R Markdown & LaTeX

Lecture 5

Louis SIRUGUE

CPES 2 - Fall 2022

Last time we saw

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()
```

Last time we saw

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()

Last time we saw

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 every **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)
```

Today we learn how use 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

2. Useful features

- 2.1. Inline code
- 2.2. Tables
- 2.3. Preset themes

3. LaTeX for equations

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

4. Wrap up!

5. Guidelines for the homework

Today we learn how use 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
  - Last year homework
  - Example of research project
  - Supplementary material
  - Course webpage and material

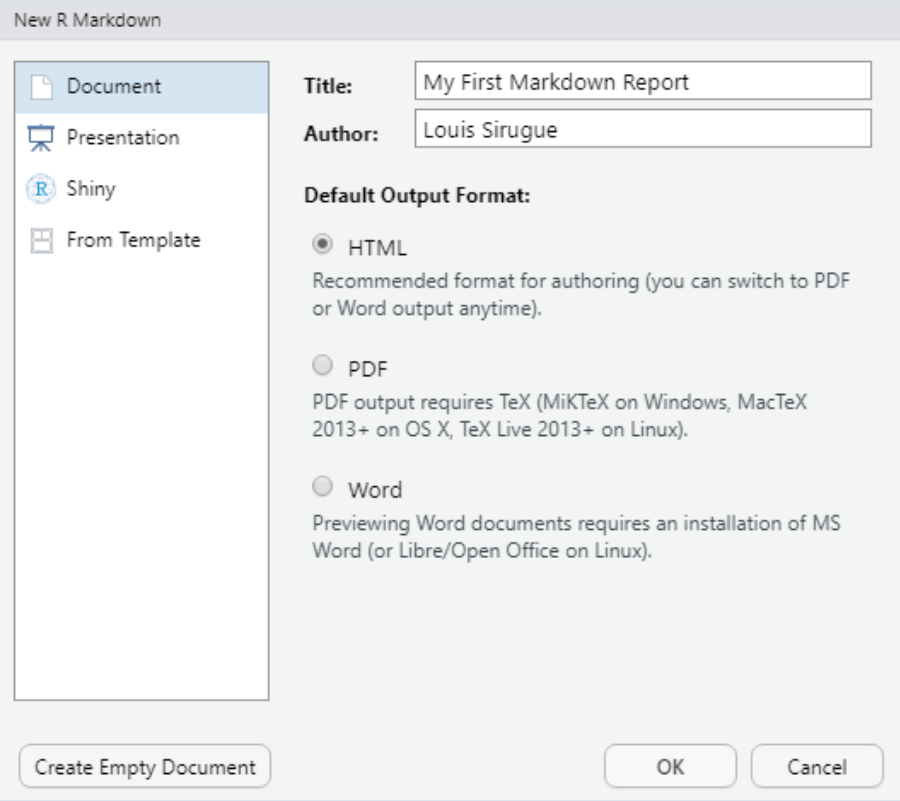
- It is structured around **3 types of content**:
  - **Code chunks** to run and render the output
  - **Editable text** to display
  - **YAML metadata** for the R Markdown build process

➜ 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 up 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!

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

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 quote
  - 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	Output
`Plain text   End a line with two spaces for line break`	Plain text End a line with two spaces for line break
`*italics*`	*italics*
`**bold**`	**bold**
`# Header 1`	# Header 1
`## Header 2`	## Header 2
...	...
`###### Header 6`	###### Header 6
`[link](https://www.rstudio.com)`	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

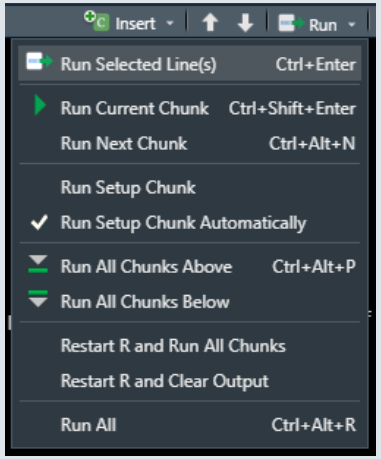
1. ordered list
2. item 2
   - sub-item 1
   - sub-item 2

1. Basic principles

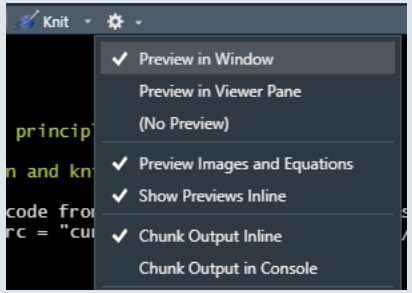
1.5. Run and knit your code

- 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** Knit

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

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

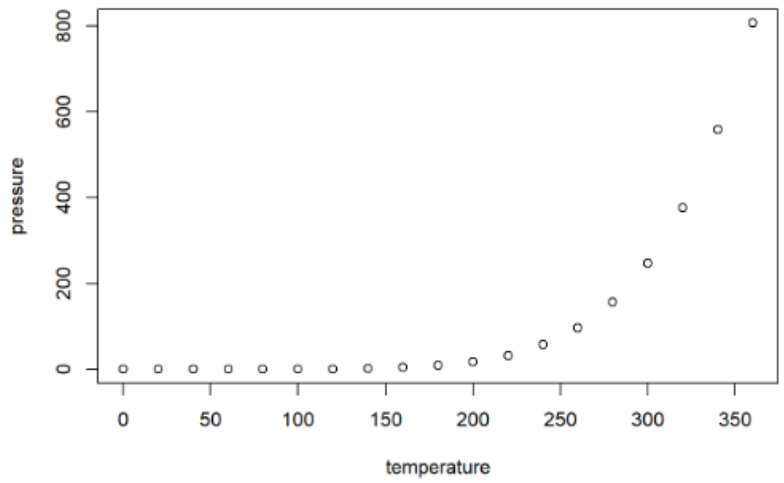
1.5. Run and knit your code

- To **render** an R Markdown file, click on the **knit button** 


```
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
```

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

2. Useful features

- 2.1. Inline code
- 2.2. Tables
- 2.3. Preset themes

3. LaTeX for equations

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

4. Wrap up!

5. Guidelines for the homework

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. 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

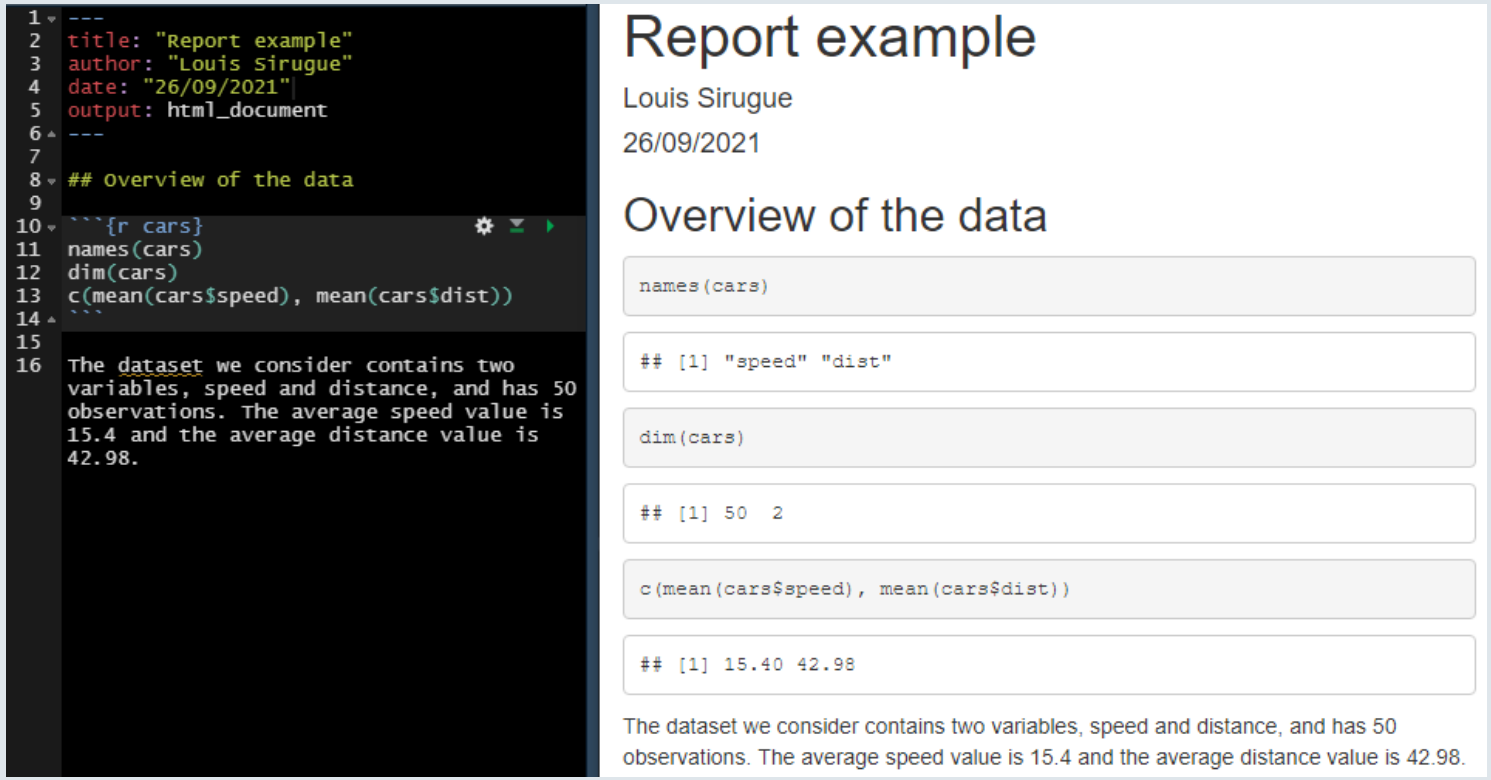
- Imagine that you forgot to filter an irrelevant group of observations at the beginning
  - If you simply filter your data at the beginning in a code chunk
  - All your tables and figures will **update automatically**

- 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

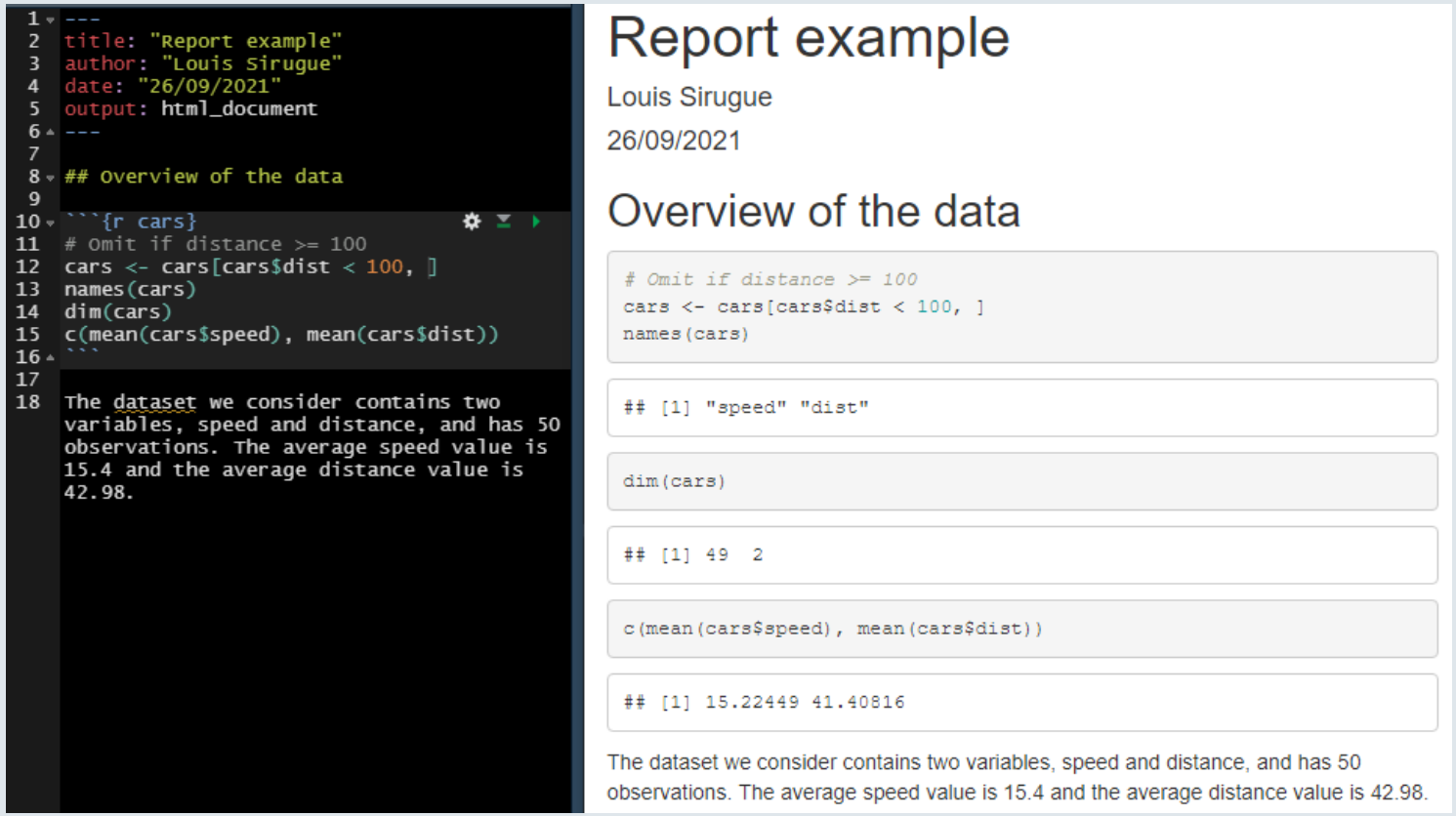
- Consider the following report :



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



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

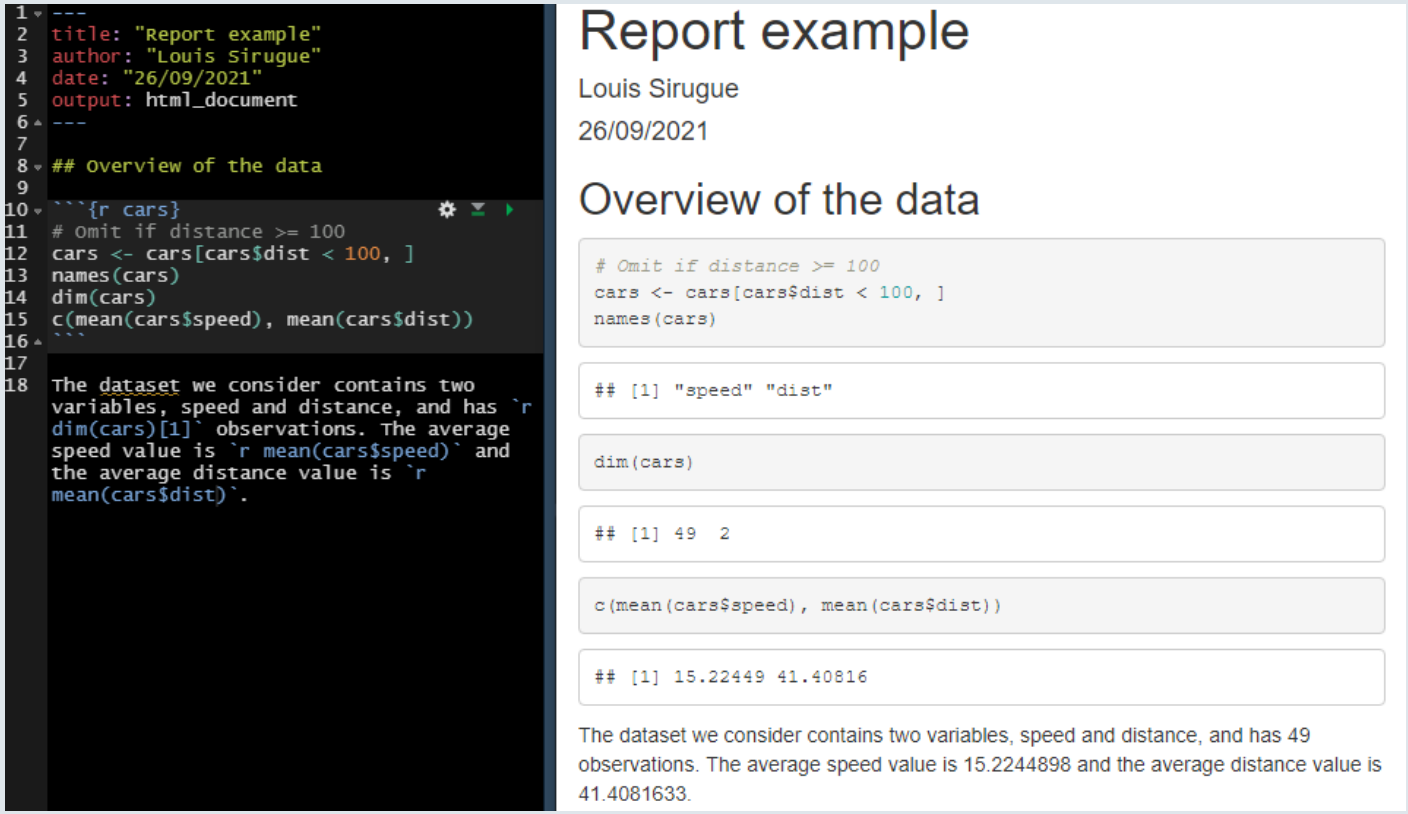
- R code outside code chunks should be included between backticks:
  - Surrounding code with **backticks** in a text area will **change** the **format** to that of the code chunk
  - **Adding** the `r` letter right after the first backtick will **show** the **output** of the code instead of the code

Syntax	Output
`` `paste("a", "b", sep = "-")` ``	`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**:



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")
```

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

```
install.packages("DT") # if not already installed
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

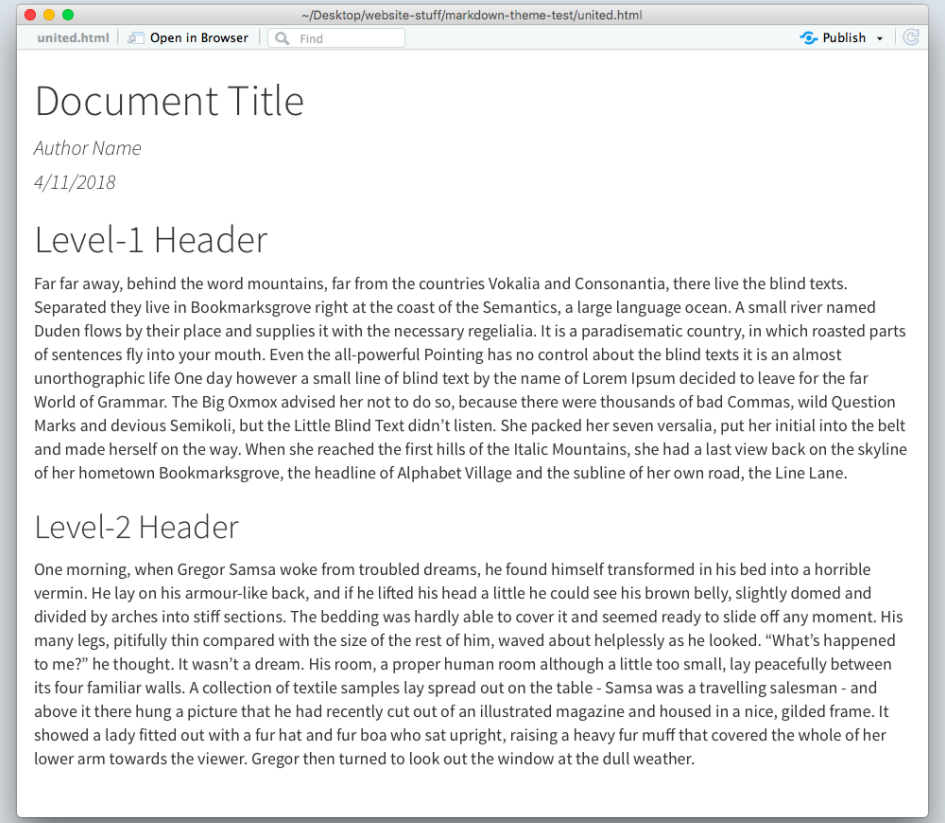
```
---
title: "My First Markdown Report"
author: "Louis Sirugue"
date: "24/09/2021"
output:
  html_document:
    theme: cosmo
---
```

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

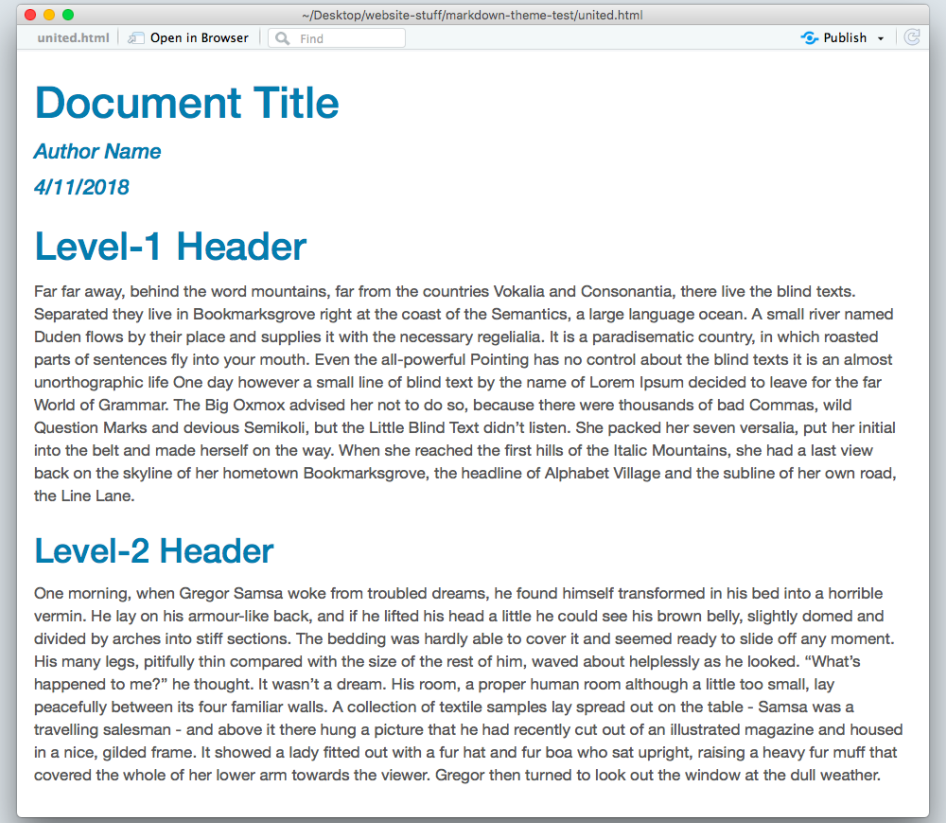
2. Useful features

2.3. Preset themes

cosmo



cerulean



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 apprearance 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 apprearance 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)

Creating Pretty Documents From R Markdown

THE LEONIDS 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 apprearance of themes.

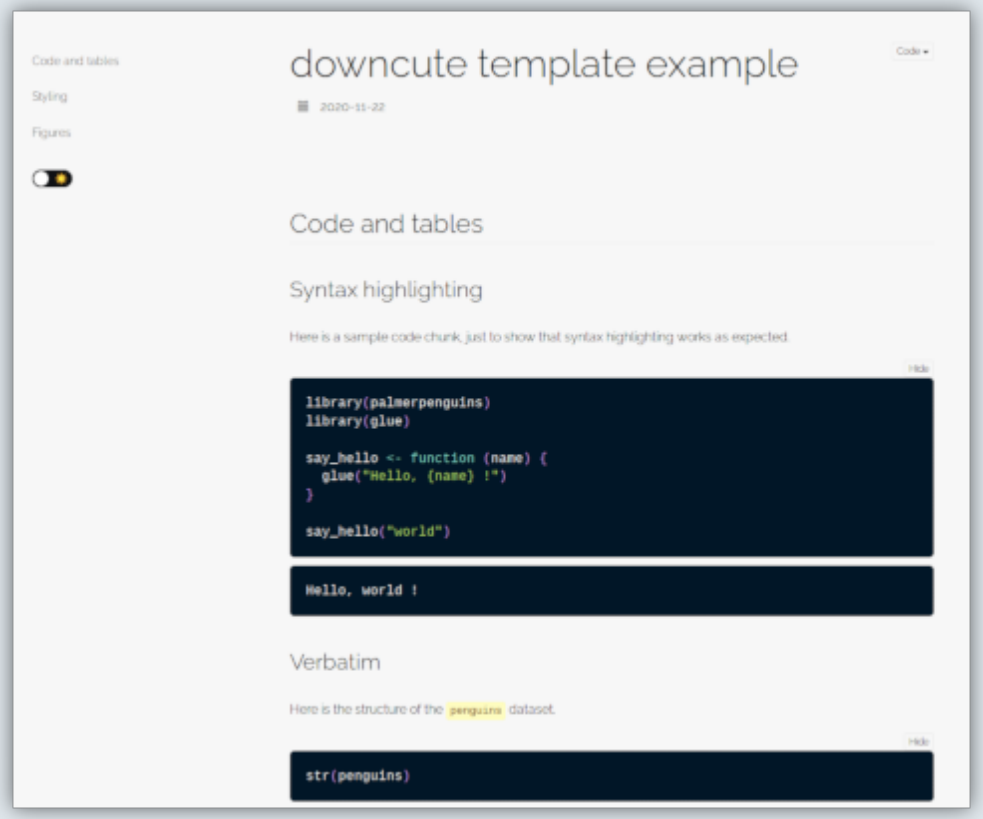
Tables

	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			

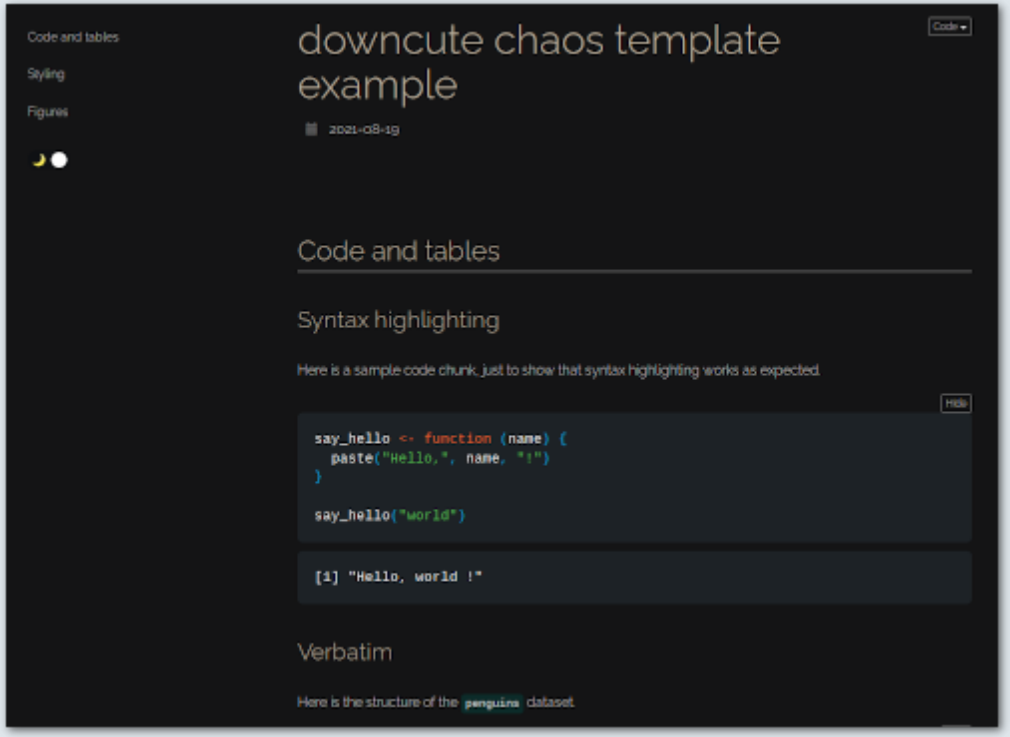
2. Useful features

2.3. Preset themes

downcute (from rmdformats)



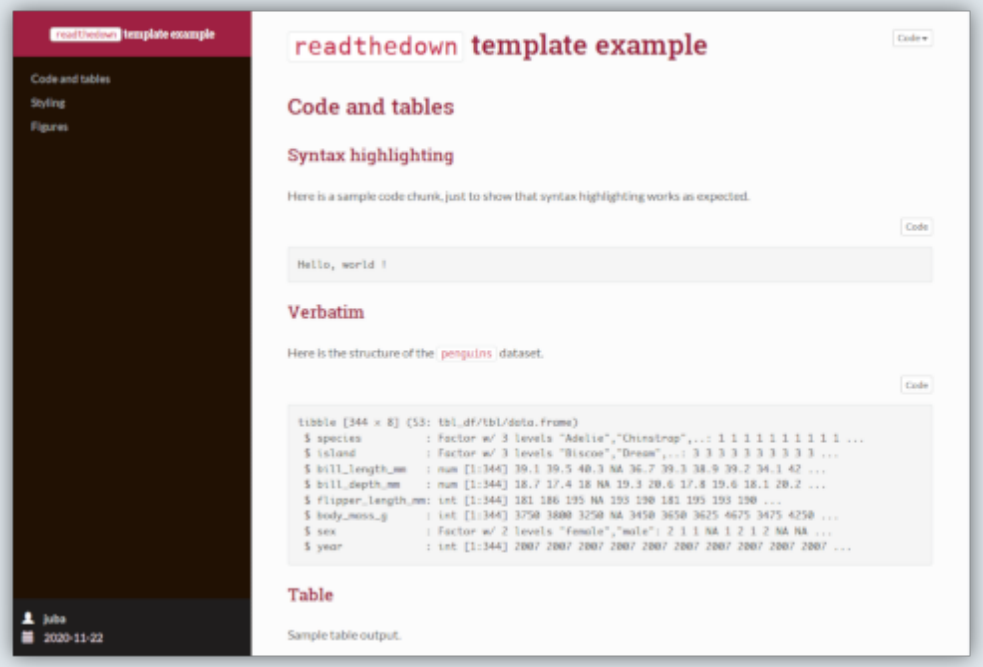
downcute chaos (from rmdformats)



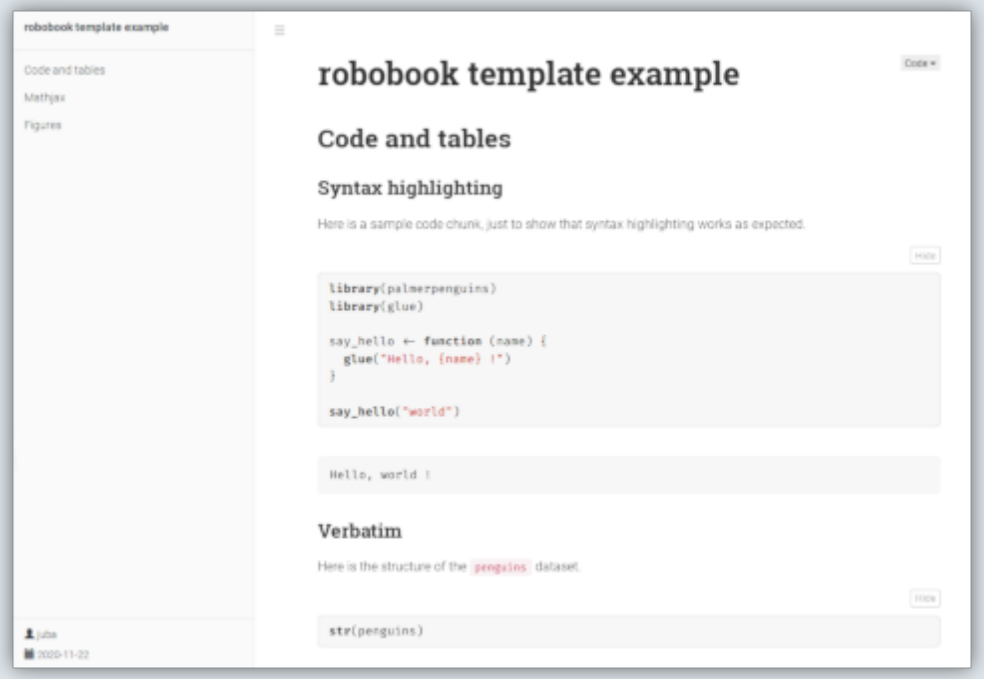
2. Useful features

2.3. Preset themes

readthedown (from rmdformats)



robobook (from rmdformats)



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

3. LaTeX for equations

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

4. Wrap up!

5. Guidelines for the homework

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

3. LaTeX for equations

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

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, but just how to make equations

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

3. LaTeX for equations

3.1. What is LaTeX?

- 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: `$x \neq \frac{\alpha \times \beta}{2}$`

➜ 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
`$\alpha$`	α
`$\beta$`	β
`$\gamma$` `$\Gamma$`	γ Γ
`$\delta$` `$\Delta$`	δ Δ
`$\epsilon$` `$\varepsilon$`	ϵ ε
`$\lambda` `\Lambda$`	λ Λ
`$\phi$` `$\Phi$`	ϕ Φ
`$\pi$` `$\Pi$`	π Π
`$\psi$` `$\Psi$`	ψ Ψ
`$\theta$` `$\Theta$`	θ Θ
`$\sigma$` `$\Sigma$`	σ Σ
...	...

3. LaTeX for equations

3.2. LaTeX syntax

→ Common symbols

Syntax	Output
`$+ - \pm$`	$+ - \pm$
`$\times \div$`	$\times \div$
`$= \neq \equiv \approx$`	$= \neq \equiv \approx$
`$> < \geq \leq \lessgtr$`	$> < \geq \leq \lessgtr$
`$\rightarrow \leftarrow \Leftrightarrow$`	$\rightarrow \leftarrow \Leftrightarrow$
`$\in \notin$`	$\in \notin$
`$\forall \exists \nexists$`	$\forall \exists \nexists$
`$\infty$`	∞
`$\sum \prod \int$`	$\sum \prod \int$
...	...

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^a_b
`$x^{a, i}_{b, j}$`	$x^{a,i}_{b,j}$
`$\hat{\beta} \widehat{\beta_{i,j}}$`	$\hat{\beta}\ \widehat{\beta_{i,j}}$
`$\tilde{\beta} \widetilde{\beta_{i,j}}$`	$\tilde{\beta}\ \widetilde{\beta_{i,j}}$
`$\overline{x} \underline{x}$`	$\overline{x}\ \underline{x}$
`$\overrightarrow{x} \underleftarrow{x}$`	$\overrightarrow{x}\ \underleftarrow{x}$
...	...

3. LaTeX for equations

3.2. LaTeX syntax

→ Math constructs and variable sized symbols

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

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 $**
  - It puts the equation at the **center of a new line**
  - And gives **more vertical space** to the equation

- To surround 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

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

The mean formula with two `$` on each side

➜ For large/emphasized equations

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

3. LaTeX for equations

3.3. Large equations

- Sometimes you do not want to **consecutive lines** of equations to be centered
  - You may want to **align** them based on **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:

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

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

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

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

Practice

Reproduce the following html output using R markdown

You've got 15 minutes!

Lecture 5 - Practice

Your name

03/10/2022

1. Setup

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

```
library(tidyverse)
```

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

2. Computations

The `rnorm(n, mean, sd)` command allows to generate n observations drawn from a normal distribution with a given mean and standard deviation.

```
x <- rnorm(1000, 0, 1)
```

We can compute the mean $\bar{x}$ of this variable:

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

```
mean(x)
```

```
## [1] 0.002126865
```

Here is the result, written in **inline code** such that it updates automatically: 0.0021269

Solution

```
---
title: "Lecture 5 - Practice"
author: "Your name"
date: "03/10/2022"
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(tidyverse)
```

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

Solution

```
### 2. Computations

The `rnorm(n, mean, sd)` command allows to generate n observations drawn from a normal distribution
with a given mean and standard deviation.

```{r}
x <- rnorm(1000, 0, 1)
```

We can compute the mean $\overline{x}$ of this variable:

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

```{r}
mean(x)
```

Here is the result, written in **inline code** such that it updates automatically: `r mean(x)`
```

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

3. LaTeX for equations ✓

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

4. Wrap up!

5. Guidelines for the homework

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

3. LaTeX for equations ✓

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

4. Wrap up!

4. Wrap up!

1. Three types of contents

YAML header →

Code chunks →

Text →

```
---
title: "Report example"
author: "Louis Sirugue"
date: "26/09/2021"
output: html_document
---

## Overview of the data

```{r cars}
# Omit if distance >= 100
cars <- cars[cars$dist < 100, ]
names(cars)
dim(cars)
c(mean(cars$speed), mean(cars$dist))
```

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.

4. Wrap up!

2. Useful features

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

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

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

The mean formula with two `$` on each side

➜ For large/emphasized equations

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

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

2. Useful features ✓

- 2.1. Inline code
- 2.2. Tables
- 2.3. Preset themes

3. LaTeX for equations ✓

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

4. Wrap up! ✓

5. Guidelines for the homework

5. Guidelines for the homework

- The homework should be done with R Markdown
  - Both the .Rmd and the corresponding .html should be sent
  - Deadline: October 10

- You are expected to use (when relevant) the features covered today. It will be taken into account in the grading (see the **indicative** grading system here)
  - Inline code
  - kable
  - LaTeX equations
  - ...

- Your Markdown report should run without error
  - If you use absolute paths that I need to change, define it once at the very beginning and use paste0()

```
path <- "C:/Users/name/Desktop/homework/"
data1 <- read.csv(paste0(path, "file1.csv"))

...

data2 <- read.csv(paste0(path, "file2.csv"))
```