www.bdd.rdplf.org Volume 2, n° 4, December 2019
https://doi.org/10.25796/bdd.v2i4.52303
ISSN 2607-9917

Le Bulletin de la Dialyse à Domicile

Statistics initiation with the free R software : make your first visualizations with the ggplot2 package

Initiation au Logiciel de statistiques R : Réalisez vos premières visualisations avec le package ggplot2

Note : ce texte est la traduction la version originale française disponible à la même adresse URL : https://doi.org/10.25796/bdd.v2i4.52303

Claire Della Vedova

1087 chemin de Sainte Roustagne, 04100 MANOSQUE, France

Editor's note: The main purpose of the RDPLF (French Language Peritoneal Dialysis and Home Hemodialysis Registry) is to help physicians and nurses, caring for patients treated by home dialysis, to evaluate their clinical practices and also to conduct studies based on anonymous exports of the data base . To this end, since June 2019, a series of training articles on the use of free R-software is published quarterly with each issue of the Bulleti, de la Dialyse à Domicile. The goal is to allow all teams to perform basic statistics and quickly visualize their data.

The first article in this tutorial was devoted to downloading and installing R software on Macintosh and PC computers: https://doi.org/10.25796/bdd.v2i2.20513.

The second article was devoted to graphical visualization of statistical data with the Esquisse package, easy to use and requiring little learning: https://doi.org/10.25796/bdd.v2i3.21313

This third article is devoted to this ggplot2 package and requires a learning effort, but it allows the realization of many types of visualizations, with renderings that can be of very high quality, and thus directly usable in publications.

As in the previous issues, an example file, taken from the RDPLF database will be used.

Claire Della Vedova is an engineer in biostatistics / data analyst, she daily uses the software R to analyze data. She has worked for more than 15 years in the fields of environment and health, and has trained many students and researchers in the use of R. She runs since November 2017 the blog Statistics and Software R whose goal is to help beginners to better understand the standard statistical methods and to use the R software more effectively, especially through tutorials: https://statistique-et-logiciel-r.com/.

The total training is done over 15 months, at the rate of one article per quarter each publication of the BDD. This will leave ample time to assimilate and test the knowledge gained between each article. For those who wish to go faster, they can go to the blog (https://statistique-et-logiciel-r.com/).

Dates of next publications:

- article 4 (April 2020): the realization of automated statistical analysis reports with Rmarkdown
- article 5 (June 2020): data manipulation (with dplyr, including the group_by and summarize functions)
- article 6 (September 2020): the realization of descriptive analyzes (statistical parameters and graphs) in the form of dashboard with the flexboard package

Mots clés : biostatistics, épidémiology, R software, RDPLF

journal officiel du Registre de Dialyse Péritonéale de Langue Française **RDPLF** www.rdplf.org

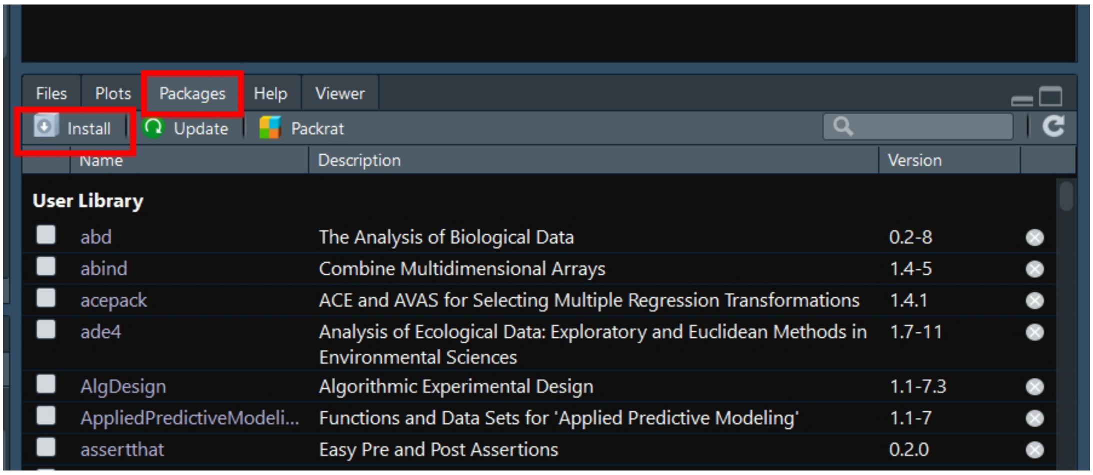
Table des matières

1. Introduction

We have previously seen how to realize visualizations, under R, thanks to the graphical interface Esquisse. This add-in is actually based on the features of the ggplot2 package. The mastery of this ggplot2 package requires a learning effort but allows the realization of many types of visualizations, with renderings that can be of very high quality, and thus directly usable in publications.
In this article, we will take a closer look at how to use this ggplot2 package, based on some practical examples.

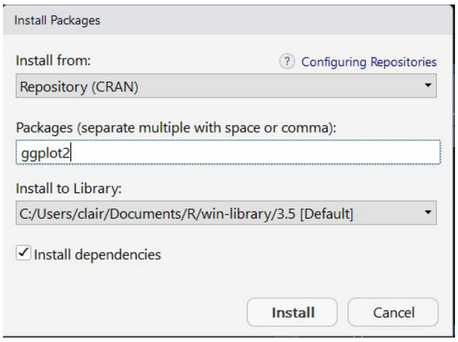
2. Installation

Like any package, you have to start by importing it. For this, we can use the R Studio package installation tool:



Name	Description	Version
User Library		
abd	The Analysis of Biological Data	0.2-8
abind	Combine Multidimensional Arrays	1.4-5
acepack	ACE and AVAS for Selecting Multiple Regression Transformations	1.4.1
ade4	Analysis of Ecological Data: Exploratory and Euclidean Methods in Environmental Sciences	1.7-11
AlgDesign	Algorithmic Experimental Design	1.1-7.3
AppliedPredictiveModeli...	Functions and Data Sets for 'Applied Predictive Modeling'	1.1-7
assertthat	Easy Pre and Post Assertions	0.2.0

then :



Install from: Repository (CRAN)

Packages (separate multiple with space or comma): ggplot2

Install to Library: C:/Users/clair/Documents/R/win-library/3.5 [Default]

☑ Install dependencies

journal officiel du Registre de Dialyse Péritonéale de Langue Française   RDPLF   www.rdplf.org

Once the package is installed, it is necessary to load it:

```
library(ggplot2)
```

It is also necessary to import the data that you want to represent!

To do this, place your data file, in csv format, in a «data» folder that you have previously created in your working folder associated with an R project. If you need more information to complete this step, consult the «Introduction to data analysis with R software» https://www.bdd.rdplf.org/index.php/bdd/article/view/20513/19163
Once the data file is placed in the «data» folder, use the following command:

```
mydata <- read.csv2 («data / MyDatafile.csv»)
```

In this article, we will use the same data as for the two previous articles, they are downloadable in csv format at this address: https://www.rdplf.org/exempleR/FileExampleStat.csv.
To import this data:

```
mydata <- read.csv2 («data / SampleStateFile.csv»)
```

Note: To reproduce the examples below, you can also download the data directly into R (without first placing them in the «data» folder), using the following command:

```
mydata <- read.csv2 («https://www.rdplf.org/exampleR/ExampleState.csv»)
```

3. Principe of operation

The ggplot2 package works in successive layers. The first of them, is a little canvas of the graph. It consists of indicating where the data are, and which variables are to be represented.
Then, a second layer is added, it consists, for example, to indicate the type of graph which one wishes to realize: scatterplot, boxplot, barplot etc ...
Then come the ripening layers, which will allow you to define new colors, axis scales, legend options, etc.

3.1 Definition of canevas layer

To define the canvas layer (or the skeleton of the graph), we use the function ggplot () and its argument, the function aes () for «aesthetic», which defines the variables to be represented and the visual properties of the graphical representation. This can include size, shape, dot color of graphs, etc.

Graphs built with ggplot2 always start with this type of line of code:

```
ggplot (dataset, aes (x =, y =))
```

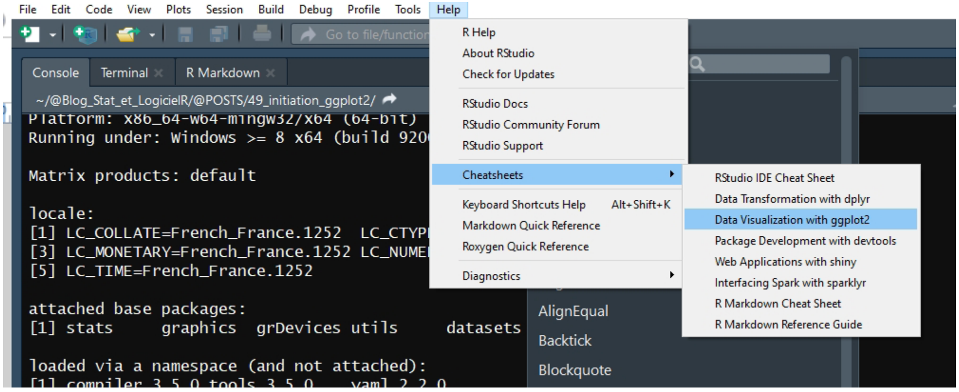
What to add the type of graph that you want.

3.2 Definition of the plot type : geom_XXX

It is now a question of defining the type of graphic that one wishes to realize: a scatter plot, a boxplot, a barplot, etc ...
For this, we add a plus sign at the end of the first line (that of the canvas), and add a new line with the appropriate function:
- geom_point () for a scatter plot,
- geom_boxplot () for a boxplot,
- geom_bar () for a barplot etc ...

All available geom_XXX () functions are described in the «Geoms» part of the cheatsheet of the ggplot package.

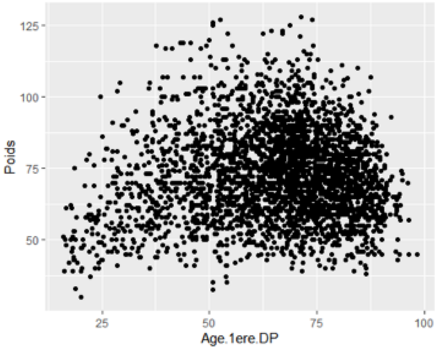
This cheatsheet can be downloaded automatically by going to the Help -> Cheatsheets -> Data Visualization with ggplot2 tab.



```
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows >= 8 x64 (build 920

Matrix products: default

locale:
[1] LC_COLLATE=French_France.1252  LC_CTYP
[3] LC_MONETARY=French_France.1252 LC_NUME
[5] LC_TIME=French_France.1252

attached base packages:
[1] stats     graphics  grDevices utils     datasets

loaded via a namespace (and not attached):
[1] compiler_3.5.0 tools_3.5.0    yaml_2.2.0
```

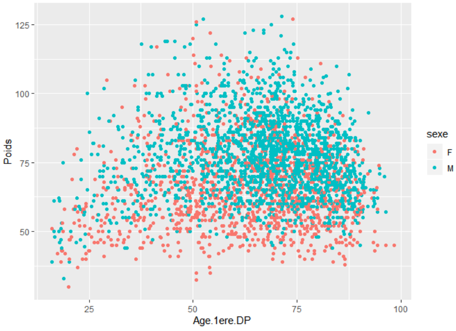
4. The main types of graphics or plots

4.1. The basic scatterplot



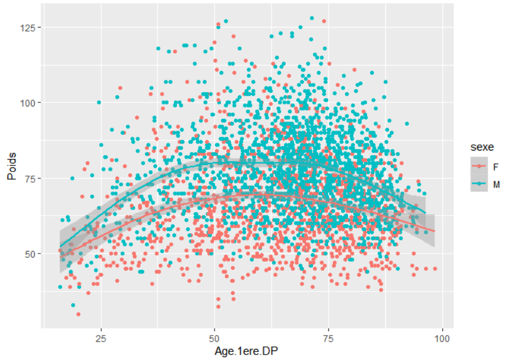
Let’s imagine that we want to make a scatterplot of the variables Age.1ere.DP (in X) and Weight (in Y). In this case, you will need to use the following commands:

```
ggplot (mydata, aes (x = Age.1ere.DP, y = Weight)) +
    geom_point ()
```



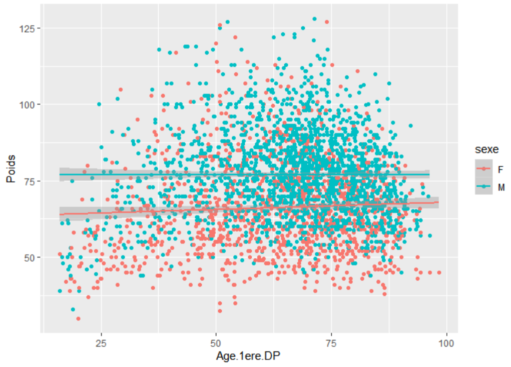
To represent the points with a different color for men and women, we will define the color argument in the aes () function of the «canvas» part:

```
ggplot (mydata, aes (x = Age.1ere.DP, y = Weight, color = sex)) +
geom_point ()
```



It is still possible to add local regression lines with the geom_ smooth function:

```
ggplot (mydata, aes (x = Age.1ere.DP, y =Weight, color = sex))+
  geom_point () +
 geom_smooth ()
```



To add a linear regression line, it is necessary to add the argument method = «lm» in the geom_smooth () function, like this:

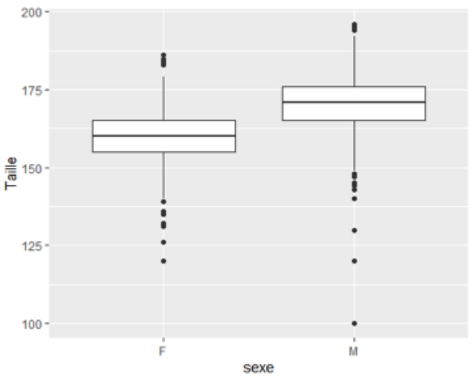
```
ggplot (mydata, aes (x = Age.1ere.DP, y = Weight, color = sex))+
 geom_point () +
geom_smooth (method = «lm»)
```

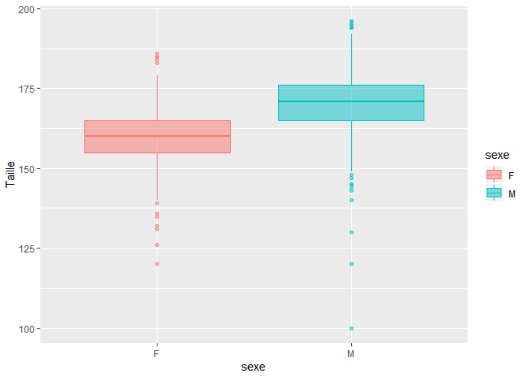
4.2. The basic boxplot

If we want to make a boxplot to visualize the distribution of patient sizes by sex, we can use the following commands:

```
ggplot (mydata, aes (x = gender, y = Size)) + geom_boxplot ()
```

To use colors in the boxes, here we use the argument fill and not color, in the aes () function. The color argument in aes () adds colors to points and lines. While the fill argument can fill forms. It is also possible to reduce the intensity of the color of the boxplots using the alpha argument.



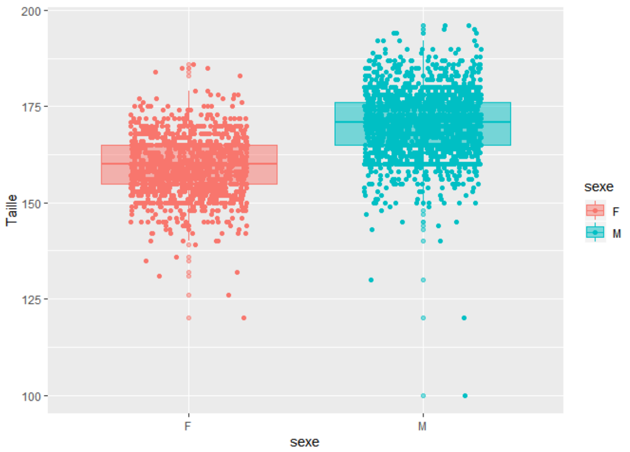


So to distinguish with colors men and women, we can use the following commands:

```
ggplot(mydata, aes( x = sexe , y = Taille, fill=sexe, colour=sexe))+geom_boxplot(alpha=0.5)
```

If we want to add the points by shifting them slightly horizontally, we can use the layer geom_jitter () with the argument width = 0.25 so that the points remain contained in the boxes. The argument heigth = 0, allows not to have a vertical offset.

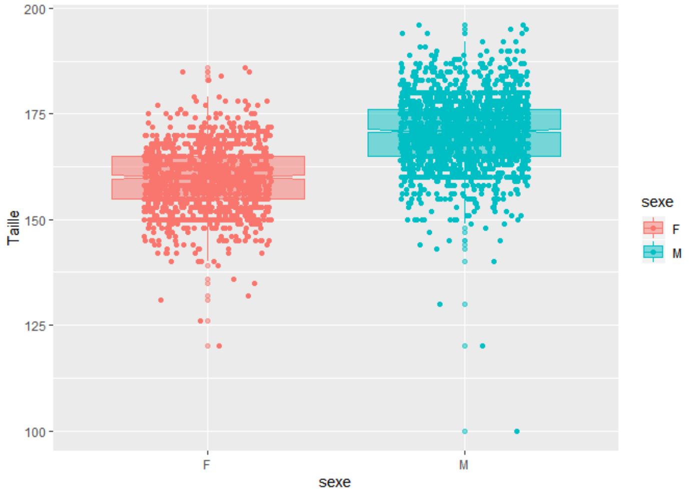
```
ggplot(mydata, aes( x = sexe , y = Taille, fill=sexe, colour=sexe))+
    geom_boxplot(alpha=0.5)+
```



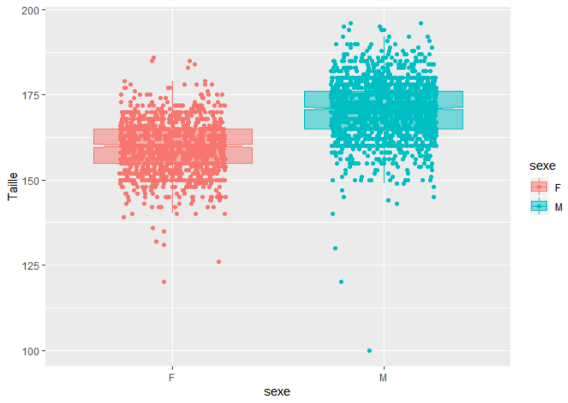
```
    geom_jitter(width=0.25, height=0)
```

We can still add notchs, that is, notches that correspond to the 95% confidence intervals of the medians, by using the notch = TRUE argument in the geom_boxplot () function:

```
ggplot(mydata, aes( x = sexe , y = Taille, fill=sexe, colour=sexe))+
    geom_boxplot(alpha=0.5, notch=TRUE)+
```



```
    geom_jitter(width=0.25, height=0)
```



Finally, the outliers, visualized by points beyond the vertical lines, are represented twice, a first time by the geom_boxplot () layer (the points are clearer) and a second time by the geom_jitter () layer. To avoid this double representation, it is possible to make them invisible in the geom_boxplot () layer, using the argument outlier.alpha = 0:

```
ggplot(mydata, aes( x = sexe , y = Taille, fill=sexe, colour=sexe))+
geom_boxplot(alpha=0.5, notch=TRUE, outlier.alpha=0)+
   geom_jitter(width=0.25, height=0)
```

4.3. Réaliser des barplots avec ggplot2

4.3.1 «Univariate» Barplots

We use here the geom_bar () function.

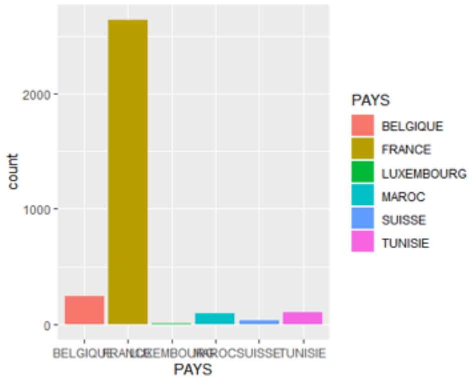
Imagine, for example, that we want to represent the number of data by country (pays in french). In this situation we will define the variable x as the PAYS variable. And to add a different color, we will add the argument fill = PAYS.

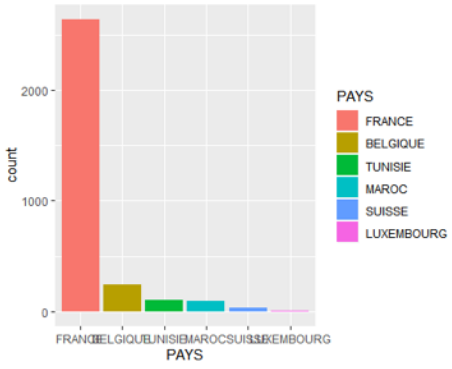
```
ggplot (mydata, aes (x = PAYS, fill = PAYS)) +
  geom_bar ()

ggplot(mydata, aes(x=PAYS, fill=PAYS)) +
 geom_bar()
```

It is of course possible to reorganize the countries in descending order of strength. For that, we can use the forecats package (it is necessary to install it before loading it). Then redefine the order of the modalities of the country variable, using the function fct_infreq (), like this:

```
library(forcats)
mydata$PAYS <- fct_infreq(mydata$PAYS)
```

then redo the bar graph:





```
ggplot(mydata, aes(x=PAYS, fill=PAYS)) +
geom_bar()
```

4.3.2 «Bivariate» Barplots

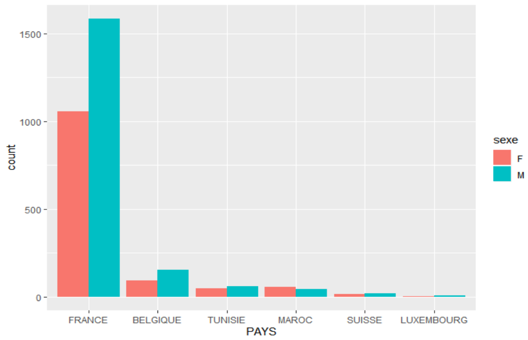
It is also possible to very easily represent the data size of a variable, according to the modalities of a second variable. For example, we might need to represent the amount of data for each country, but distinguishing between men and women.

Three types of representations are then possible. They are named “dodge”, “stack” and “fill” in ggplot2.

T

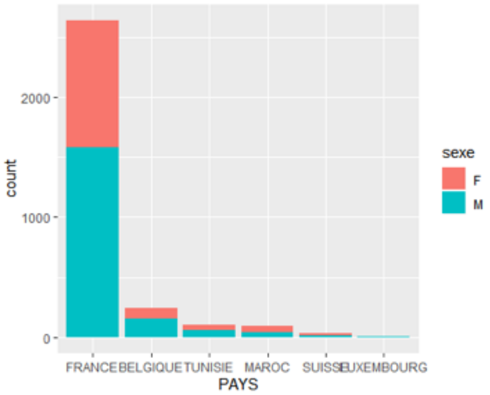
he position «dodge» or juxtaposed, can be obtained by using the argument fill = sex in the function aes (), then the argument position = dodge in the function geom_bar ():
# no y because it’s a count

```
ggplot(mydata, aes(x=PAYS, fill=sexe)) +
 geom_bar(position=»dodge»)
```



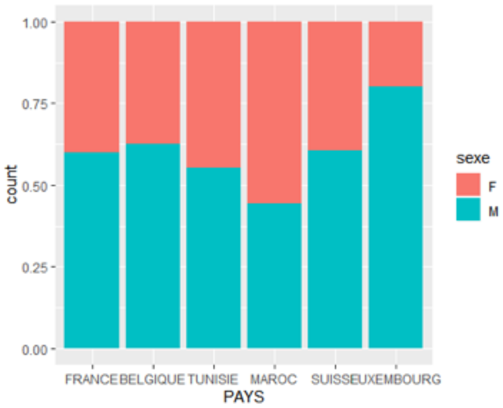
The position “stack” or stacked, can be obtained by simply using the argument fill = gender in the function aes ():
# no y because it’s a count

```
ggplot(mydata, aes(x=PAYS, fill=sexe)) +
 geom_bar()
```



To reduce the number to 100%, add the argument fill = sex in the function aes (), then the argument position = fill in the function geom_bar ():
# no y because it’s a count

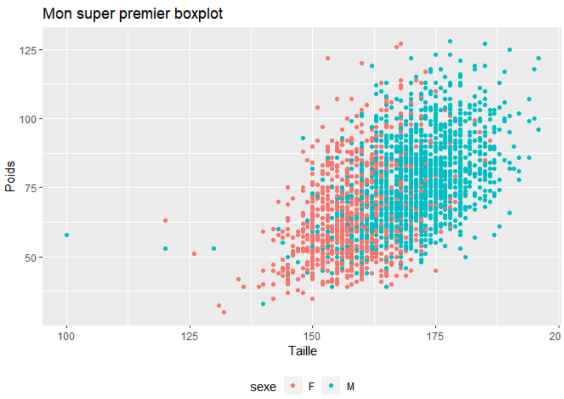
```
ggplot(mydata, aes(x=PAYS, fill=sexe)) +
 geom_bar(position=»fill»)
```



4.4. Les axes, titles et legend

You can give a title to your graph with the ggtitle () function, then rename the axes with ylab () and xlab (). The position of the legend is managed with theme (legend.position = «»).

```
ggplot(mydata, aes(x=Taille, y=Poids, colour=sexe))+
   geom_point()+
 ggtitle(«Mon super premier boxplot»)+
   ylab(«Poids»)+
   xlab(«Taille»)+
   theme(legend.position=»bottom»)
```



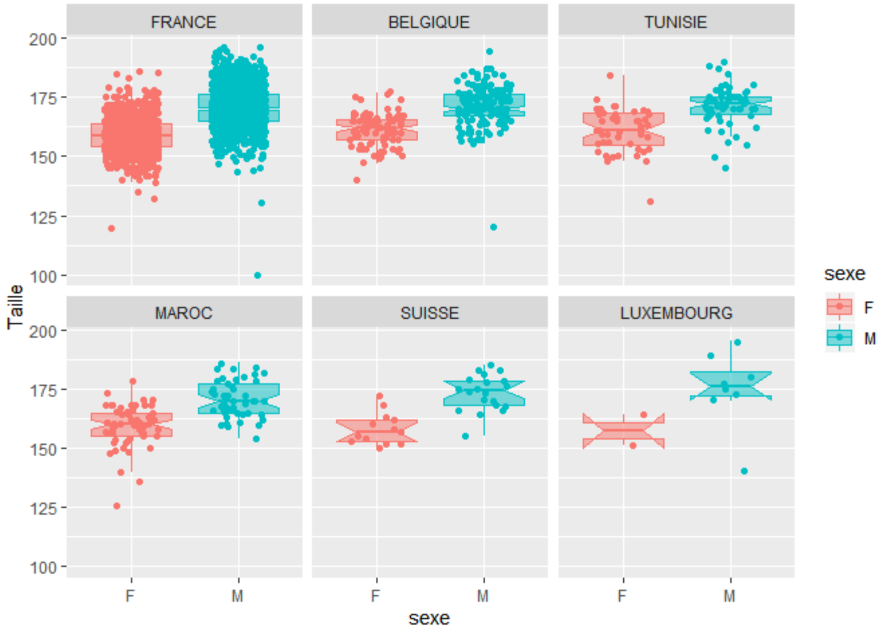
5. The faceting

This is one of the great possibilities of ggplot2. This consists of sub-dividing a graph, according to the modalities of one or more variables.

Here, for example, we will now see the distribution of patient sizes by country, always distinguishing between women and men.

```
ggplot(mydata, aes( x = sexe , y = Taille, fill=sexe, colour=sexe))+
   geom_jitter(width=0.25)+
   geom_boxplot(alpha=0.5, notch=TRUE, outlier.alpha=0)+
   facet_wrap(vars(PAYS))
```

The facetting is carried out using the functions
 facet_grid() et facet_wrap().



6. How to find help and progress ?

French resources on the ggplot2 package are rare. Nevertheless, many information and examples are provided in Chapter 8 of Julien Barnier's introduction to tidyverse: https://juba.github.io/tidyverse/08-ggplot2.html. You can also read my articles «Introduction to the visualization under R with the ggplot2 package» (https://statistics-and-software-r.com/introduction-at-the-visualization-with-the-package -ggplot2 /) and and «How to change colors with ggplot2» (https://statistics-and-software-r.com/__trashed-2/).
You can still:
• Use the cheat sheet of the ggplot2 package: it contains a lot of information,
• Use help on functions: for example? Geom_text (),
• consult the reference book: «R Graphics Cookbook» https://r-graphics.org/
• See the examples of «The R Graph Gallery» (https://www.r-graph-gallery.com/) that contain the lines of code
• write your question in english in google / stackoverflow.

Conclusion

I hope this article will make you want to try making your first graphs with ggplot2. And that he will then be a step in the stirrup towards other more complete representations. The ggplot2 package has almost unlimited possibilities, which will allow you to realize effective visualizations to explore your data, but also elegant representations for your communication media, or for your publications.