

Le travail de Xueting Yin **est disponible ici**

BENOUCIEF Amine

22/12/2020

```
knitr::opts_chunk$set(echo = TRUE)
```

SYNTHESE DU TRAVAIL EN QUESTION

ggplot2 est un package de data visualisation qui améliore la qualité et l'esthétique des graphiques. C'est un bon travail qui résume bien les fonctionnalités du package. Xueting a su bien expliquer ses lignes de codes. Cependant d'autres type de visuel aurait pu être intégré au code pour mettre en valeur la richesse et la variété de ggplot2. De même, pas de bibliographie nous ne savons pas si elle s'est inspirée d'un travail ou elle a créé elle-même ses exemples (si oui il fallait le préciser).

Introduction

Ggplot2 est un package dans le langage R qui est utilisé pour la visualisation des données. Grâce à son dessin plus coloré, il attire plus l'attention des gens et est plus populaire auprès des chercheurs scientifiques.

Gg signifie grammaire graphique, -plot signifie dessin, il est similaire à la grammaire du langage naturel, il a de différents attributs, tels que le titre, le canevas, l'axe des coordonnées, etc., ces attributs constituent ensemble la grammaire graphique.

Avant de l'utiliser, il faut installer un package soit tidyverse, soit ggplot2, tidyverse est un système d'exploitation de packages de données cohérent, y compris le package ggplot2.

```
#install.packages("ggplot2")
```

Après l'installation, nous chargeons le package : `library(ggplot2)`

Ensuite, nous importons des données, il existe trois méthodes: utiliser les données existantes, utiliser les données externes, et entrer les données.

Dessiner un nuage de point

Dans ce minituto, nous allons vous présenter comment dessiner un nuage de points. Nous utiliserons les données mpg existantes, mpg sont des données sur la consommation de carburant de 1999 à 2008 pour 38 modèles de voitures populaires, il y a 234 lignes et 11 variables, et les attributs de données peuvent être visualisés via `?mpg`.

Nous nous concentrerons sur quatre variables: `displ`(cylindrée du moteur, en litres) `drv`(le type de transmission) `hwy`(kilomètres d'autoroute par gallon) `class`("type" de voiture).

Maintenant, nous voulons savoir: quel est le rapport entre `displ` et `hwy`? nous utilisons les codes ci-dessous:

```
library(ggplot2) ggplot(data = mpg) + # utiliser les données mpg ggplot() est créer une nouvelle toile aes(x = displ, y = hwy) + # définir l'axe de x et l'axe de y geom_point() # dessiner des points, si on veut changer la couleur du point, on entre geom_point(color = "blue")
```

Introduisons une nouvelle variable "class", nous voulons savoir quelle est la relation entre le type de voiture et les deux autres variables?

```
library(ggplot2) ggplot(mpg) + aes(x = displ, y = hwy) + geom_point(aes(color = class))
```

Pourquoi pouvions-nous juste écrire `color = "bleu"`, mais ici il faut rajouter `aes()`?

Parce que le bleu est une couleur qui existe dans la nature et que `drv` est une variable, si nous rajoutons une variable, il faut utiliser `aes()`, soit `aesthesis` qui est une fonction spécialement utilisée pour les opérations d'affectation.

Si vous n'aimez pas les couleurs par défaut, vous pouvez utiliser la palette de couleurs. Soit `scale_color_brewer` (`palette = "Set1"`), vous pouvez chercher sur google les différents palettes.

Les paramètres de `geom_point()` peuvent être modifiés, nous pouvons utiliser `?geom_point()` pour voir tous les paramètres.

Par exemple:

```
geom_point(shape = 1, color = "black", fill = "white", size = 5, stroke = 5)
```

`#shape` est pour changer la forme des points, `color` est la couleur du contour, `fill` est la couleur de remplissage, `size` est la #taille des points, `stroke` est le diamètre intérieur des points

Si nous ne voulons pas rajouter la variable par couleur, nous pouvons également utiliser la taille du point pour représenter différentes voitures : `geom_point(aes(size = class))`, Mais attention, `r` nous incitera à ne pas utiliser des dimensions pour représenter des variables continues.

Il est également inapproprié d'utiliser différentes formes pour représenter différentes voitures, car `r` gère jusqu'à six formes et supprime les valeurs qui ne sont pas affectées à la forme, le code est comme celui-ci: `geom_point(aes(shape = class))`.

Overplotting

Lorsque la quantité de données est importante, de nombreux points se chevauchent.

Quelques solutions:

1. Remplacer `geom_point()` en `geom_count()` # Plus il y a de points qui se chevauchent, plus le point est grand.
2. Changer `geom_point()` en `geom_bin2d()` # Plus il y a de points coïncidents, plus la couleur est claire et représentée par un carré.
3. Ajuster la transparence, utilisez `alpha =`, ex: `geom_point(alpha = 0.3)` # Plus le point est sombre, plus il y a de chevauchement.
4. Faire une courbe d'ajustement, utilisez: `geom_smooth()` # Plus il y a de chevauchement, plus l'écart type est petit.

EVALUATION DU TRAVAIL EN QUESTION

Critère 1 : Visuel sur pdf 3/4 Tres agreable a lire.

Critère 2 : Originalite du code 3/4 Utilisation de fonctions essentiels mais reste simple.

Critère 3 : Fonctionnalité du code 4/4 Le code ne fonctionne.

Critère 4 : Lisibilité du code 3/4 très lisible.

Critère 5 : Explications données 3/4 De bonnes explications pour toutes les fonctions. Rien a dire

CONCLUSION

Globalement c'est un bon travail avec des explications clair et precise. L'integration d'autres type de graphiques aurait rendu le travail excellent.