aligned}$$



3. LaTeX for equations

3.3. Large equations

- The same principle applies within **cases environment**

```
 $$\text{Med}(x) = \begin{cases} x[\frac{N+1}{2}] & \text{if } N \text{ is odd} \\ \frac{x[\frac{N}{2}]+x[\frac{N}{2}+1]}{2} & \text{if } N \text{ is even} \end{cases}
```

$$\text{Med}(x) = \begin{cases} x[\frac{N+1}{2}] & \text{if } N \text{ is odd} \\ \frac{x[\frac{N}{2}]+x[\frac{N}{2}+1]}{2} & \text{if } N \text{ is even} \end{cases}$$

- Note that the **text function** allows to write text without it being interpreted as mathematical letters:

```
 $$\text{Mean}(x)=\frac{1}{N}\sum_{i=1}^N x_i$$
```

```
 $$\text{Mean}(x)=\frac{1}{N}\sum_{i=1}^N x_i$$
```

$$\text{Mean}(x) = \frac{1}{N} \sum_{i=1}^N x_i$$

$$\text{Mean}(x) = \frac{1}{N} \sum_{i=1}^N x_i$$

Practice

03 : 00

1) Inside your .Rmd, reproduce the following mathematical expression

$$Y_i = \alpha + \beta X_i + \varepsilon_i$$

2) Then reproduce the following sentence

\hat{Y}_i denote the fitted values of the model.

You've got 3 minutes!

Solution

1) Inside your .Rmd, reproduce the following mathematical expression

$$Y_i = \alpha + \beta X_i + \varepsilon_i$$

```
$$Y_i = \alpha + \beta X_i + \varepsilon_i$$
```

2) Then reproduce the following sentence

\hat{Y}_i denote the fitted values of the model.

```
$\hat{Y}_i$ denote the fitted values of the model.
```



Overview

1. Basic principles ✓

- 1.1. What is R Markdown?
- 1.2. YAML header
- 1.3. Code chunks
- 1.4. Text formatting
- 1.5. Run and knit your code

3. LaTeX for equations ✓

- 3.1. What is LaTeX?
- 3.2. LaTeX syntax
- 3.3. Large equations

4. Wrap up!

2. Useful features ✓

- 2.1. Inline code
- 2.2. Tables
- 2.3. Preset themes
- 2.4. Report parameters



4. Wrap up!

1. Three types of contents

YAML header →

```
1 ---  
2 title: "Report example"  
3 author: "Louis Sirugue"  
4 date: "26/09/2021"  
5 output: html_document  
6 ---  
7  
8 ## Overview of the data  
9  
10 ```{r cars}  
11 # Omit if distance >= 100  
12 cars <- cars[cars$dist < 100, ]  
13 names(cars)  
14 dim(cars)  
15 c(mean(cars$speed), mean(cars$dist))  
16  
17  
18 The dataset we consider contains two variables, speed and distance, and has `r dim(cars)[1]` observations. The average speed value is `r mean(cars$speed)` and the average distance value is `r mean(cars$dist)`.
```

Code chunks →

Text →

Report example

Louis Sirugue

26/09/2021

Overview of the data

```
# Omit if distance >= 100  
cars <- cars[cars$dist < 100, ]  
names(cars)
```

```
## [1] "speed" "dist"
```

```
dim(cars)
```

```
## [1] 49  2
```

```
c(mean(cars$speed), mean(cars$dist))
```

```
## [1] 15.22449 41.40816
```

The dataset we consider contains two variables, speed and distance, and has 49 observations. The average speed value is 15.2244898 and the average distance value is 41.4081633.



4. Wrap up!

2. Useful features

→ **Inline code** allows to include the output of some **R code within text areas** of your report

Syntax

```
`paste("a", "b", sep = "-")`
```

```
`r paste("a", "b", sep = "-")`
```

Output

```
paste("a", "b", sep = "-")
```

a-b

→ **kable()** for clean **html tables** and **datatable()** to navigate in **large tables**

```
kable(results_table)  
datatable(results_table)
```



4. Wrap up!

3. LaTeX for equations

- *LATEX* is a convenient way to display **mathematical** symbols and to structure **equations**
 - The **syntax** is mainly based on **backslashes ** and **braces {}**

→ What you **type** in the text area: `$x \neq \frac{\alpha \times \beta}{2}$`

→ What is **rendered** when knitting the document: $x \neq \frac{\alpha \times \beta}{2}$

To **include** a **LaTeX equation** in R Markdown, you simply have to surround it with the **\$ sign**

The mean formula with one \$ on each side

→ For inline equations

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$$

The mean formula with two \$ on each side

→ For large/emphasized equations

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$$