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

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

R software statistics intitiation : make your first visualizations with the ggplot2 package

Claire Della Vedova

1087 chemin de Sainte Roustagne, 04100 MANOSQUE, France

NDLR : Le RDPLF a pour but principal d'être une aide pour permettre aux équipes de dialyse à domicile d'évaluer leurs pratiques cliniques et également conduire des études à partir d'exports anonymisés des données qu'elles saisissent. A cette fin, depuis juin 2019, un article de formation à l'utilisation du logiciel Libre R est publié trimestriellement à chaque parution du Bulletin de la Dialyse à Domicile. Le but est de permettre à toutes les équipes de réaliser des statistiques de bases et visualiser rapidement leur données.

Le premier article de cette série d'initiation était consacré au téléchargment et à l'installation du logiciel R sur les ordinateurs Macintosh et PC : https://doi.org/10.25796/bdd.v2i2.20513 .

Le second article était consacré à la visualisation graphiques des données statistiques avec le package Esquisse, simple d'utilisation et nécessitant peu d'apprentissage : https://doi.org/10.25796/bdd.v2i3.21313

Ce troisième article est consacré au package ggplot2 et nécessite un effort d'apprentissage, mais il permet la réalisation de nombreux types de visualisations, avec des rendus qui peuvent être de très haute qualité, et ainsi directement utilisables dans des publications.

Comme dans les numéros précédent un fichier exemple, tiré de la base de données du RDPLF sera utlisé.

Claire Della Vedova est Ingénieure en biostatistique / data analyste, Elle utilise quotidiennement le logiciel R pour analyser des données. Elle a travaillé pendant plus de 15 ans dans les domaines de l'environnement et de la santé, et a formé de nombreux étudiants et chercheurs à l'utilisation de R. Elle anime depuis novembre 2017 le blog Statistique et Logiciel R dont le but est d'aider les débutants à mieux appréhender les méthodes statistiques classiques et à utiliser le logiciel R plus efficacement, notamment au travers de tutoriels : https://statistique-et-logiciel-r.com/.

La formation totale se fait sur 15 mois, au rythme d'un article par trimestre à chaque parution du BDD. Cela laissera largement le temps d'assimiler et tester les connaissances acquises entre chaque article. Pour ceux qui souhaiteront aller plus vite, ils pourront aller sur le blog (https://statistique-et-logiciel-r.com/).

Dates des prochaines parutions :

- article 4 (Avril 2020) : la réalisation de rapports d'analyses statistiques automatisées avec Rmarkdown
- article 5 (Juin 2020) : la manipulation de données (avec dplyr, notamment les fonction group_by et summarise)
- article 6 (Septembre 2020) : la réalisation d'analyses descriptives (paramètres statistiques et graphs) sous forme de dashboard avec le package flexboard

Mots clés : biostatistique, épidémiologie, logiciel R, RDPLF

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

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

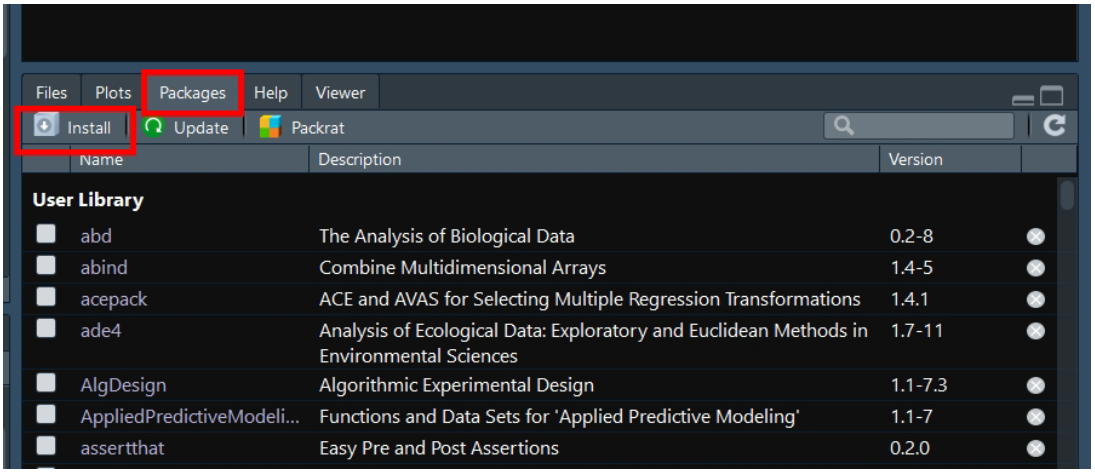
Table des matières

1. Introduction

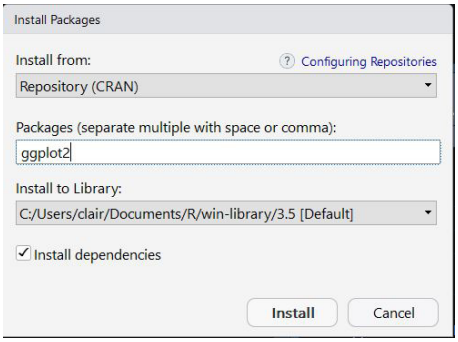
Nous avons vu précédemment comment réaliser des visualisations, sous R, grâce à l’interface graphique Esquisse. Cet add-in est en réalité basé sur les fonctionnalités du package ggplot2. La maîtrise de ce package ggplot2 nécessite un effort d’apprentissage mais permet la réalisation de nombreux types de visualisations, avec des rendus qui peuvent être de très haute qualité, et ainsi directement utilisables dans des publications. Dans cet article, nous allons regarder de plus près comment utiliser ce package ggplot2, en nous basant sur quelques exemples pratiques.

2. Installation

Comme tout package, il faut commencer par l’importer. Pour cela, nous pouvons utiliser l’outil d’installation des packages de R Studio :



puis :



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

Une fois que le package est installé, il est nécessaire de le charger :

```
library(ggplot2)
```

Il est également nécessaire d'importer les données que l'on souhaite représenter!
Pour cela, placez votre fichier de données, au format csv dans un dossier "data" que vous aurez préalablement créé dans votre dossier de travail associé à un projet R. Si vous avez besoin de plus d'informations pour réaliser cette étape, consultez l'article "Introduction à l'analyse de données avec le logiciel R" :
https://www.bdd.rdplf.org/index.php/bdd/article/view/20513/19163
Une fois le fichier de données placé dans le dossier "data", utilisez la commande suivante:
mydata <- read.csv2(«data/MonfichierdeDonnees.csv»)

Dans cet article, nous allons utiliser les même données que pour les deux articles précédents, elles sont téléchargeables au format csv à cette adresse: https://www.rdplf.org/exempleR/FichierExempleStat.csv.
Pour importer ces données:
mydata <- read.csv2(«data/FichierExempleStat.csv»)

Remarque : Pour reproduire les exemples ci-dessous, vous pouvez également télécharger directement les données dans R (sans les placer au préalable dans le dossier "data"), en utilisant la commande suivante :
mydata <- read.csv2(«https://www.rdplf.org/exempleR/FichierExempleStat.csv»)

3. Principe de fonctionnement

Le package ggplot2 fonctionne par couches successives. La première d'entre elles, est un peu le canevas du graph. Elle consiste à indiquer où se trouvent les données, et quelles sont les variables que l'on souhaite représenter.
Ensuite, une seconde couche est ajoutée, elle consiste, par exemple, à indiquer le type de graph que l'on souhaite réaliser : scatterplot, boxplot, barplot etc…
Viennent ensuite les couches d'affinage en quelque sorte, qui vont permettre de définir de nouvelles couleurs, les échelles des axes, les options de légende, etc…

3.1 Définition de la couche canevas

Pour définir la couche canevas (ou le squelette du graph), on utilise la fonction ggplot() et son argument, la fonction aes() pour "aesthetic", qui définit les variables à représenter et les propriétés visuelles de la représentation graphique. Cela peut inclure la taille, la forme, la couleur des points des graphs, etc…

Les graphs construits avec ggplot2 commencent donc toujours par ce type de ligne de code :

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

Ce à quoi il faut ajouter le type de graph que l'on souhaite.

3.2 Définition du type de plot : geom_XXX

Il s'agit donc à présent de définir le type de graphique que l'on souhaite réaliser : un scatter plot, un boxplot, un barplot, etc…

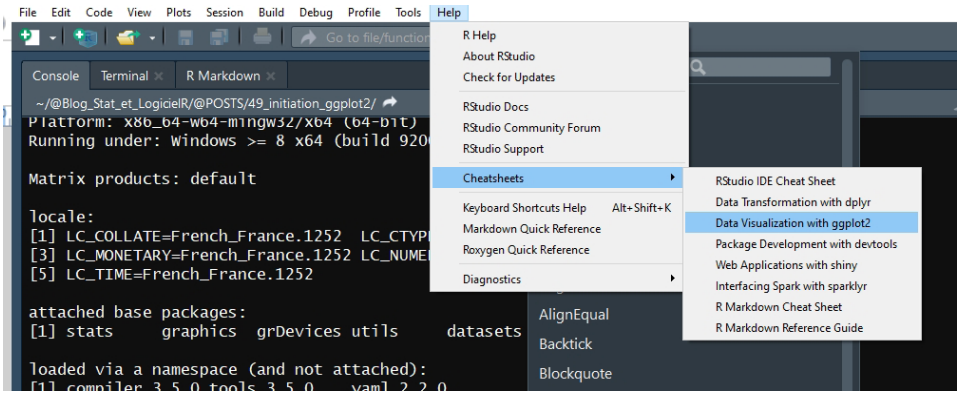
Pour cela, on rajoute un signe plus en bout de la première ligne (celle du canevas), et on ajoute une nouvelle ligne avec la fonction adéquate :
- geom_point() pour un scatter plot,
- geom_boxplot() pour un boxplot,
- geom_bar() pour un barplot etc…

Toutes les fonctions geom_XXX() disponibles sont décrites dans la partie "Geoms" de la cheatsheet du package ggplot.

Cette cheatsheet peut se télécharger automatiquement, en allant dans l'onglet

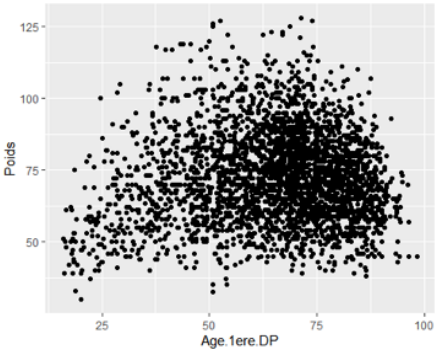
www.bdd.rdplf.org Volume 2, n° 4, December 2019
https://doi.org/10.25796/bdd.v2i4.52303
ISSN 2607-9917

Help –> Cheatsheets –> Data Visualization with ggplot2.



4. Les principaux types de graphiques ou plots

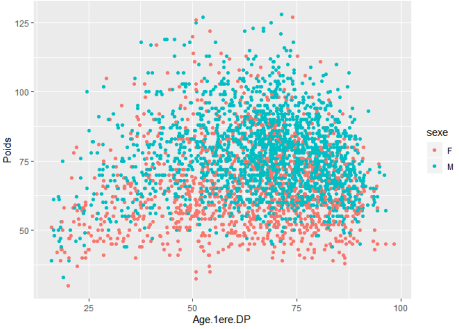
4.1. Le scatterplot de base



Imaginons que l'on souhaite réaliser un scatterplot des variables Age.1ere.DP (en X) et Poids (en Y). Dans ce cas, il faudra utiliser les commandes suivantes :

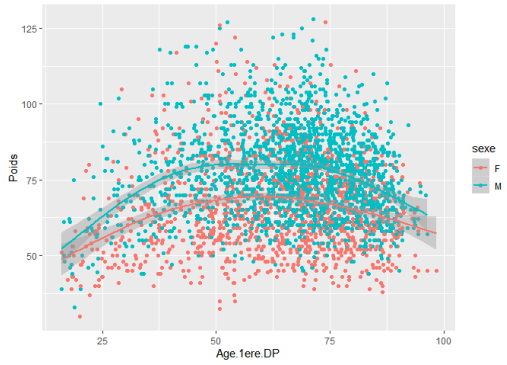
```
ggplot(mydata, aes(x=Age.1ere.DP, y=Poids))+
  geom_point()
```

Pour représenter les points avec une couleur différente pour les hommes et les femmes, nous allons définir l'argument colour dans la fonction aes() de la partie "canevas" :



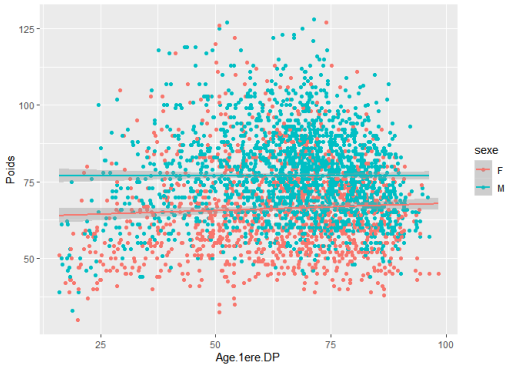

```
ggplot(mydata, aes(x=Age.1ere.DP, y=Poids, colour=sexe))+
geom_point()
```

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



Il est encore possible de rajouter des droites de régression locales avec la fonction geom_smooth:

```
ggplot(mydata,aes(x=Age.1ere.DP,y=Poids, colour=sexe))+
    geom_point()+
    geom_smooth()
```



Pour ajouter une droite de régression linéaire, il est nécessaire d'ajouter l'argument method=»lm» dans la fonction geom_smooth(), comme ceci :

```
ggplot(mydata, aes(x=Age.1ere.DP, y=Poids, colour=sexe))+
    geom_point()+
    geom_smooth(method=»lm»)
```

4.2. Le boxplot de base

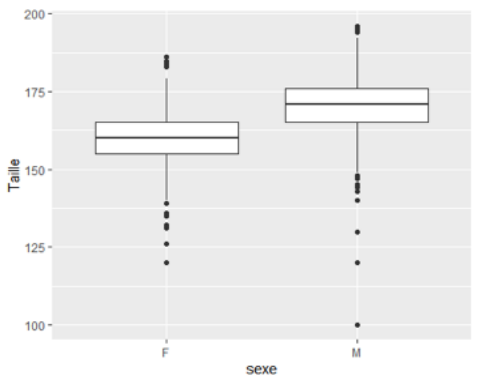
Si l'on souhaite réaliser un boxplot pour visualiser la distribution des tailles des patients par sexe, nous pouvons utiliser les commandes suivantes:

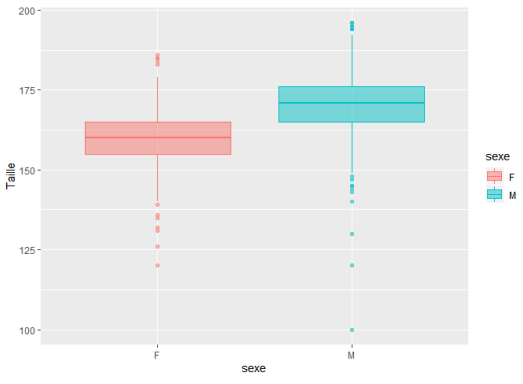
```
ggplot(mydata, aes( x = sexe , y = Taille)) + geom_boxplot()
```

Pour utiliser des couleurs dans les boites, ici on utilise l'argument fill et non colour, dans la fonction aes(). L'argument colour dans aes() permet d'ajouter des couleurs aux points et aux lignes. Alors que l'argument fill permet de remplir des formes. Il est également possible de réduire l'intensité de la couleur des boxplots en utilisant l'argument alpha.

Ainsi pour distinguer avec des couleurs les hommes et les femmes, nous pouvons utiliser les commandes suivantes :

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

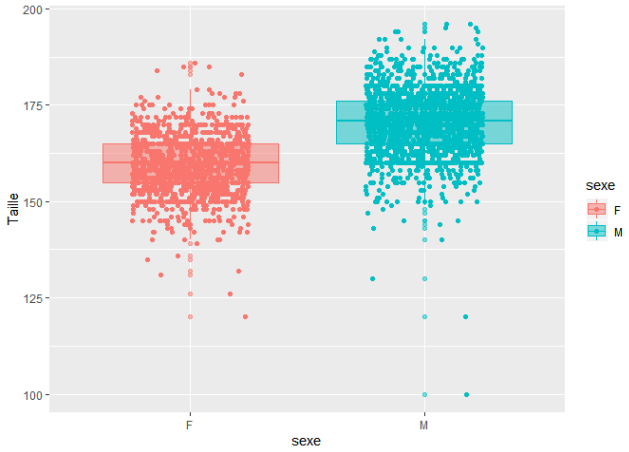




www.bdd.rdplf.org Volume 2, n° 4, December 2019
https://doi.org/10.25796/bdd.v2i4.52303
ISSN 2607-9917

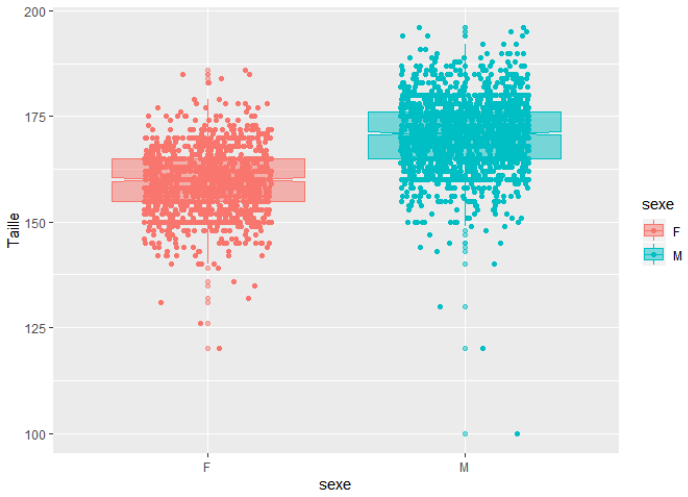
Si nous voulons ajouter les points en les décalant légèrement horizontalement, nous pouvons utiliser la couche geom_jitter() avec l'argument width=0.25 pour que les points restent contenus dans les boites. L'argument heigth=0, permet de ne pas avoir de décalage vertical.

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

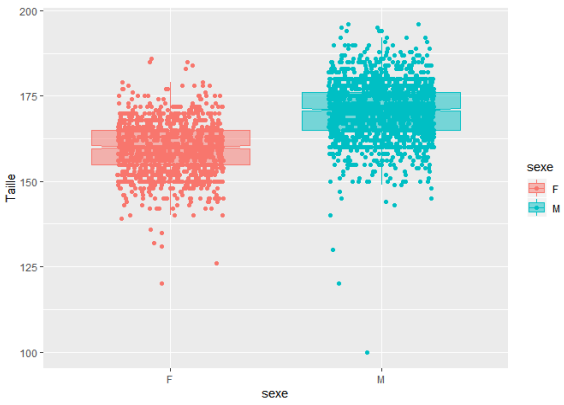


Nous pouvons encore ajouter des notchs, c'est à dire des encoches qui correspondent aux intervalles de confiance à 95% des médianes, en employant l'argument notch=TRUE dans la fonction geom_boxplot():

```
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)
```



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



Enfin, les outliers, visualisés par des points au-delà des traits verticaux, sont représentés deux fois, une première fois par la couche geom_boxplot() (les points sont plus clairs) et une seconde fois par la couche geom_jitter(). Pour éviter cette double représentation, il est possible de les rendre invisible dans la couche geom_boxplot(), en utilisant l'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 Barplots "univariés"

On utilise ici la fontion geom_bar().
Imaginons, par exemple, que l'on souhaite représenter le nombre de données par pays. Dans cette situation nous allons définir la variable x comme étant la variable PAYS. Et pour ajouter une couleur différentes, nous allons ajouter l'argument fill=PAYS.

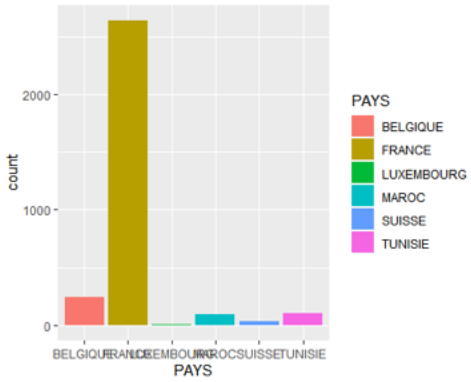
```
ggplot(mydata, aes(x=PAYS, fill=PAYS)) +
 geom_bar()
```

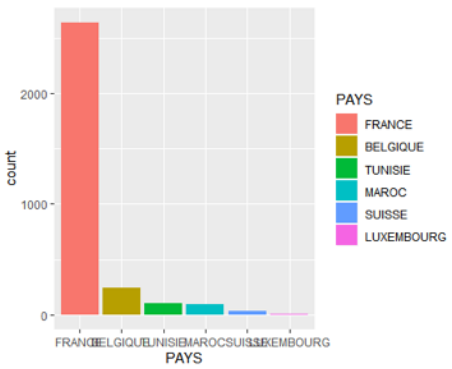
Il est bien sûr possible de réorganiser les pays dans l'ordre d'effectif décroissant. Pour cela, nous pouvons utiliser la package forecats (il est nécessaire de l'installer avant de le charger). Puis de redéfinir l'ordre des modalités de la variable pays, à l'aide de la fonction fct_infreq(), comme ceci :

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

Puis de refaire le graphique en barre :

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





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

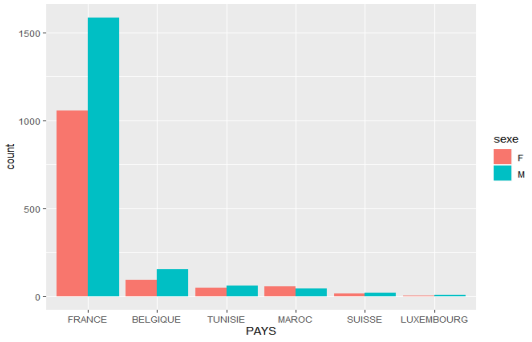
4.3.2 Barplots “bivariés”

Il est également possible de représenter très facilement les effectifs des données d’une variable, en fonction des modalités d’une seconde variable. Par exemple, nous pourrions avoir besoin de représenter le nombre de données relatives à chacun des pays, mais en distinguant les hommes et les femmes.

Trois type de représentations sont alors possibles. Elle sont nommées “dodge”, “stack” et “fill” dans ggplot2.

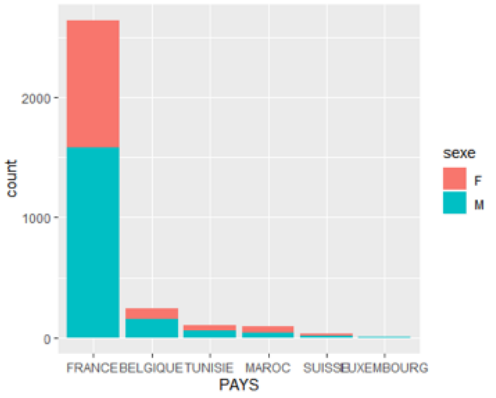
La position “dodge” ou juxtaposée, peut être obtenue en utilisant l’argument fill=sexe dans la fonction aes(), puis l’argument position=dodge dans la fonction geom_bar():

```
# pas de y car c’est un comptage
ggplot(mydata, aes(x=PAYS, fill=sexe)) +
 geom_bar(position=»dodge»)
```



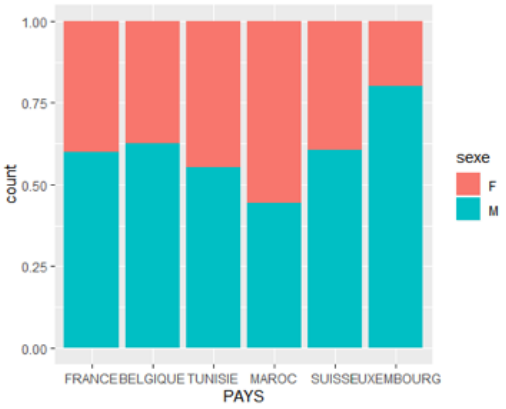
La position “stack” ou empilée, peut être obtenue en utilisant simplement l’argument fill=sexe dans la fonction aes():

```
# pas de y car c’est un comptage
ggplot(mydata, aes(x=PAYS, fill=sexe)) +
 geom_bar()
```



Pour ramener les effectifs à 100%, il faut ajouter l’argument fill=sexe dans la fonction aes(), puis l’argument position=fill dans la fonction geom_bar():

```
# pas de y car c’est un comptage
ggplot(mydata, aes(x=PAYS, fill=sexe)) +
 geom_bar(position=»fill»)
```

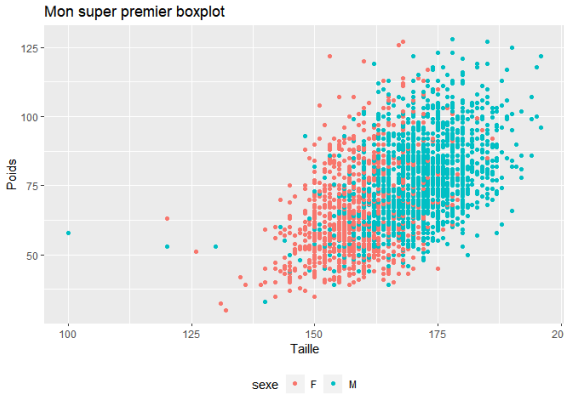


www.bdd.rdplf.org Volume 2, n° 4, December 2019
https://doi.org/10.25796/bdd.v2i4.52303
ISSN 2607-9917

4.4. Les axes, titres et légende

Vous pouvez donner un titre à votre graph avec la fonction ggtitle(), puis renommer les axes avec ylab() et xlab(). La position de la légende se gère avec 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»)
```

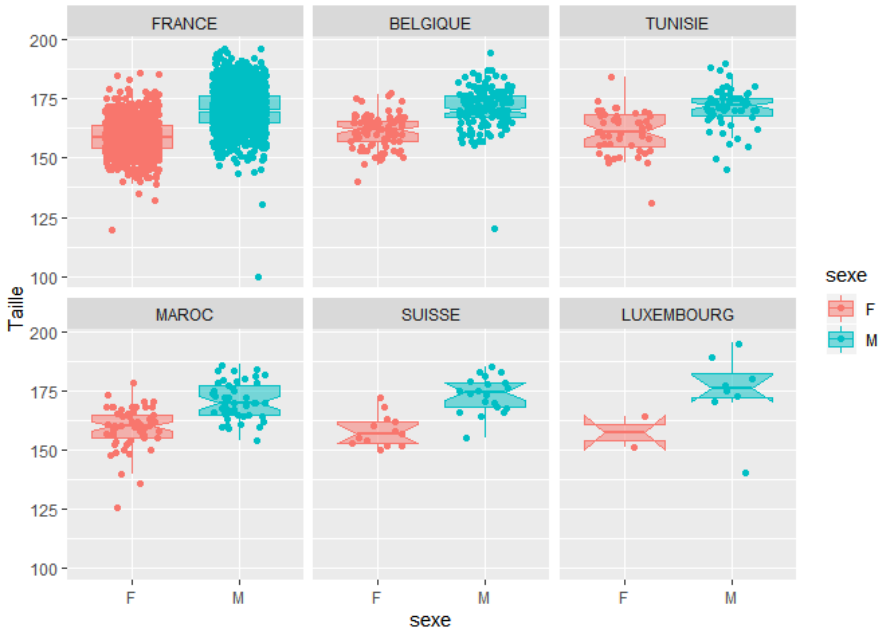


5. Le faceting

C'est une des grandes possibilités de ggplot2. Cela consiste à sous-diviser un graphique, selon les modalités d'une ou plusieurs variables.
Ici par exemple, nous allons maintenant visualiser la distribution des tailles des patients par pays, en distinguant toujours les femmes et les hommes.

```
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))
```

Le facetting se réalise à l'aide des fonctions
facet_grid() et facet_wrap().



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

6. Comment trouver de l'aide et progresser ?

Les ressources francophones sur le package ggplot2 sont rares. Néanmoins, de nombreuses informations et exemples sont fournis dans le chapitre 8 de l'introduction au tidyverse, de Julien Barnier :

https://juba.github.io/tidyverse/08-ggplot2.html

Vous pouvez également consulter mes articles "Introduction à la visualisation sous R avec le package ggplot2" (https://statistique-et-logiciel-r.com/introduction-a-la-visualisation-sous-r-avec-le-package-ggplot2/) et et "Comment modifier les couleurs avec ggplot2" (https://statistique-et-logiciel-r.com/__trashed-2/).

Vous pouvez encore :

- Utiliser la cheat sheet du package ggplot2 : elle contient beaucoup d'informations,
- Utiliser l'aide sur les fonctions : par exemple ?geom_text(),
- consulter le livre de référence : "R Graphics Cookbook" https://r-graphics.org/
- consulter les exemples de "The R Graph Gallery"(https://www.r-graph-gallery.com/) qui contiennent les lignes de codes
- écrire votre question en anglais dans google / stackoverflow.

Conclusion

J'espère que cet article vous donnera envie d'essayer de faire vos premiers graphiques avec ggplot2. Et qu'il sera alors un pied à l'étrier vers d'autres représentations plus complètes. Le package ggplot2 à des possibilités quasi illimitées, qui vous permettront de réaliser des visualisations efficaces pour explorer vos données, mais aussi des représentations élégantes pour vos supports de communication, ou encore pour vos publications.