GGPlot2
L'Essentiel

Visualisation Magnifique des Données dans R

Alboukadel KASSAMBARA

Edition 1
datanovia.com/en

GGPlot2 - L'Essentiel Pour une Visualisation Magnifique des Données dans R

Alboukadel KASSAMBARA

Publié par Datanovia (`https://www.datanovia.com/en`), Alboukadel Kassambara

Contact : Alboukadel Kassambara <alboukadel.kassambara@gmail.com>

Contents

Préface

0.1 Ce que vous apprendrez

GGPlot2 est un package R puissant et flexible, implémenté par Hadley Wickham, pour produire des graphiques élégants pièce par pièce.

ggplot2 est devenu un package populaire pour la visualisation de données. La documentation officielle du paquet est disponible à l'adresse suivante : `https://ggplot2.tidyverse.org/reference/`. Cependant, parcourir cette documentation complète peut "vous rendre fou" !

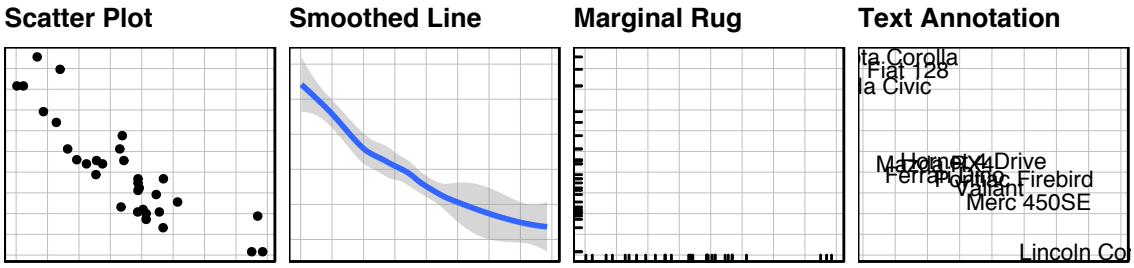
Ce livre présente l'essentiel de ggplot2 pour créer facilement de beaux graphiques dans R.

0.2 Principales caractéristiques de ce livre

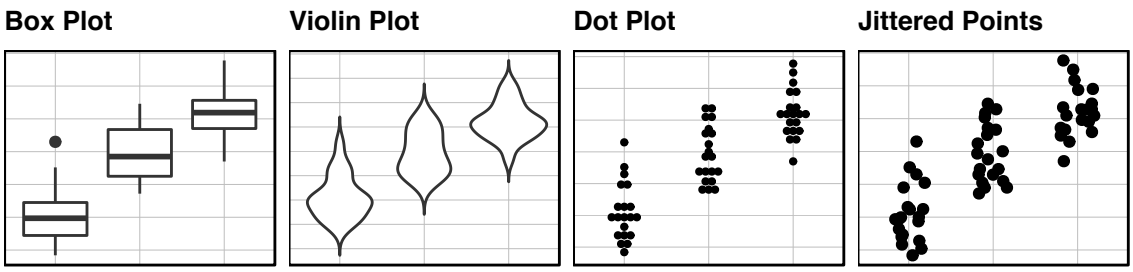
- Couvre les fonctions graphiques les plus importantes
- Chapitres courts et complets avec des exemples pratiques.

Quelques exemples de graphiques, décrits dans ce livre, sont présentés ci-dessous.

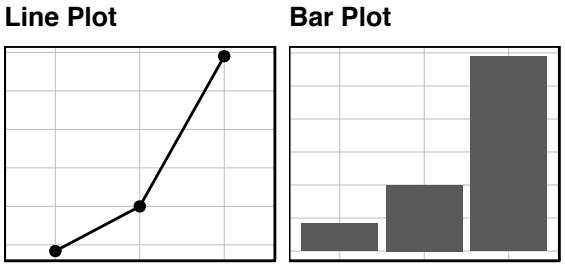
- Créer des **diagrammes de dispersion** pour afficher la relation entre deux variables continues x et y



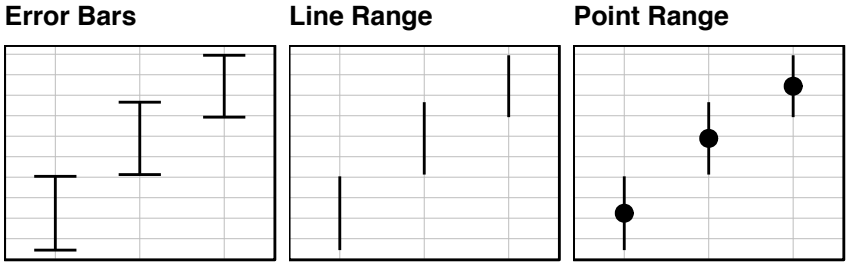
- Utilisation de Box plots et d'alternatives pour visualiser les données groupées en fonction d'une variable catégorielle



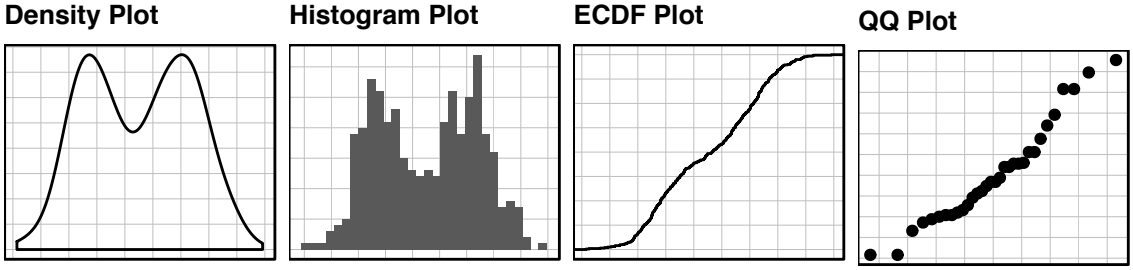
- Bar et Line Plots



- Visualisation des barres d'erreur



- Examen de la distribution d'une **variable continue** à l'aide de **diagrammes de densité**, **histogrammes** et alternatives



Vous apprendrez également comment combiner plusieurs ggplots en une seule figure.

0.3 Site du livre

The website for this book is located at : `https://www.datanovia.com/en/`. It contains number of resources.

0.4 Exécution des codes R à partir du PDF

Pour un code R en une seule ligne, vous pouvez simplement copier le code du PDF vers la console R.

Pour un code R sur plusieurs lignes, une erreur est parfois générée lorsque vous copiez et collez directement le code R depuis le PDF vers la console R. Si cela se produit, une solution consiste à:

- Collez d'abord le code dans votre éditeur de code R ou dans votre éditeur de texte
- Copiez le code de votre éditeur texte/code dans la console R

0.5 Colophon

Ce livre a été construit avec R 3.3.2 et les packages suivants :

```
##       name version                   source
## 1 bookdown   0.9.1 Github:rstudio/bookdown
## 2  cowplot   0.9.4                     CRAN
## 3    dplyr   0.8.3                     CRAN
## 4    dplyr   0.8.3                     CRAN
## 5  ggplot2   3.2.0                     CRAN
## 6   ggpubr   0.2.4                     CRAN
## 7  ggrepel   0.8.0                     CRAN
## 8    readr   1.3.1                     CRAN
```

0.6 Not found

The website for this book is located at : `https://www.datanovia.com/en/` It contains number of resources

A propos de l'auteur

Alboukadel Kassambara est docteur en bioinformatique et biologie du cancer. Il travaille depuis de nombreuses années sur l'analyse et la visualisation de données génomiques (pour en savoir plus : `http://www.alboukadel.com/`).

Il a une expertise forte dans l'identification des signatures de biomarqueurs pronostiques et prédictifs par l'analyse intégrative des données génomiques et cliniques à grande échelle.

Il est l'auteur de:

1) l'outil bioinformatique **GenomicScape** (www.genomicscape.com), un outil Web facile à utiliser pour l'analyse et la visualisation des données d'expression de gènes.
2) les sites Web **Datanovia** (`https://www.datanovia.com/en/`) et **STHDA** (`http://www.sthda.com/english/`), qui contiennent de nombreux cours et **tutoriels** sur l'exploration de données et les statistiques d'aides à la décision.
3) plusieurs packages **R** populaires pour l'analyse de données multivariées, l'analyse de survie, la visualisation de matrices de corrélation et la visualisation basique des données (`https://rpkgs.datanovia.com/`).
4) de nombreux **livres** sur l'analyse des données, la visualisation et l'apprentissage automatique (`https://www.datanovia.com/en/shop/`)

Chapter 1

Introduction à R

R est un logiciel statistique gratuit et puissant pour **analyser** et **visualiser** des données. Si vous voulez apprendre facilement l'essentiel de la programmation R, visitez notre série de tutoriels disponibles sur STHDA : `http://www.sthda.com/english/wiki/r-basics-quick-and-easy`.

Dans ce chapitre, nous fournissons une très brève introduction à **R**, pour installer R/RStudio ainsi que pour importer vos données dans R et installer les packages requis.

1.1 Installer R et RStudio

R et RStudio peuvent être installés sur les plates-formes Windows, MAC OSX et Linux. RStudio est un environnement de développement intégré pour R qui facilite l'utilisation de R. Il comprend une console, un éditeur de code et des outils de traçage.

1. R peut être téléchargé et installé à partir de la page Web du Comprehensive R Archive Network (CRAN) (`http://cran.r-project.org/`)
2. Après avoir installé le logiciel R, installez également le logiciel RStudio disponible sur : `http://www.rstudio.com/products/RStudio/`.
3. Lancez RStudio et commencez à utiliser R à l'intérieur de Rstudio.

1.2 Installer et charger les package R requis

Un package R est un ensemble de fonctionnalités qui étend les capacités de base de R. Par exemple, pour utiliser le code R fourni dans ce livre, vous devez installer les packages R suivants:

- `tidyverse`: collection de packages R partageant la même philosophie de programmation. Ces packages comprennent:
  - `readr`: pour importer des données dans R
  - `dplyr`: pour la manipulation des données
  - `ggplot2`: pour la visualisation des données.
- `ggpubr` package, qui facilite, pour les débutants, la création de graphiques prêt-à-publication.

1. **Installer le package tidyverse**. L'installation de tidyverse installera automatiquement readr, dplyr, ggplot2 et plus encore. Tapez le code suivant dans la console R:

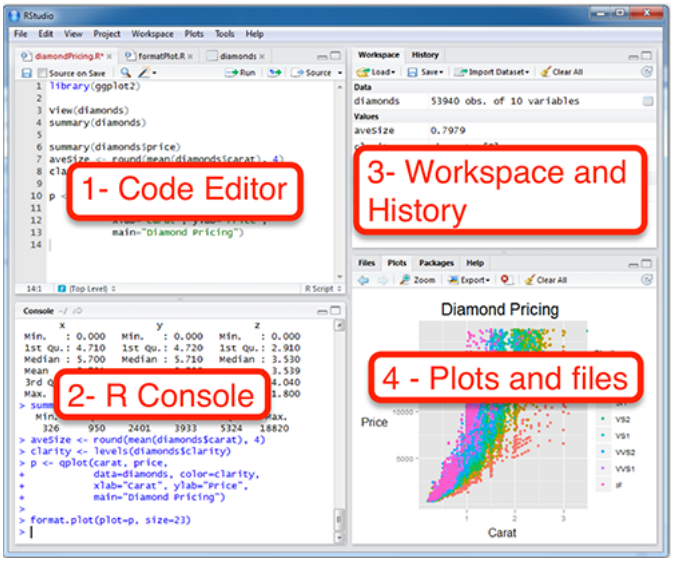


Figure 1.1: Interface Rstudio

```
install.packages("tidyverse")
```

2. **Installer le package ggpubr**.

```
install.packages("ggpubr")
```

3. **Charger les packages requis**. Après l'installation, vous devez d'abord charger le package pour utiliser les fonctions qu'il contient. La fonction `library()` est utilisée pour cette tâche. Une autre fonction est `require()`. Par exemple, pour charger les packages ggplot2 et ggpubr, tapez ceci:

```
library("ggplot2")
library("ggpubr")
```

Maintenant, nous pouvons utiliser des fonctions R, telles que *ggscatter*()[dans le package ggpubr] pour créer un nuage de points.

Si vous voulez de l'aide sur une fonction donnée, par exemple ggscatter(), tapez ceci dans la console R: `?ggscatter`.

1.3 Format des données

Vos données doivent être de format rectangulaire, où les colonnes sont des variables et les lignes des observations (individus ou échantillons).

- Les noms des colonnes doivent être compatibles avec les conventions d'appellation R. Évitez les colonnes avec des espaces vides et des caractères spéciaux. Bons noms de colonnes: `long_jump` or `long.jump`. Mauvais nom de colonne: `long jump`.
- Évitez de commencer les noms de colonnes par un nombre. Utilisez plutôt une lettre. Bons noms de colonnes: `sport_100m` or `x100m`. Mauvais nom de colonne: `100m`.
- Remplacer les valeurs manquantes par `NA` (pour non disponible)

Par exemple, vos données devraient ressembler à ceci:

```
  manufacturer model displ year cyl      trans drv
1         audi    a4   1.8 1999   4   auto(l5)   f
2         audi    a4   1.8 1999   4 manual(m5)   f
3         audi    a4   2.0 2008   4 manual(m6)   f
4         audi    a4   2.0 2008   4   auto(av)   f
```

Plus d'informations ici : [Meilleures pratiques pour la préparation des fichiers de données en vue de leur importation dans R] (`http://www.sthda.com/english/wiki/best-practices-in-preparing-data-files-for-importing-into-r`)

1.4 Importez vos données dans R

Tout d'abord, enregistrez vos données au format txt ou csv et importez-les comme suit (il vous sera demandé de choisir le fichier):

```
library("readr")

# Lecture des fichiers délimités par des tabulations (.txt tab)
my_data <- read_tsv(file.choose())

# Lit les fichiers délimités par des virgules (,) (.csv)
my_data <- read_csv(file.choose())

# Lit les fichiers delimités par des points-virgules (;) (.csv)
my_data <- read_csv2(file.choose())
```

Pour en savoir plus sur l'importation de données dans R, consultez le lien suivant : `http://www.sthda.com/english/wiki/importing-data-into-r`

1.5 Données de démonstration

R est livré avec plusieurs jeu de données de démonstration pour jouer avec les fonctions R. Les jeu de données de démo R les plus utilisés sont les suivants : **USArrests**, **iris** and **mtcars**. Pour charger un jeu de données de démonstration, utilisez la fonction **data**() comme suit. La fonction `head()` est utilisée pour inspecter les données.

```
data("iris")   # Chargement
head(iris, n = 3)  # Affichage des premières n = 3 lignes
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
```

Pour en savoir plus sur le jeu de données iris, tapez ceci:

```
?iris
```

Après avoir tapé le code R ci-dessus, vous verrez la description du jeu de données "iris" : ce jeu de données iris donne respectivement les mesures en centimètres des variables longueur et largeur des sépales, longueur et largeur des pétales, pour 50 fleurs de chacune des 3 espèces d'iris. Les espèces d'Iris sont setosa, versicolor et virginica.

1.6 Manipulation des données

Après avoir importé vos données dans R, vous pouvez facilement les manipuler à l'aide du `dplyr` package (**?**), which can be installed using the R code: `install.packages("dplyr")`.

Après avoir chargé dplyr, vous pouvez utiliser les fonctions R suivantes:

- `filter()`: Sélectionner des lignes (observations/échantillons) en fonction de leurs valeurs.
- `distinct()`: Supprimer les lignes en double.
- `arrange()`: Trier les lignes.
- `select()`: Sélectionner les colonnes (variables) par leur nom.
- `rename()`: Renommer les colonnes.
- `mutate()`: Ajouter/créer de nouvelles variables.
- `summarise()`: Calculer des statistiques descriptives (p. ex., calculer la moyenne ou la somme)
- `group_by()`: Analyser par sous-groupe de données.

Notez que le package dplyr permet d'utiliser l'opérateur d'enchaînement (%>%) pour combiner plusieurs opérations. Par exemple, x %>% f est équivalent à f(x). A l'aide de l'opérateur (%>%), la sortie de chaque opération est transmise à l'opération suivante. Ceci facilite la programmation de R.

Pour en savoir plus sur la manipulation des données, cliquez sur ce lien : `https://www.datanovia.com/en/courses/data-manipulation-in-r/`

1.7 Fermez votre session R/RStudio

Chaque fois que vous fermez R/RStudio, il vous sera demandé si vous souhaitez sauvegarder les données de votre session R. Si vous décidez de sauvegarder, les données seront disponibles dans les prochaines sessions R.

Chapter 2

Introduction à GGPlot2

2.1 Qu'est-ce que ggplot2

GGPlot2 est un package R puissant et flexible, implémenté par Hadley Wickham, pour produire des graphiques élégants pièce par pièce (Wickham et al., 2019).

Le **gg** dans ggplot2 signifie *Grammaire du Graphique*, un concept graphique qui décrit les graphes en utilisant une "grammaire". Selon le concept ggplot2, un graphique peut être divisé en différentes parties fondamentales : **Graphique = données + Esthétique + Géométrie** (ou en anglais **Plot = data + Aesthetics + Geometry**)

- **data** : data frame
- **esthétique** (ou **aesthetics** en anglais) : permet d'indiquer les variables **x** et **y**. Il peut également être utilisé pour contrôler la **couleur**, la **taille** et la **forme** des points, etc…
- **géométrie** : correspond au type de graphique (histogramme, box plot, line plot, …..)

La syntaxe de ggplot2 peut sembler opaque pour les débutants, mais une fois que vous comprenez les bases, vous pouvez créer et personnaliser tous les types de graphiques que vous voulez.

Notez que, pour réduire cette opacité, nous avons récemment créé un package R, nommé **ggpubr** (ggplot2 Based Publication Ready Plots), pour rendre ggplot plus simple pour les étudiants et chercheurs ayant des connaissances en programmation non avancées.

2.2 Fonctions clés

Après avoir installé et chargé le package ggplot2, vous pouvez utiliser les fonctions clés suivantes:

Types de graphes	Fonctions GGPlot2
Initialiser un ggplot	ggplot()
Nuage de points	geom_point()
Box plot	geom_boxplot()
Violon plot	geom_violin()
Strip chart	geom_jitter()
Dot plot	geom_dotplot()

Types de graphes	Fonctions GGPlot2
Bar plot	geom_bar() ou geom_col()
Line plot	geom_line()
Histogramme	geom_histogram()
Graphique de densité	geom_density()
Barres d'erreur	geom_errorbar()
QQ plot	stat_qq()
ECDF plot	stat_ecdf()
Titres et libellés des axes	labs()

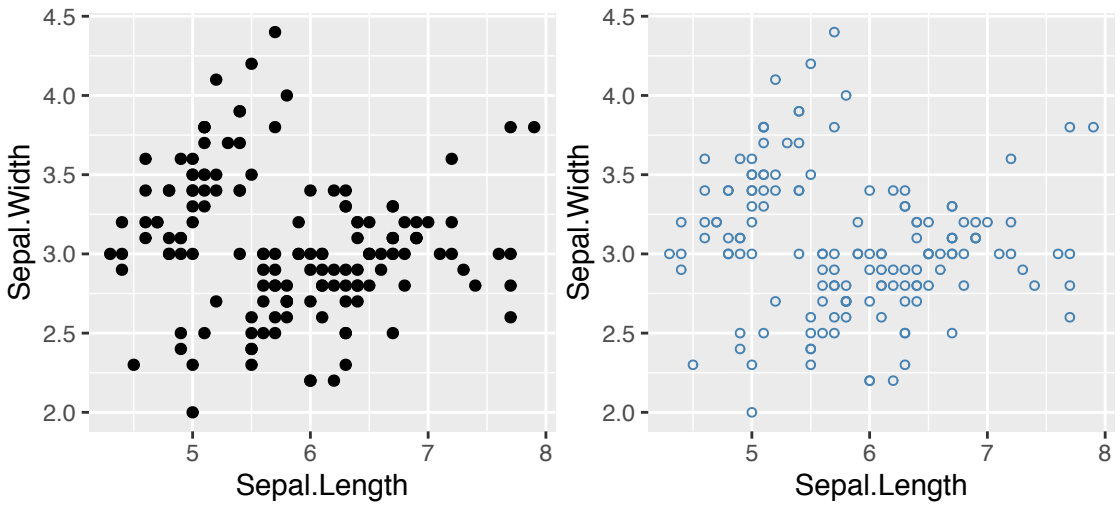
2.3 Exemple de graphiques

La fonction principale du package ggplot2 est `ggplot()`, qui peut être utilisée pour initialiser le système graphique avec des données et les variables x/y.

Par exemple, le code R suivant prend le jeu de données `iris` pour initialiser le ggplot et ensuite ajoute une couche (`geom_point()`) pour créer un diagramme de dispersion de `x = Sepal.Length` en fonction de `y = Sepal.Width`:

```
library(ggplot2)
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width))+
  geom_point()

# Modifier la taille, la couleur et la forme des points
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width))+
  geom_point(size = 1.2, color = "steelblue", shape = 21)
```



Notez que, dans le code ci-dessus, la forme des points est spécifiée avec un chiffre. Les différentes formes de points disponibles dans R, sont: