

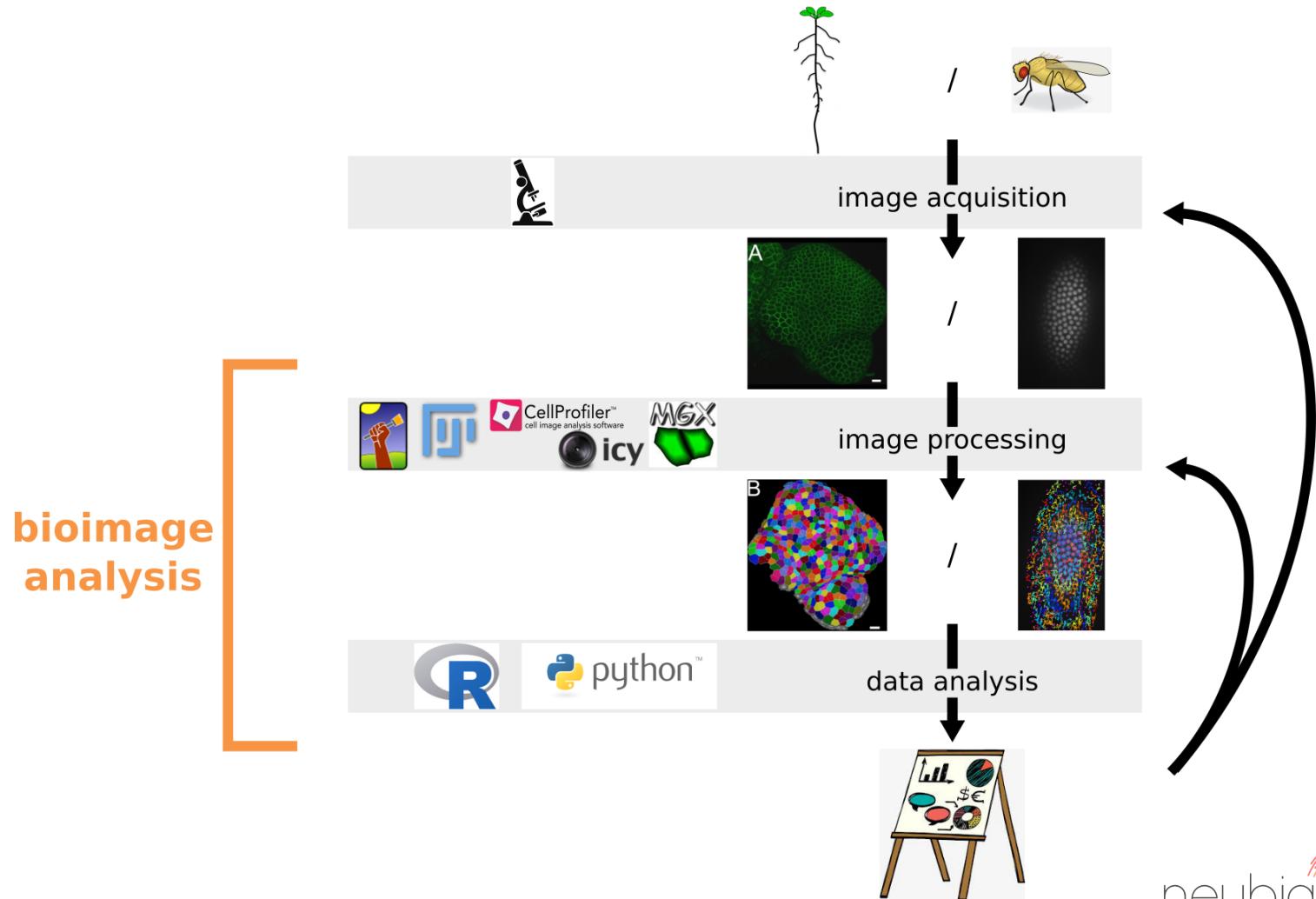
NEUBIAS - TS10

# Introduction to R for bioimage analysis

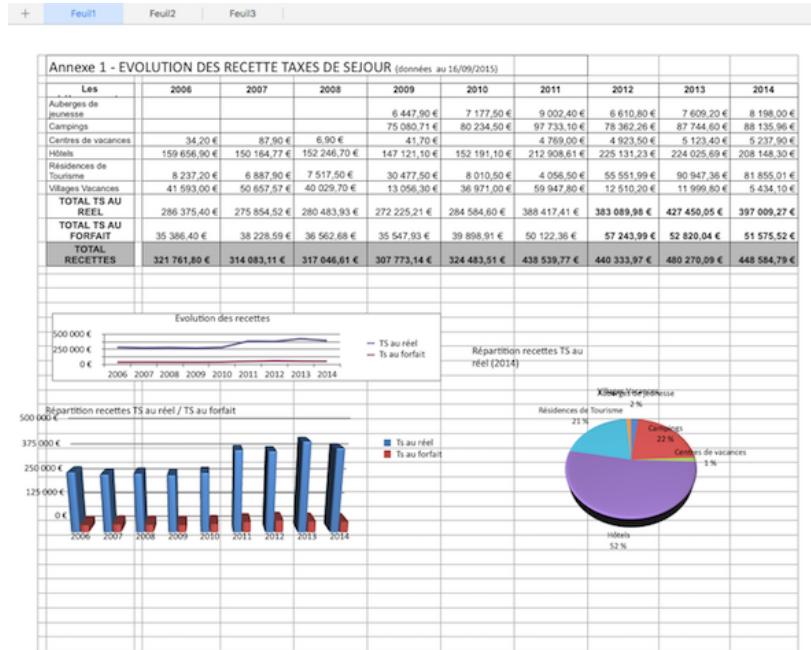
Marion Louveaux

2019-01-26

# Bioimage analysis includes data analysis

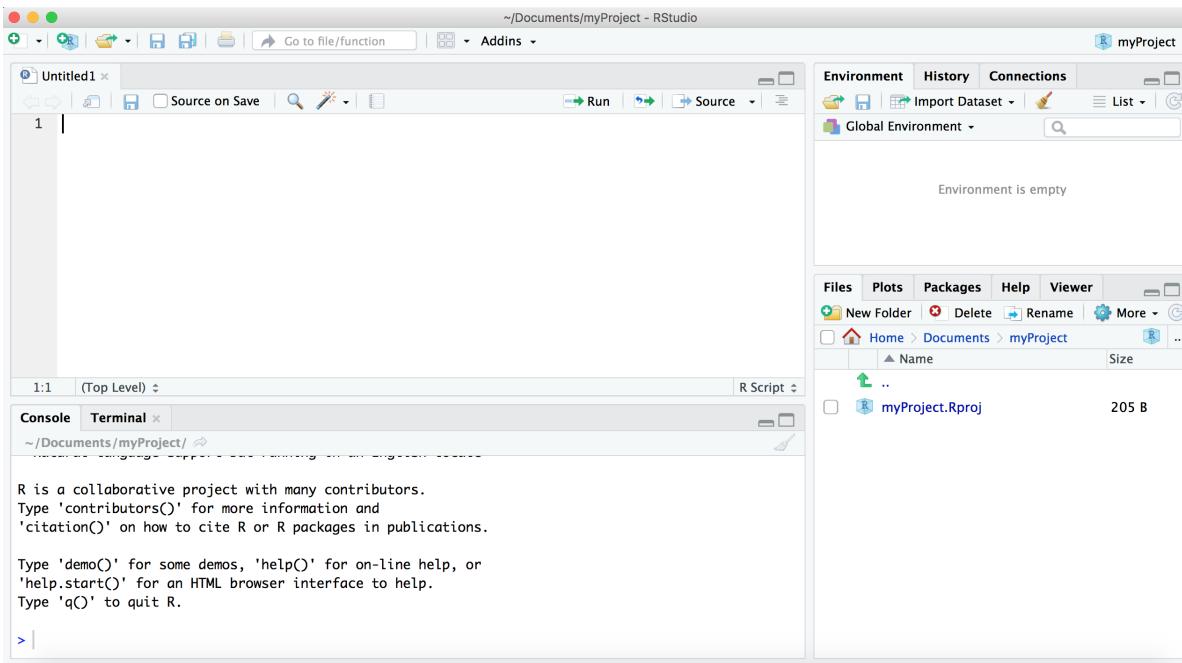


# Why using a programming language to perform data analysis?



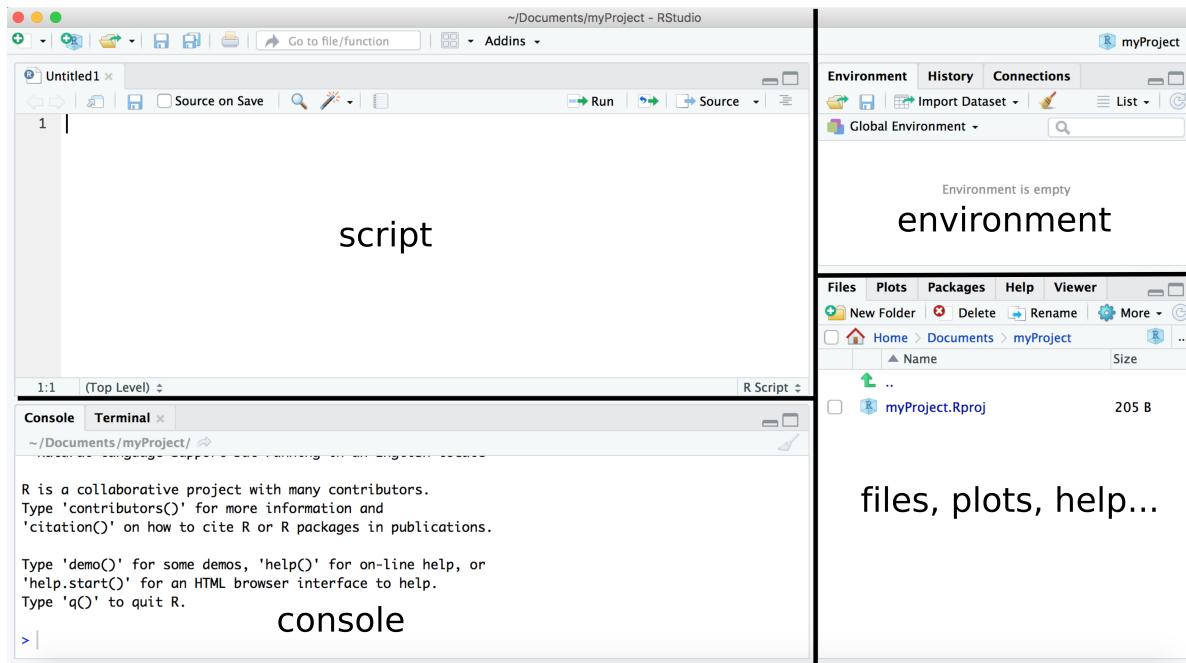
- get reproducible analyses
- share with others
- re-use later
- keep raw data raw

# Get Started: R studio and .Rproj



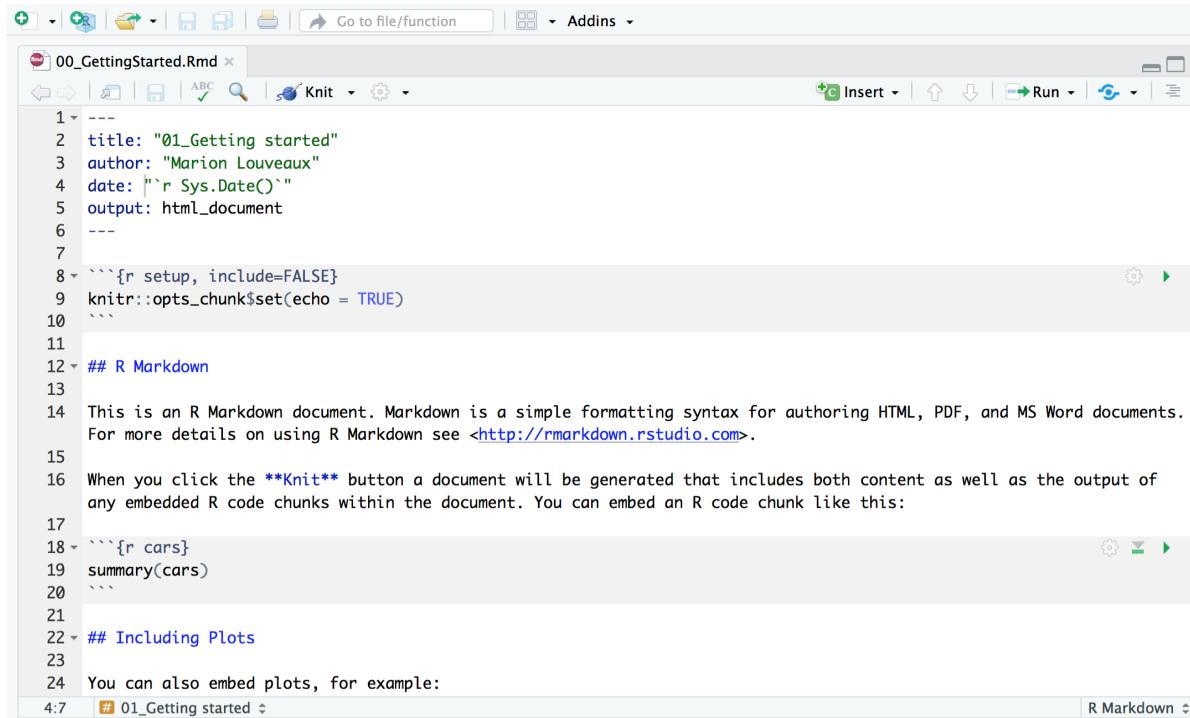
Double clic on TS10\_Rintro.Rproj

# Get Started: R studio and .Rproj



Double clic on TS10\_Rintro.Rproj

# 00\_Getting started: R markdown



```
1: ---  
2: title: "01_Getting started"  
3: author: "Marion Louveaux"  
4: date: |`r Sys.Date()`|  
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:  
4:7 # 01_Getting started
```

Bottom right panel of Rstudio: Files -> scripts  
Click on 00\_GettingStarted.Rmd

# 00\_Getting started: R markdown

```
1 -> ----  
2 <title: "01_Getting started"  
3 <author: "Marion Louveaux"  
4 <date: |"r Sys.Date()"  
5 <output: html_document  
6 ---  
7  
8 `r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10 ...  
11  
12 ## R Markdown ← Paragraph title  
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:  
4:7 # 01_Getting started
```

Header - Do not modify!

Text - Add your notes

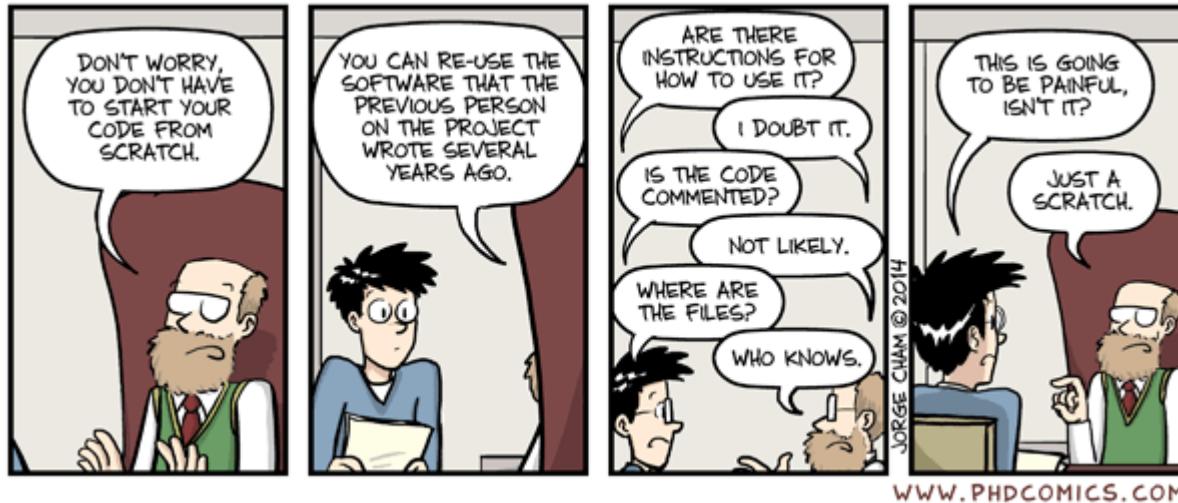
Paragraph title

Chunk - Add your code

Bottom right panel of Rstudio: Files -> scripts  
Click on 00\_GettingStarted.Rmd

# Why using an R project and Rmarkdown

We aim at avoiding this situation:



<http://phdcomics.com/comics/archive.php?comicid=1689>

# 01\_Rbasics

## Execute a command

ctrl + enter to execute a command.

NB: cmd + enter on Mac.

## Assign a value to a variable

```
my_age <- 29
```

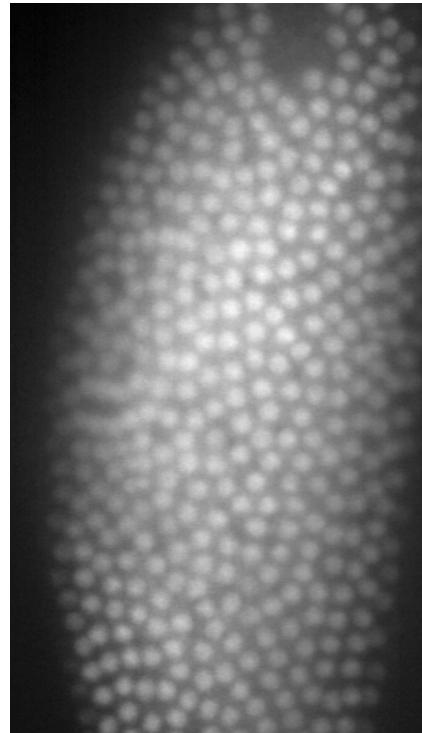
my\_age is a **variable**. We **assign** 29 to my\_age.

Bottom right panel of Rstudio: Files -> scripts  
Click on 00\_GettingStarted.Rmd

# 02\_Exercise1

We'll work on the data from the tracking session:

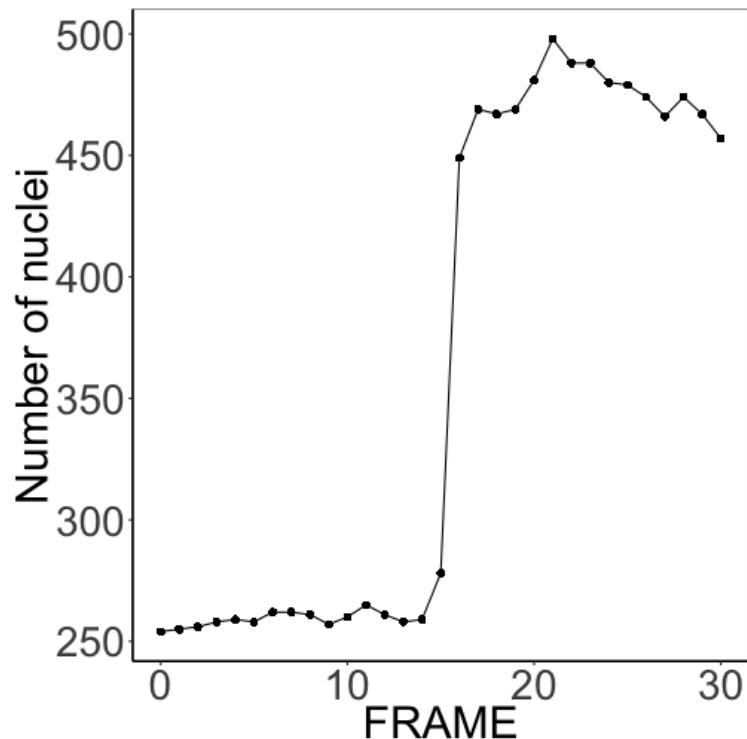
- developing Drosophila embryo, carrying a histone-RFP marker
- timelapse (31 frames, frame delay = 30 seconds)
- pixel size  $0.52 \times 0.52 \times 2.5 \mu\text{m}^3$



| All data are available in the .Rproj folder "/datasets/fly\_embryo".

# 02\_Exercise1: Goals

To be able to produce a plot like this one...



# 02\_Exercise1: Goals

... using only a few lines of code

```
df_fly_spots <- read_csv(here(  
  "datasets/fly_embryo/division_timelapse_spot_stats.csv"  
)  
  
nb_nuclei <- df_fly_spots %>%  
  group_by(FRAME) %>%  
  summarise(nb = n())  
  
df_fly_spots2 <- left_join(df_fly_spots, nb_nuclei)  
  
ggplot(df_fly_spots2, aes(FRAME, nb)) +  
  geom_point() +  
  geom_line() +  
  xlab("FRAME") +  
  ylab("Number of nuclei") +  
  custom_plot_theme()
```

# 02\_Exercise1: librairies

R uses additionnal librairies, that must be loaded for each script using `library()`.

```
library(readr)
```

Here, we will mainly work with

- `readr` (to read csv)
- `dplyr` (to wrangle data)
- `magrittr` (`%>%` symbol)
- and `ggplot2` (to plot)

# 02\_Exercise1: read and view a dataframe

To read a .csv file, we will use the `read_csv` function from `readr`. This function reads the content of the csv and gives an object called "tibble"" (= a sort of dataframe) as an output.

```
dataframe <- read_csv("path_to_my_csv.csv")
```

We can visualise the content of the dataframe using `head()` or `View()`.

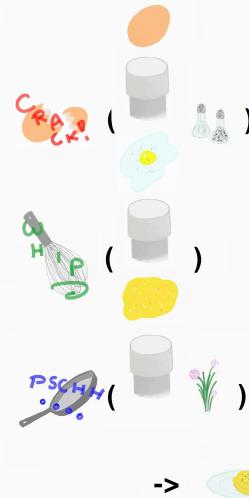
```
dataframe %>%  
  head()  
  
# To see the whole table (be careful with big dataframes!!)  
dataframe %>%  
  View()
```

# 02\_Exercise1: the pipe symbol %>%

The pipe symbol is used **to chain instructions**, as you would do for a cooking recipe:

```
omelette_with_chives <- egg %>%  
  crack(add_seasoning) %>%  
  beat() %>%  
  cook(add_chives)
```

Piped command line



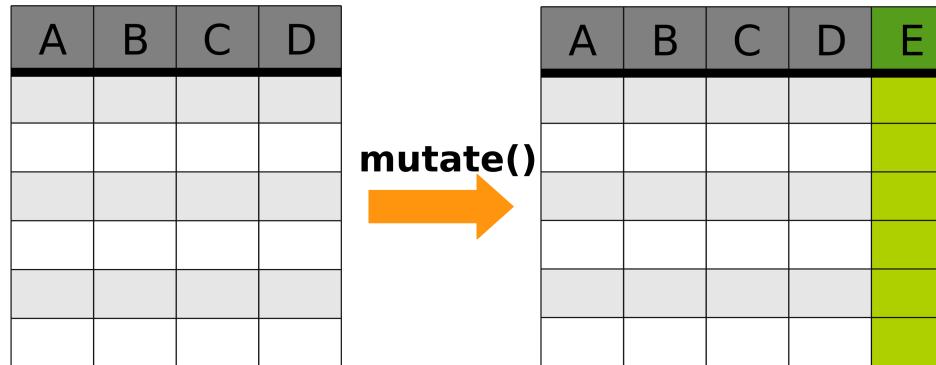
@LVaudor

From: <http://perso.ens-lyon.fr/lise.vaudor/utiliser-des-pipes-pour-enchaîner-des-instructions/>

# 02\_Exercise1: mutate()

mutate() = to add a new column

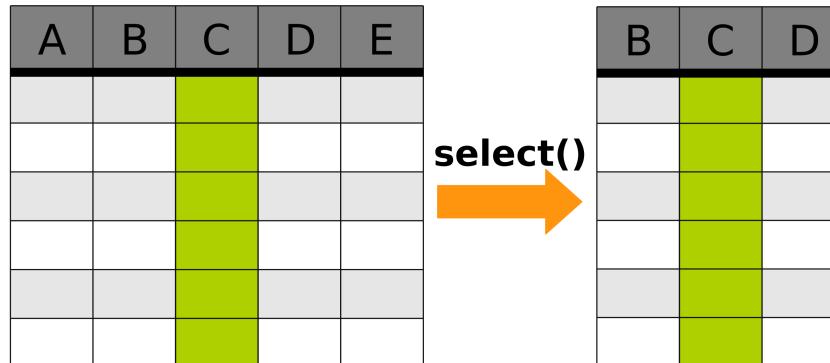
```
result <- data %>%  
  mutate(new_column_name = expression)
```



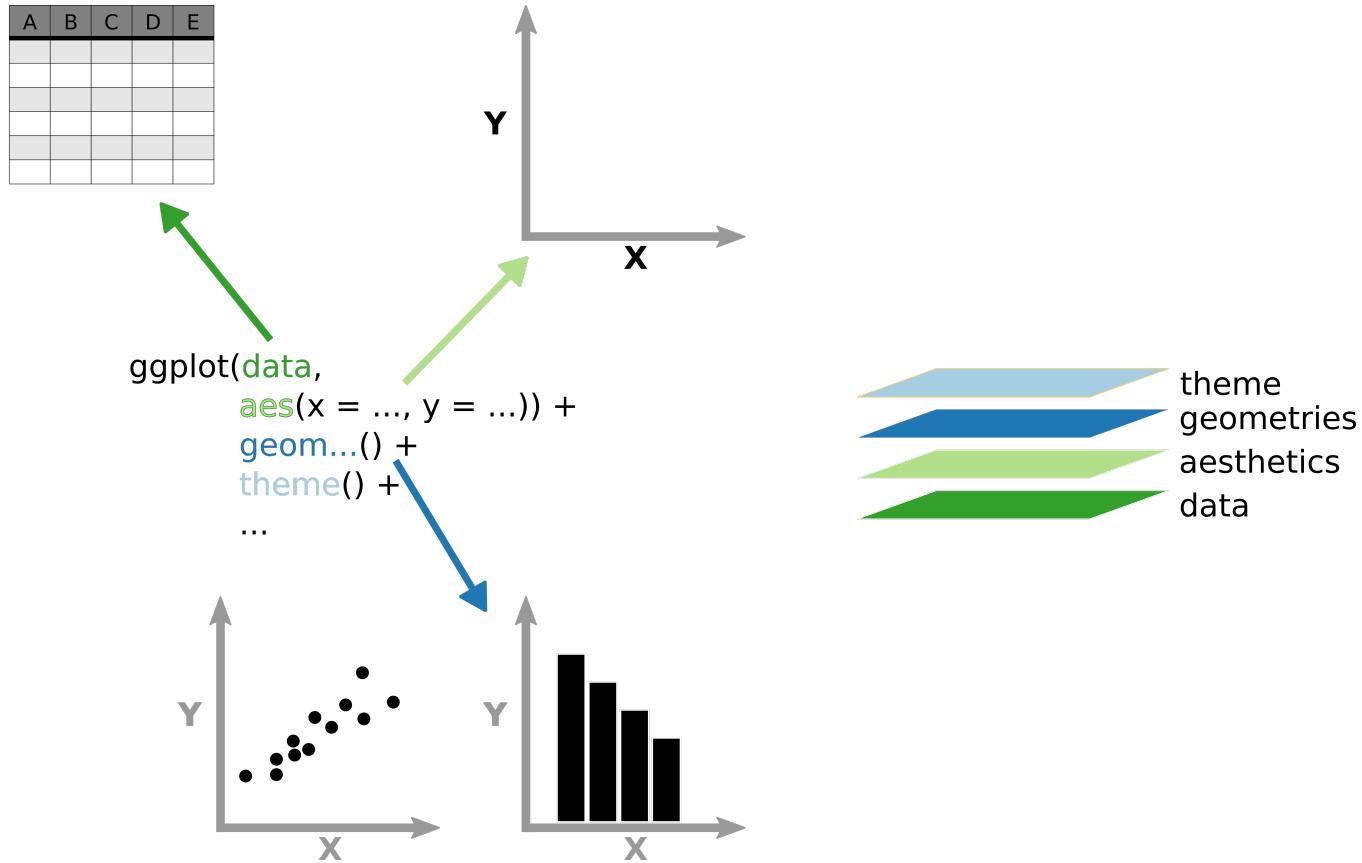
# 02\_Exercise1: select()

select() = to select or exclude **columns** in the dataframe

```
result <- data %>%  
  select(column1_to_keep, column2_to_keep, column3_to_keep)  
  
result <- data %>%  
  select(-column_to_exclude)
```



# 02\_Exercise1: Plots with ggplot2

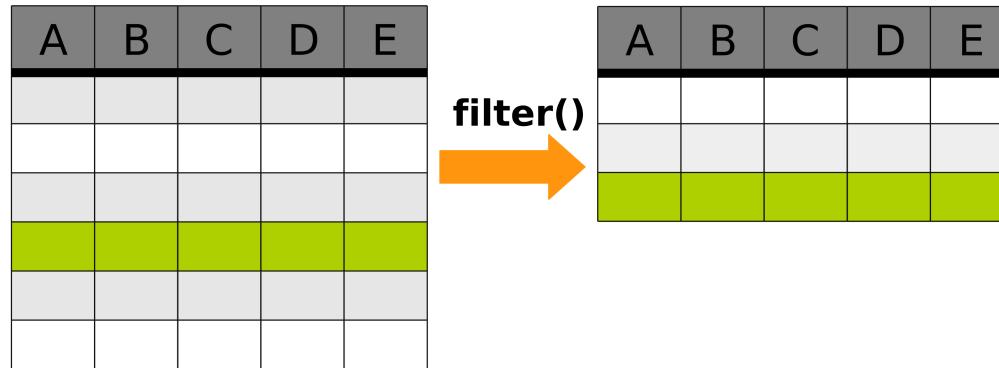


! Do not mix %>% and + (specific to plots) !

# 02\_Exercise1: filter()

filter() = to filter **rows** on one or several conditions

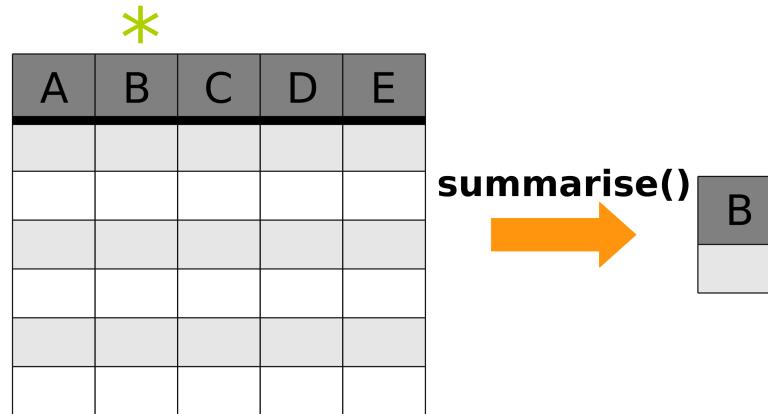
```
result <- data %>%  
  filter(condition1)  
  
result <- data %>%  
  filter(condition1 & condition2) # AND  
  
result <- data %>%  
  filter(condition1 | condition2) # OR
```



# 02\_Exercise1: summarise()

summarise() = to **resume** a column into one value (mean, median, sum...)

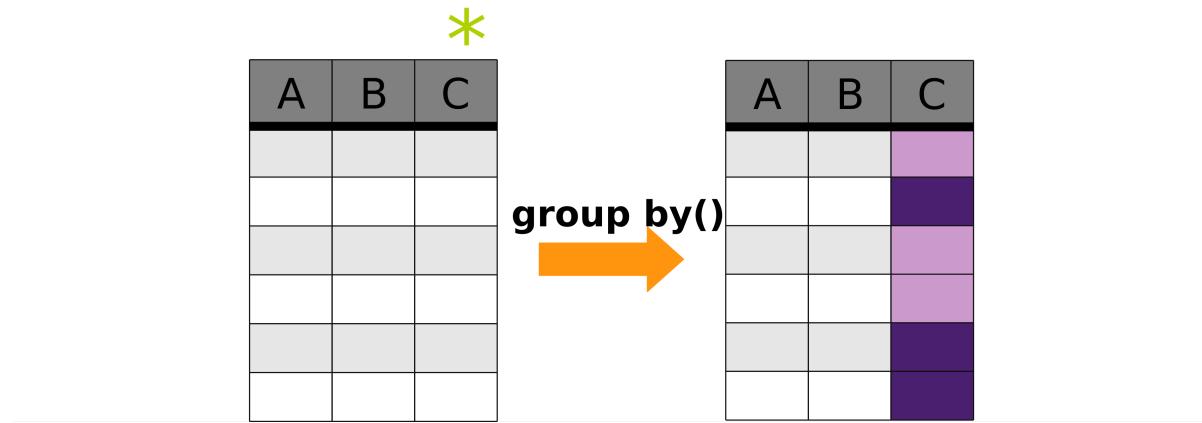
```
result <- data %>%  
  summarise(new_column_name = function(column_of_interest))
```



# 02\_Exercise1: group\_by()

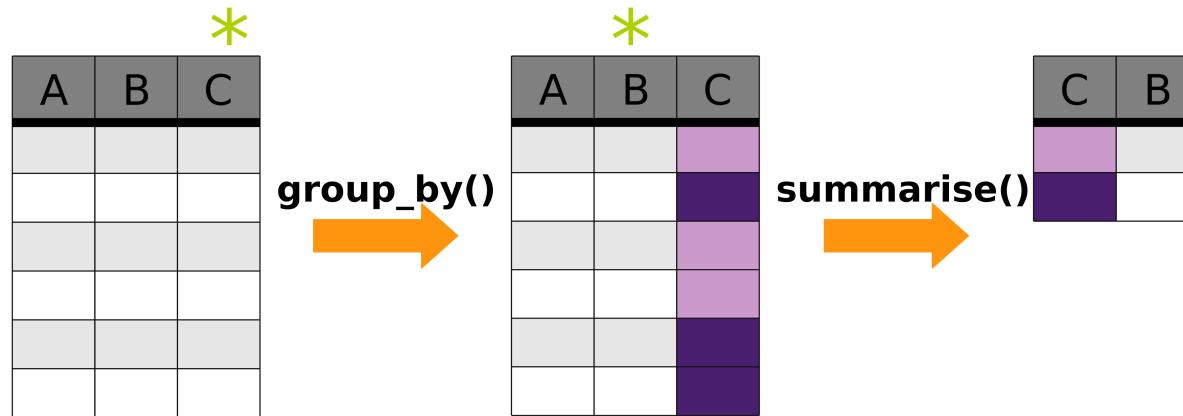
group\_by() = to group the rows according to a categorical variable.

```
result <- data %>%  
  group_by(categorical_variable_of_interest)
```



# 02\_Exercise1: group\_by() and summarise()

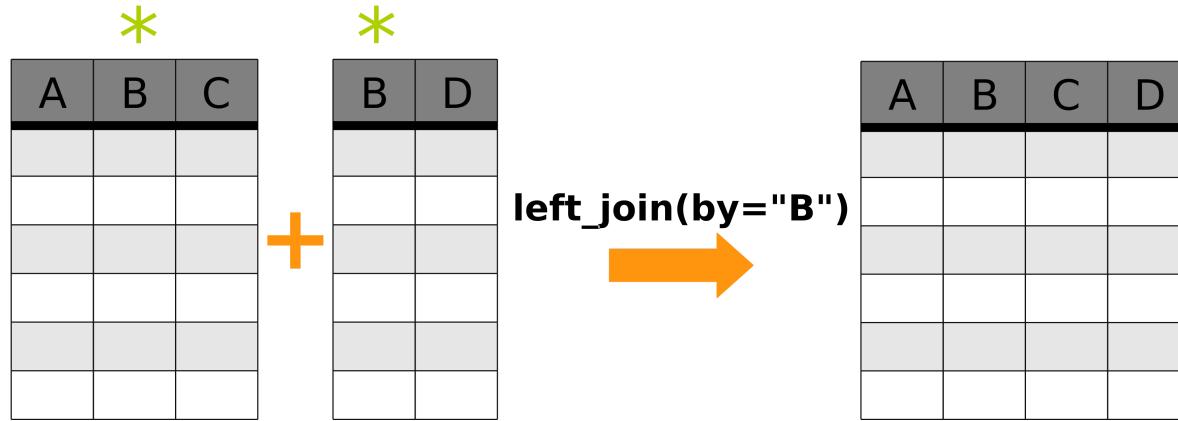
```
result <- data %>%  
  group_by(categorical_variable_of_interest)  
  summarise(new_column_name = function(column_of_interest))
```



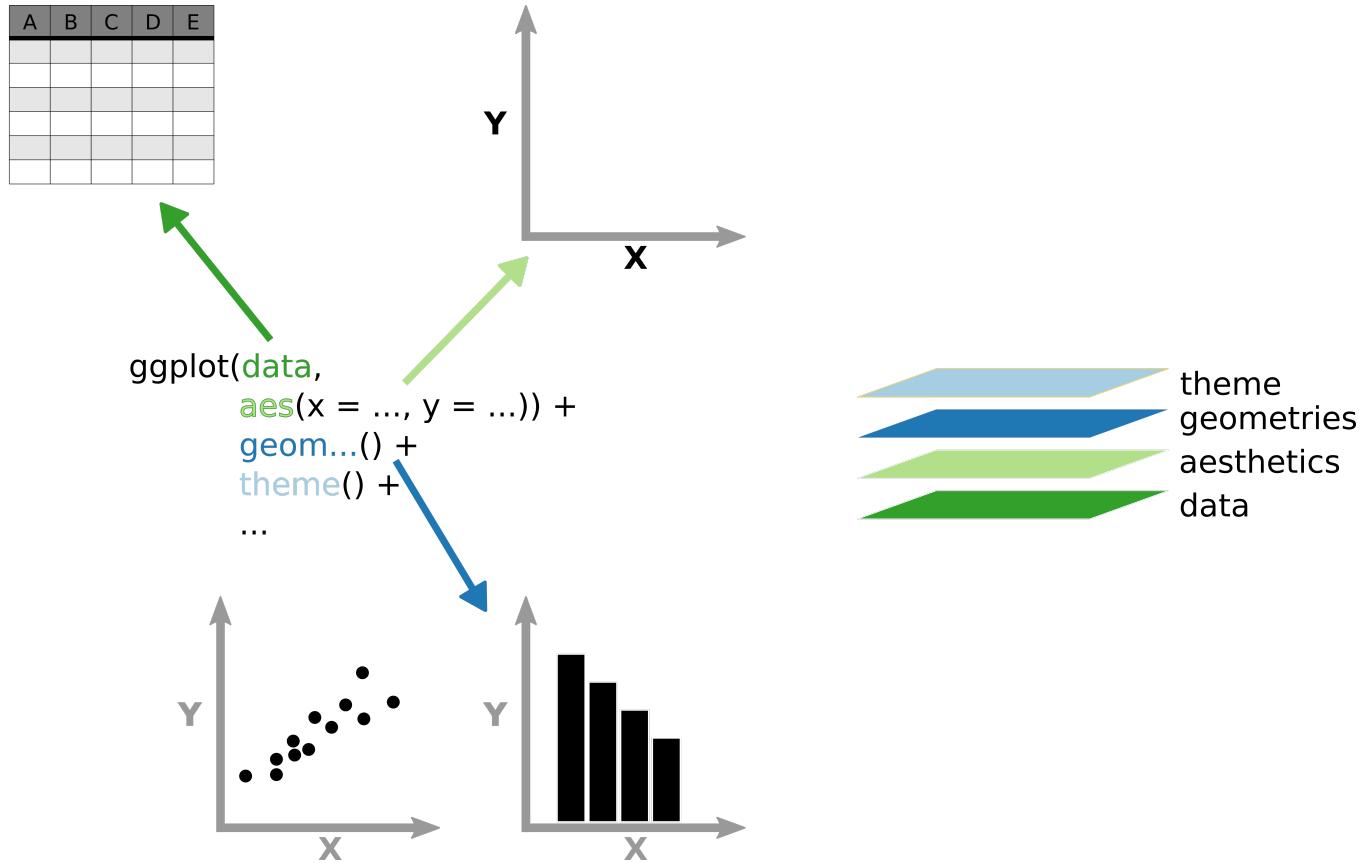
# 02\_Exercise1: left\_join()

left\_join() = to combine tables, with the left table as reference.

```
result <- data1 %>%
  left_join(data2, by = "column_name")
```



# 02\_Exercise1: Plots with ggplot2 - reminder



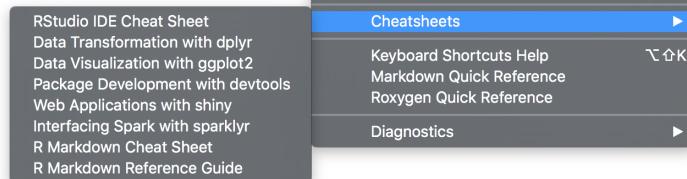
! Do not mix %>% and + (specific to plots) !

# 03\_Find\_help: in R/Rstudio

## Package documentation

```
# Package documentation  
help(package = "NAME_of_PACKAGE")  
  
# Help for a specific function  
?NAME_of_FUNCTION
```

## Rstudio cheatsheets



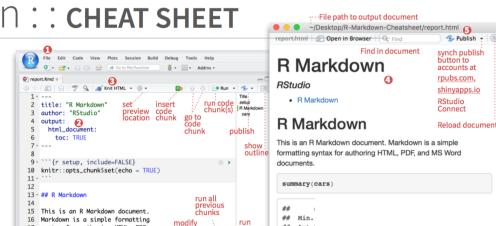
## R Markdown :: CHEAT SHEET

### What is R Markdown?

Rmd files - An R Markdown (.Rmd) file is a record of your work. It is designed to make it easy for a scientist needs to reproduce your work along with the context that a reader needs to understand your work.

Reproducible Research - At the click of a button, or the type of a command, you can generate a report from your Rmd file to reproduce your work and export the results as a finished report.

Flexibility - You can choose to export the finished report in a variety of formats, including html, pdf, MS Word or RTF documents, html help based files, Notebooks, and more.



# 03\_Find\_help: Online

Google, StackOverflow, Twitter...

but also

R meetups (R ladies, UseR groups...):

- <https://www.meetup.com>
- <https://rladies.org/>

Conferences:

useR! (Toulouse, July 9-12 2019)

SatRday (Paris, 23rd of February 2019)

Remote groups:

- R ladies remote group
- French speaking remote group "Grrr"

| Dont' forget to use a **reprex** = reproducible example.



# 04\_Exercise2: Goals

- Apply knowledge from Exercise1
- Learn how to navigate through the documentation of a package

Be aware that this part of the session relies on two libraries (`mamut2r` and `cellviz3d`) that are still under development.

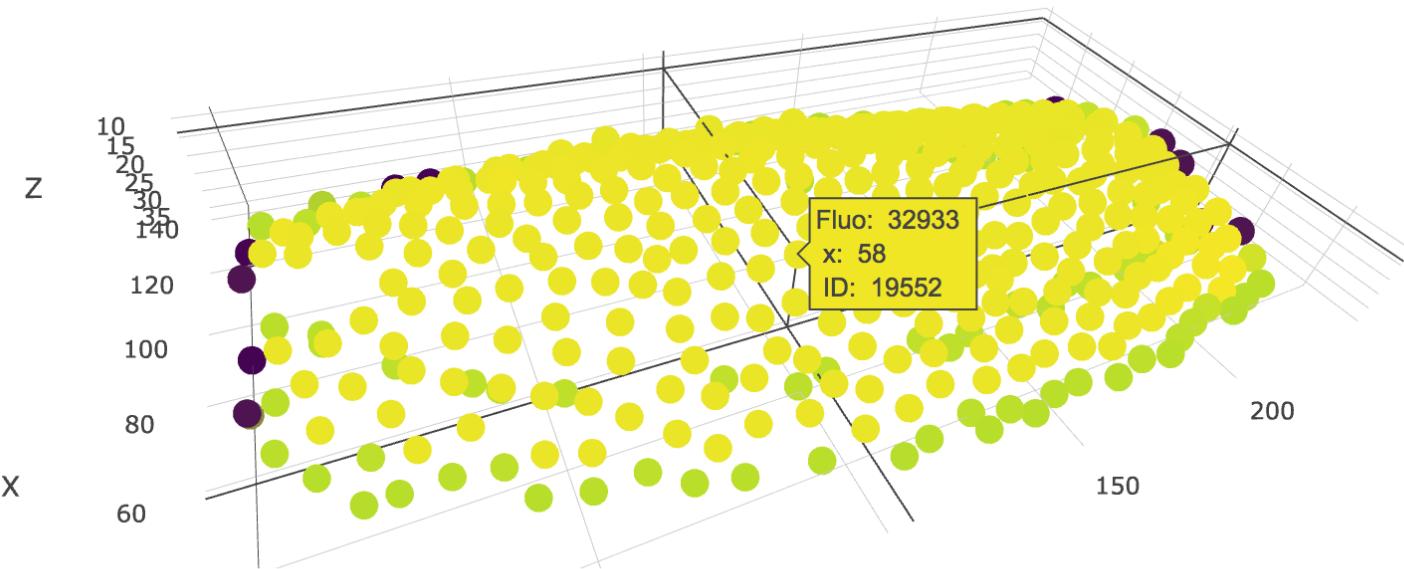


`mamut2r` and `cellviz3d` are available on GitHub:

- <https://github.com/marionlouveaux/mamut2r>
  - <https://github.com/marionlouveaux/cellviz3d>
- See `Bonus_Install.Rmd` for instructions on how to install these libraries on your own computer.

# 04\_Exercise2

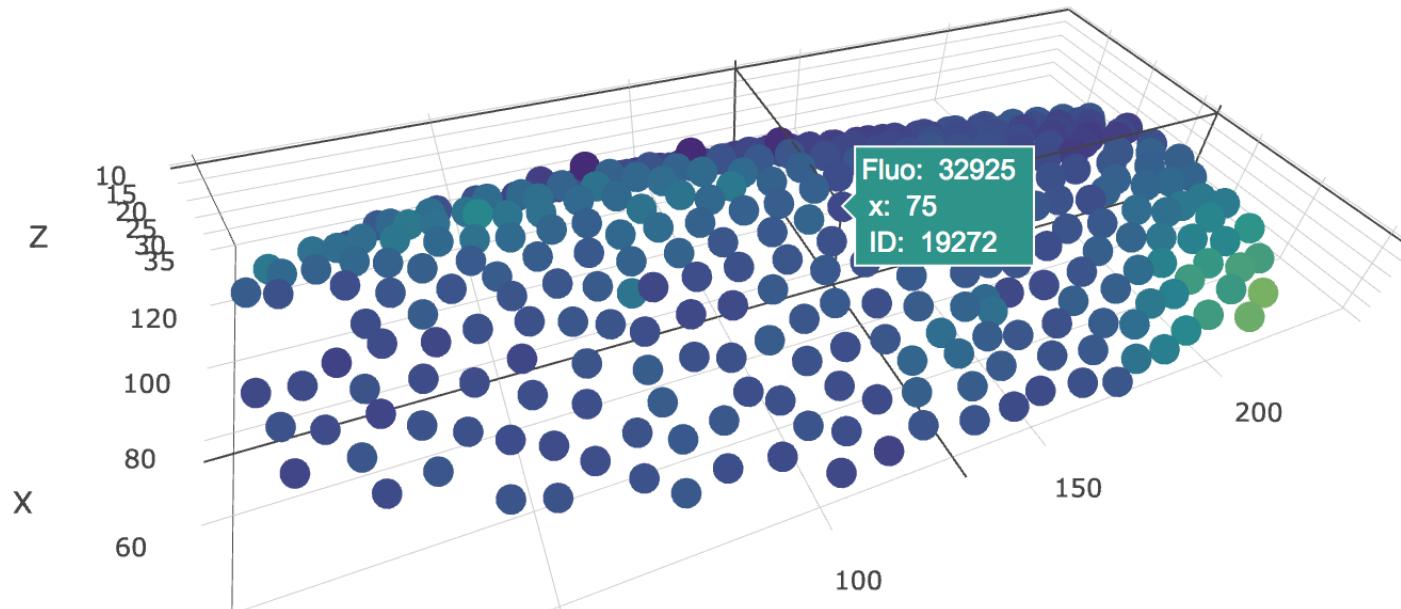
Display mean fluorescence intensity in 3D+time.



The lower values of fluorescent intensity are close to the border of the sample.

# 04\_Exercise2

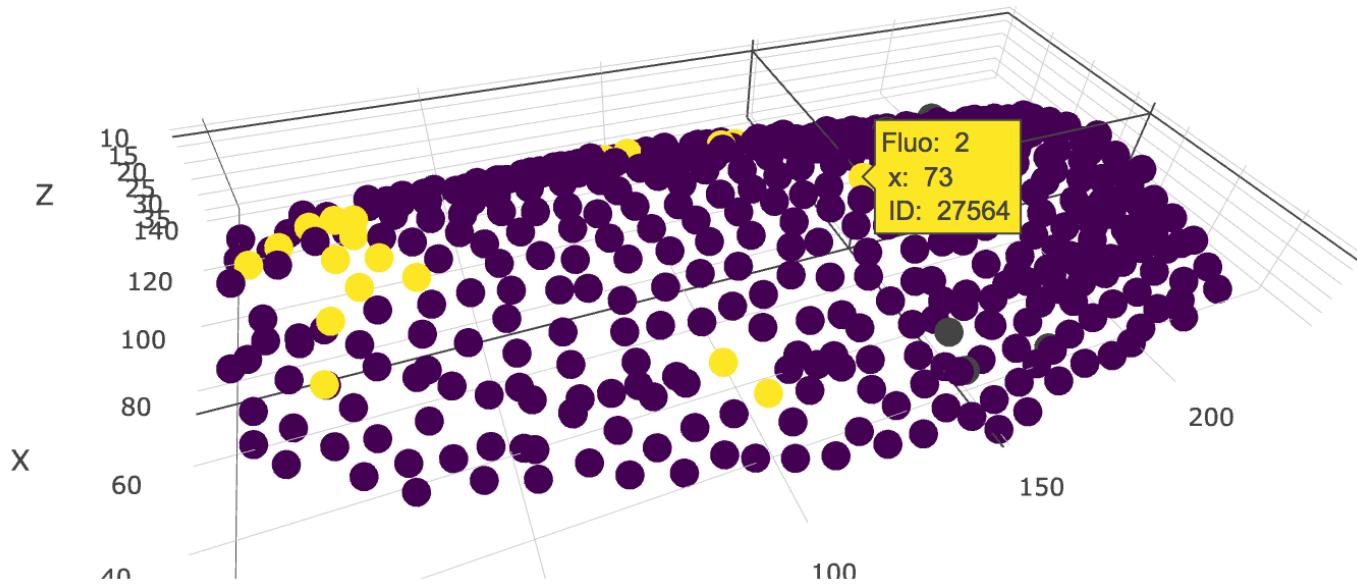
Display **filtered** mean fluorescence intensity in 3D+time.



Mean fluorescence intensity tends to be lower right after a division.

# 04\_Exercise2

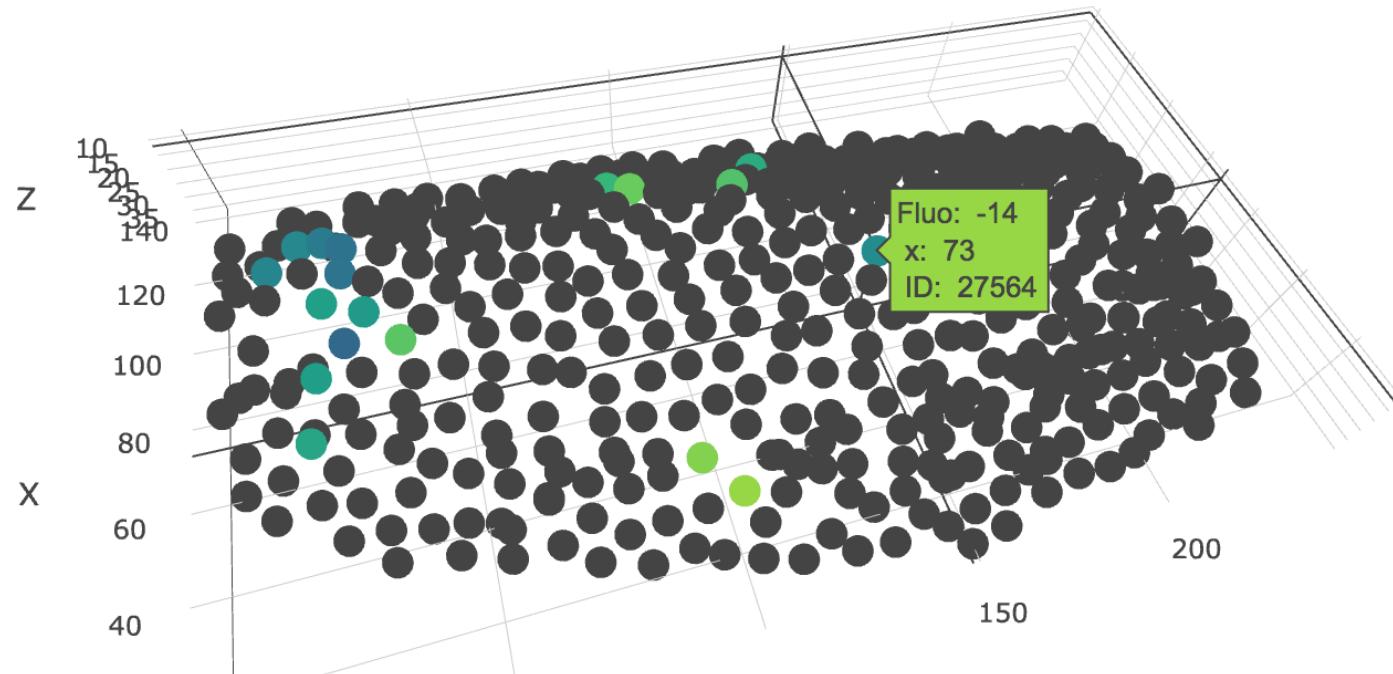
Compute and display the division status (ready to divide or not) in 3D+time.



We can spot simultaneous divisions.

# 04\_Exercise2

Compute and display a **metric to look for false detected spot** based on the change of mean fluorescence intensity over a division.



# 05\_Going\_further

## Citation

Don't forget to cite R and the packages you use!

```
# To get R citation  
citation()
```

```
# To get citation for the ggplot2 package  
citation(package = "ggplot2")
```

# Acknowledgements

- **Local organisers:** Aymeric d'Hérouel and Andreas Girod
- **NEUBIAS chair:** Julien Colombelli
- **TS10 and TS11 organisers and helpers:** Kota Miura, Jean-Yves Tinevez, Gaby Martins, Jonas Øgaard, Joyce Kao, Fabrice Cordelières, Ignacio Arganda-Carreras
- **Data:** Robert Haase
- **mamut2r and cellviz3d help:** Sébastien Rochette, Friederike Preu, Mike Smith
- **MaMuT help:** Jean-Yves Tinevez, Tobias Pietzsch, John Bogovic

The format of this session was inspired by the Little Miss Tidyverse workshop given by Danielle Navarro for the Rladies Sydney (<https://github.com/rладиessydney/littlemisstidyverse>).