Capsule 11. Visualiser des données - Partie 1

Simon Laffue
15/02/2023

Introduction

La visualisation des données nous permet de représenter graphiquement les variables et les relations entre les variables d'un jeu de données. Les graphiques nous sont aussi utiles lorsque nous voulons transmettre de l'information visuellement ou regarder des données sous un autre angle. La visualisation peut être une étape très importante telle qu'illustrée en 1973 par le statisticien Francis Anscombe avec son quartet anscombe. Un *quartet anscombe* est constitué de quatre ensembles de données dont les propriétés statistiques de base sont les mêmes. Nous ne pouvons pas alors distinguer facilement ces variables en regardant leur moyenne, leur variance, etc. La visualisation graphique est un outil qui va nous permettre de bien comprendre les différences entre ces jeux de données.

Base de données

Afin de réaliser les exemples de cette capsule, nous importons le jeu de données `apachephyter.csv` qui contient les variables utilisées pour calculer l' Acute Physiology Score ~ un indicateur de la sévérité d'une maladie à l'admission aux soins intensifs aux États-Unis durant la période de 2014-2015. Les fonctions de la librairie `ggplot2` dans le serront utilisées pour réaliser nos figures.

```
# Importation de la librairie ggplot2 pour car graphique.
library(ggplot2)

# Importation de la librairie dplyr pour la traitement des données.
library(dplyr)

# Importation des données.
apache_variables <- read.csv("apachephyter.csv", na.strings = "-1")

# Formatage des variables catégoriques.
apache_variables <- mutate_at(apache_variables,
                              c('intubated','vent','dialysis','eyes','motor','verbal','meds'),
                              factor)
```

Principes de grammaire des graphiques

La librairie `ggplot2` contient plusieurs outils et structures implémentés permettant de construire des figures diverses. La création d'un graphique commence toujours par la création d'un objet `ggplot`. Après l'argument `data` ~ qui dicte le jeu de données utilisé, le premier élément essentiel est le `mapping` ~ avec son argument `aes(x = , y = , ...)` qui dicte l'esthétique (ou propriété visuelle) des variables qu'on retrouve sur les axes de notre graphique. A ce stade, si l'information utilisée pour produire notre graphique provient d'un seul jeu de données, il est possible de faire la spécification des propriétés visuelles dès la création de l'objet `ggplot`. Celle-ci comporte les éléments des axes, de la couleur, des formes, des groupes (si il a lieu) et du jeu de données. Il est aussi possible de les spécifier dans les composantes géométriques de la création de notre graphique. A titre d'exemple, les trois commandes suivantes sont équivalentes, où la spécification des axes et du jeu de données sont faits lors de la création de l'objet ou dans la composante du type de graphique.

```
ggplot(data = apache_variables, mapping = aes(x = heartrate)) +
  geom_histogram()

ggplot(data = apache_variables) +
  geom_histogram(mapping = aes(x = heartrate))

ggplot() +
  geom_histogram(data = apache_variables, mapping = aes(x = heartrate))
```

Par la suite, les deuxièmes éléments essentiels que nous spécifions sont les éléments géométriques qui dictent la façon dont les données vont être représentées, c'est-à-dire le type de graphique. La syntaxe est généralement `geom_()` qu'on va ajouter avec l'opérateur "+" qui permet d'ajouter une couche à notre graphique. Finalement, nous pouvons spécifier des options de forme pour personnaliser notre figure. Les différents aspects de modification d'un graphique seront abordés dans la capsule 12.

Exemple

Voyons séquentiellement comment il nous est possible de construire un histogramme du rythme cardiaque avec la grammaire de base des graphiques.

```
# Création objet ggplot avec les esthétiques de notre graphique
p <- ggplot(data = apache_variables, mapping = aes(x = heartrate))
p
```


```
# Spécification du type de graphique
p <- p + geom_histogram(fill = "lightgreen", color = "white")
p
```


```
# personnalisation du graphique
p <- p + xlab("Fréquence cardiaque") +
  ylab("Effectifs") +
  ggtitle("Histogramme de la fréquence cardiaque") +
  theme(plot.title = element_text(size = 15),
        axis.title = element_text(size = 12))
p
```

Graphiques

Plusieurs types de graphiques sont disponibles avec la librairie `ggplot2`. Voyons les graphiques couramment utilisés pour représenter nos données, soit le nuage de points, le diagramme à bande, l'histogramme, la boîte à moustache et le diagramme à violon.

Nuage de points

Le graphique en nuage de points, ou diagramme de dispersion, permet d'illustrer les points des données dans un système de coordonnées de deux variables. Il permet d'illustrer la relation entre ces deux variables généralement continues, mais il est aussi possible de faire un diagramme de dispersion pour une variable continue en fonction d'une variable catégorique.

```
ggplot(data = apache_variables) +
  geom_point(mapping = aes(x = respiratoryrate, y = heartrate))
```

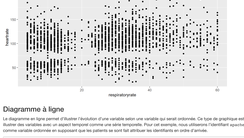


Diagramme à ligne

Le diagramme en ligne permet d'illustrer l'évolution d'une variable selon une variable qui serait ordonnée. Ce type de graphique est parfait pour illustrer des variables avec un aspect temporel comme une série temporelle. Par exemple, nous utiliserons l'identifiant `apachepatientid` comme variable ordonnée en supposant que les patients se sont fait attribuer les identifiants en ordre d'arrivée.

```
ggplot(data = apache_variables) +
  geom_line(mapping = aes(x = apachepatientid, y = temperature))
```

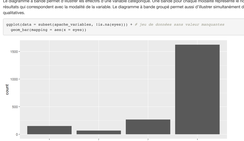
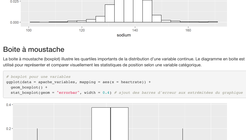


Diagramme à bande

Le diagramme à bande permet d'illustrer les effectifs d'une variable catégorique. Une bande pour chaque modalité représente le nombre de résultats qui correspondent avec la modalité de la variable. Le diagramme à bande groupé permet aussi d'illustrer simultanément deux variables qualitatives.

```
ggplot(data = subset(apache_variables, !is.na(eyes))) + # Jeu de données sans valeur manquantes
  geom_bar(mapping = aes(x = eyes))
```

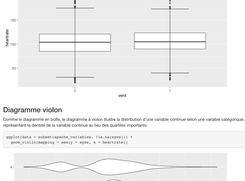

```
ggplot(data = subset(apache_variables, !is.na(eyes))) +
  geom_bar(mapping = aes(x = eyes, fill = vent),
           position = "dodge")
```



Histogramme

L'histogramme permet de représenter une distribution d'une variable continue. Pour obtenir une densité, nous devons le spécifier dans l'axe des ordonnées avec `after_stat(density)` pour que le calcule de densité se réalise. Si ce n'est pas spécifié, c'est plutôt les fréquences qui seront représentées. L'argument `bins=` permet de spécifier le nombre de bandes qui sera utilisé dans l'histogramme.

```
# Histogramme avec les fréquences
ggplot(data = apache_variables, mapping = aes(x = sodium)) +
  geom_histogram(color = 'black', fill = 'white')
```




```
# Histogramme avec la densité
ggplot(data = apache_variables, mapping = aes(x = sodium)) +
  geom_histogram(mapping = aes(y = after_stat(density)), color = 'black', fill = 'white', bins = 20)
```

Boite à moustache

La boite à moustache (boxplot) illustre les quartiles importants de la distribution d'une variable continue. La diagramme en boite est souvent utilisé pour représenter et comparer visuellement les statistiques de position selon une variable catégorique.

```
# boxplot pour une variable
ggplot(data = apache_variables, mapping = aes(x = heartrate)) +
  geom_boxplot() +
  stat_boxplot(geom = "errorbar", width = 0.4) # Ajout des bornes d'erreur aux extrémités du graphique
```


```
# boxplot groupé
ggplot(data = apache_variables, mapping = aes(y = heartrate, x = vent)) +
  geom_boxplot() +
  stat_boxplot(geom = "errorbar", width = 0.4)# Ajout des bornes d'erreur aux extrémités du graphique
```


Diagramme violon

Comme le diagramme en boîte, le diagramme à violon illustre la distribution d'une variable continue selon une variable catégorique, mais en représentant la densité de la variable continue au lieu des quartiles importants.

```
ggplot(data = subset(apache_variables, !is.na(eyes))) +
  geom_violin(mapping = aes(y = eyes, x = heartrate))
```

Alluvial plot

L'alluvial plot permet d'illustrer la stratification de plusieurs variables en termes de leur fréquence. Ce type de graphique n'est pas disponible directement dans la librairie `ggplot2`. La librairie `ggalluvial` sera nécessaire pour nous permettre de réaliser ce type de graphique. La vignette de la librairie, disponible ici, illustre plusieurs façons de réaliser un alluvial plot. En voici un exemple appliqué au jeu de données de la capsule.

```
# Importation de la librairie ggalluvial.
library(ggalluvial)

# Calcule d'effectif selon les groupes de variables.
groupe1 <- group_by(apache_variables, eyes, intubated, vent)
data_alluvial <- na.omit(tally(groupe1))

# Création du graphique.
ggplot(data = data_alluvial, mapping = aes(y = n, axis1 = vent, axis2 = intubated)) +
  geom_alluvium(aes(fill = eyes), width = 1/12) +
  geom_stratum(width = 1/12, fill = "darkgray", color = "white") +
  geom_label(stat = "stratum", aes(label = after_stat(stratum))) +
  scale_x_discrete(limits = c("vent", "intubated"), expand = c(.05, .05))
```


Cheatsheets

Des cheatsheets qui sont des aide-mémoires des fonctions de ≠ sont disponibles sur https://posit.co/resources/cheatsheets/. En particulier la cheatsheet *ggplot2* est d'une grande aide pour se rappeler de la syntaxe des graphiques.