VISUALISATION DES DONNEES AVEC GGPLOT2

Redigé par:Inoussa ZAONGO

1er Mai 2023

plan

- Introduction
- generalité
- principe de base de ggplot2
- principale fonction de ggplot2
- quelques fonctions de type geom
- Démarche de création d'un graphique
- quelques exemples
- representation de plusieurs geom
- mappage
- scale
- theme
- faceting
- packages souvent utilisé avec ggplot2
- carte geographique
- comparaison entre ggplot2 et et le système graphique R

Introduction

Le package ggplot2 a été publié pour la première fois en 2006 sous le nom de ggplot. Il fut amélioré de façon importante et renommé ggplot2 en 2007. Son créateur est Hadley Wickham, qui est aussi derrière plusieurs des packages du tidyverse, duquel ggplot2 fait partie. Le package est maintenant développé par toute une équipe, dont des employés de RStudio. Il implémente la grammaire graphique présentée dans : « Wilkinson, L. (2005). The grammar of graphics, 2e édition. Springer ».Le package ggplot2 a été conçu en ayant comme objectif la simplicité d'utilisation et la qualité des graphiques produits. Il reprend les forces suivantes des systèmes graphiques R précédents :

- système de base : création de graphiques par **couches** (ajouts séquentiels d'éléments);
- package lattice : **représentations multivariées simples**.

Introduction

tout en apportant les améliorations suivantes :

- **esthétique** par défaut pensée de façon à transmettre plus efficacement les informations contenues dans le graphique.
- **automatisation de certaines configurations** graphiques, notamment les légendes;
- **ajout de transformations statistiques** communes (p. ex. courbes de lissage, barres d'erreur) facilité.

Dans le package ggplot2, tout a été repensé pour être plus simple d'utilisation et surtout pour que le graphique produit transmette plus efficacement l'information qu'il contient.

L'utilisation du package ggplot2 est présentée ici de façon brève, mais avec tout de même assez de détails pour démarrer un apprentissage de cet outil aux possibilités vastes.

I.Généralité

1.Installation du package ggplot2

Comme tout package,il faut commencer par l'importer.pour cela il y'a plusieurs solutions:

- Utiliser l'onglet package sous R studio puis le sous onglet install.Une fois installé vous devez charger la library ggplot2 comme ci dessous.

```
library(ggplot2)
```

- installer et charger le package tidyverse car ggplot2 y est inclu.
- Par code

```
install.packages("gglot2") #installé le package ggplot2

library(ggplot2) #charger la library ggplot2
```

2.Présentation des données utilisées pour les exemples

Comme dans les autres notes sur les graphiques en R, les données quakes auxquelles deux facteurs sont ajoutés sont utilisées dans les exemples de cette fiche.Elle contient des informations sur les tremblements de terre qui ont été enregistrés entre 1964 et 1974 sur la planète Terre.Cette base compte 1000 observations pour 7 variables.

```
data("quakes")
quakes$mag_catego <- factor(floor(quakes$mag))
quakes$region <- factor(
  ifelse(quakes$long < 175, yes = "Ouest", no = "Est"),
  levels = c("Ouest", "Est")
)
str(quakes)
```

```
## 'data.frame':    1000 obs. of  7 variables:
##  $ lat        : num  -20.4 -20.6 -26 -18 -20.4 ...
##  $ long       : num  182 181 184 182 182
```

II.principe de base de ggplot2

Le principe de base derrière la grammaire graphique (d'où le gg dans ggplot) est qu'un graphique statistique est une représentation de données, dans un système de coordonnées spécifique, divisée en éléments de base :

- éléments géométriques (geoms) : points, lignes, barres, etc.; propriétés visuelles (aesthetics) des éléments géométriques : axes, couleurs, formes, tailles, etc.
- transformations statistiques, si désiré : courbe de régression ou de lissage, région d'erreur, etc.

Un graphique est spécifié en associant des variables, provenant des données, à des propriétés visuelles des éléments géométriques du graphique.

III.principale fonction de ggplot2

Les principales fonctions du package ggplot2 sont les suivantes :

- ggplot : initialisation d'un objet de classe ggplot;
- qplot : initialisation rapide (q pour quick) d'un objet ggplot; + : opérateur pour l'ajout de couches ou la modification de configurations dans un objet ggplot;
- fonctions de type geom_* (p. ex. geom_point, geom_boxplot, geom_bar, etc.) : spécification de couches à ajouter à un graphique;
- aes : création d'un mapping, soit une association entre des propriétés visuelles et des variables;
- ggsave : enregistrement d'un graphique.

Notons que l'utilisation de la fonction qplot est plus intuitive que celle de ggplot pour des gens familiers avec plot. La fonction qplot n'offre cependant pas toutes les possibilités de ggplot. Elle ne sera pas couverte dans ce document.

IV.quelques fonctions de type geom

- geom_point
- geom_line
- geom_bar
- geom_histogram
- geom_boxplot
- geom_density
- geom_qq

La liste est longue; voir http://ggplot2.tidyverse.org/reference/

V.Démarche de création d'un graphique

1.Suggestion Démarche

Voici une suggestion de démarche à suivre lors de la création d'une graphique ggplot2.
1-Planifier le travail - Répondre aux questions suivantes :
a-Je veux représenter quelles variables, de quel jeu de données ?
b-Je veux créer quel type de graphique ?
c-Quelle fonction de type geom_* me permettra de produire les éléments géométriques de ce graphique ?
d-Cette fonction accepte quelles propriétés visuelles ?
e.Je veux associer les variables concernées à quelles propriétés visuelles ?
2-Écrire et soumettre la première version du code de création du graphique.
3-Modifier le code de création du graphique pour ajuster la mise en forme (titre, nom d'axes, autres annotations, palette de couleur, etc.) selon mes besoins. - travail itératif : cycle « modification code » $\rightarrow$
« jugement du graphique créé » répété jusqu'à l'obtention d'un résultat satisfaisant.

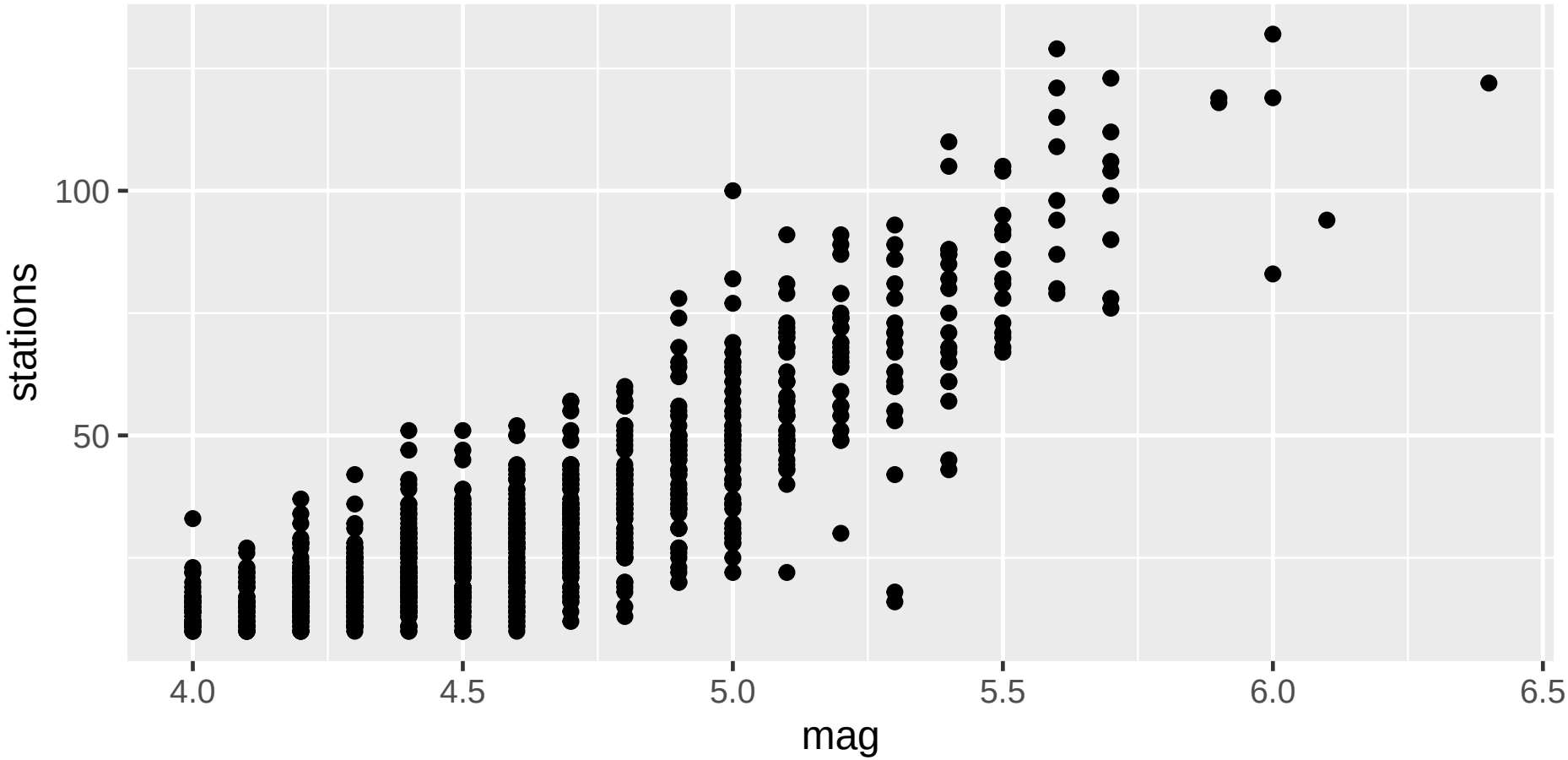
V.Démarche de création d'un graphique

2.Application

Nous voulons representer le nuage de points de la magnetude en fonction des stations.

```
mon_graph1 <- ggplot(data = quakes)
mon_graph <- mon_graph1 + geom_point(mapping = aes(x = mag, y
```

V.Démarche de création d'un graphique



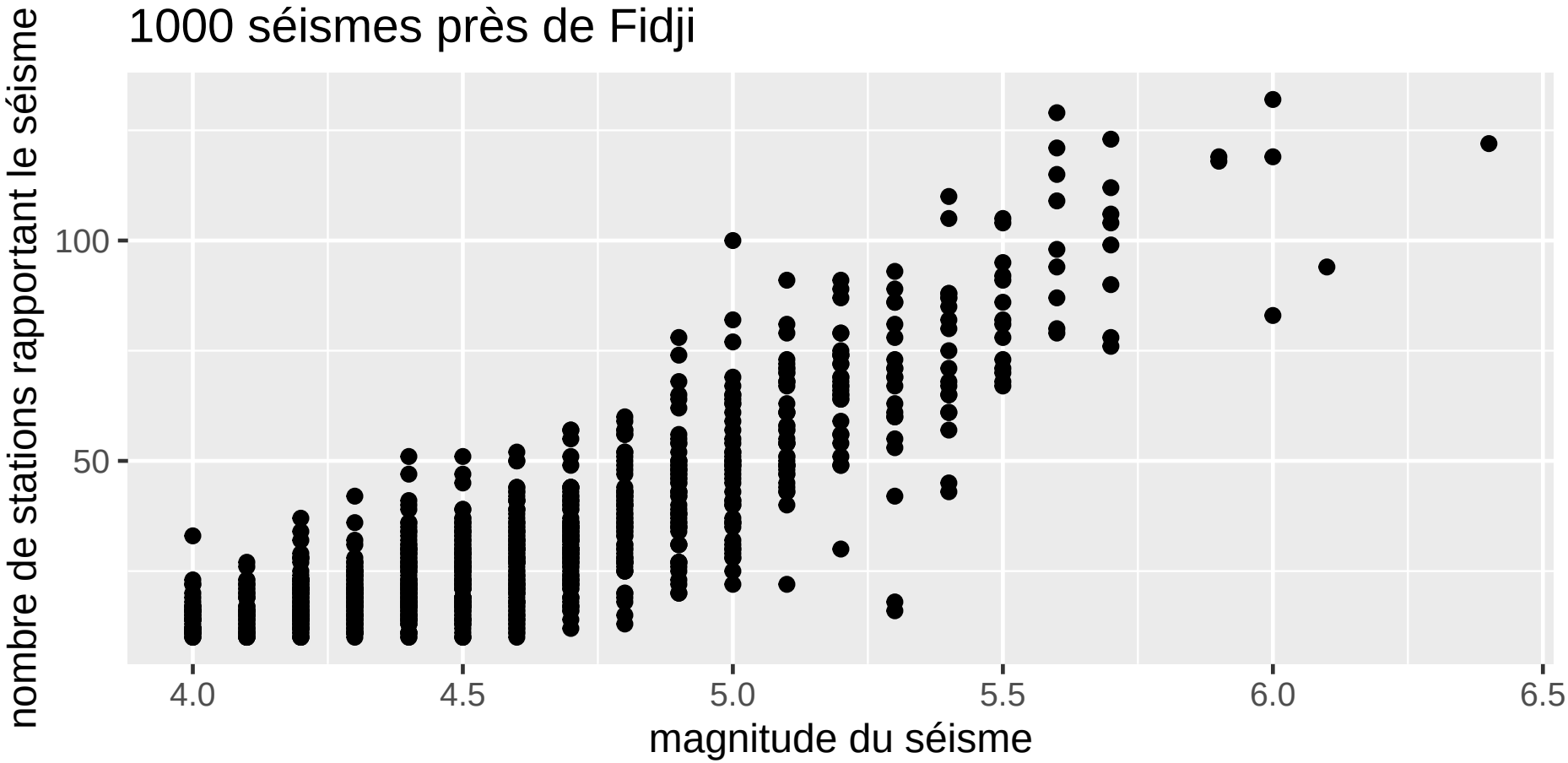
V.Démarche de création d'un graphique

- Ajout d'un titre et de noms d'axes - fonction labs.

Cette fois nous allons ajouter un titre et des noms d'axes parlants - fonction labs

```
graph2<-ggplot(data = quakes) +
  geom_point(mapping = aes(x = mag, y = stations)) +
  labs(
    title = "1000 séismes près de Fidji",          # ou ggtit
    x = "magnitude du séisme",                     # ou xlab(
    y = "nombre de stations rapportant le séisme"  # ou ylab(
  )
```

V.Démarche de création d'un graphique



V.Démarche de création d'un graphique

Vous avez envie de centrer le titre du graphique ? Voici comment faire

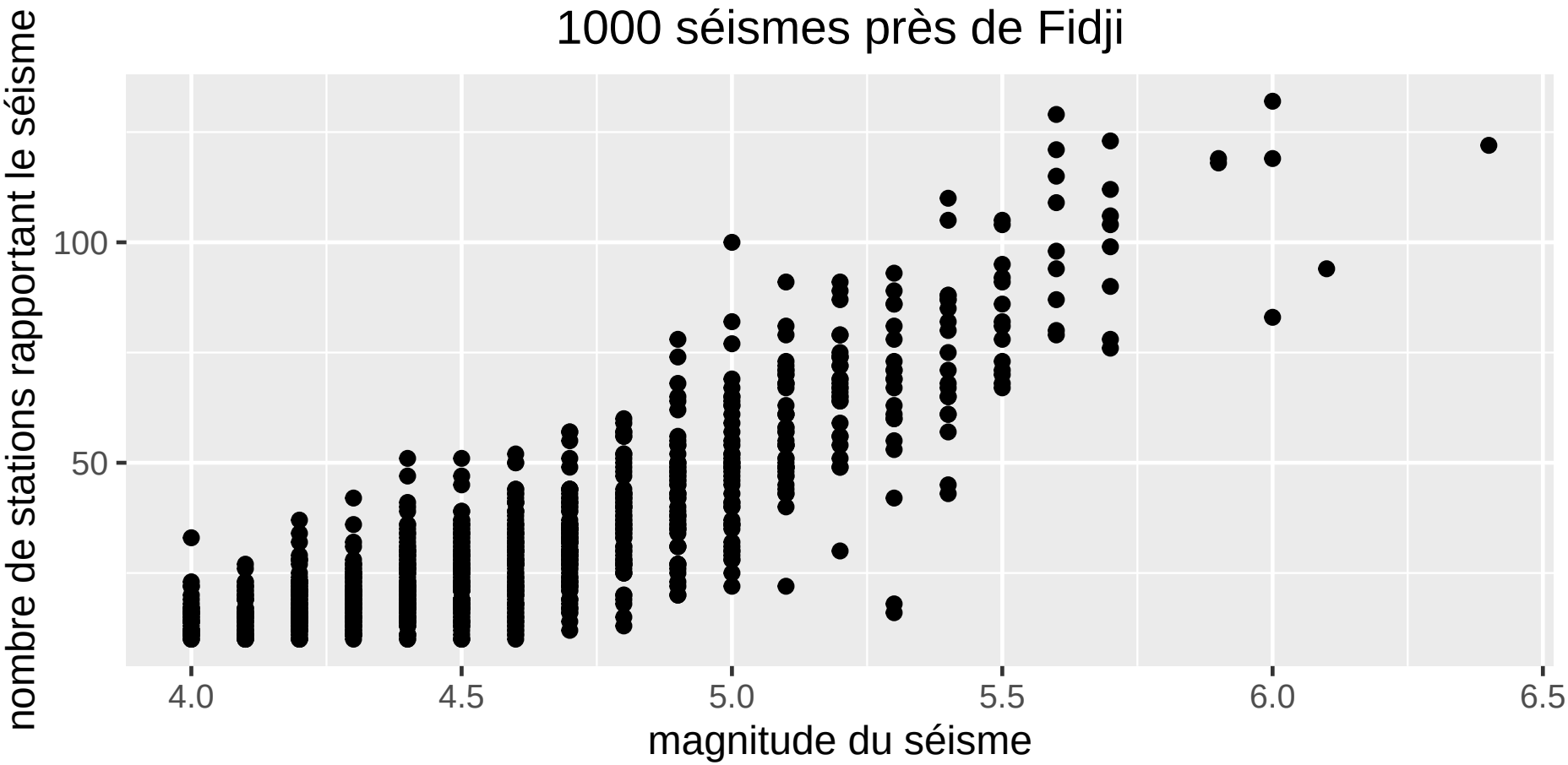
```
graph3<-ggplot(data = quakes) +
  geom_point(mapping = aes(x = mag, y = stations)) +
  labs(
    title = "1000 séismes près de Fidji",
    x = "magnitude du séisme",
    y = "nombre de stations rapportant le séisme"
  ) +
  theme(plot.title = element_text(hjust = 0.5))  # permet de
```

V.Démarche de création d'un graphique

- Ajout d'une variable associée à une propriété visuelle autre qu'un axe.

Le graphique précédent représente deux variables. Ajoutons une troisième variable au graphique, qui fera varier la couleur des points. Si la palette de couleur utilisée par défaut ne nous plaît pas, nous pouvons la changer.

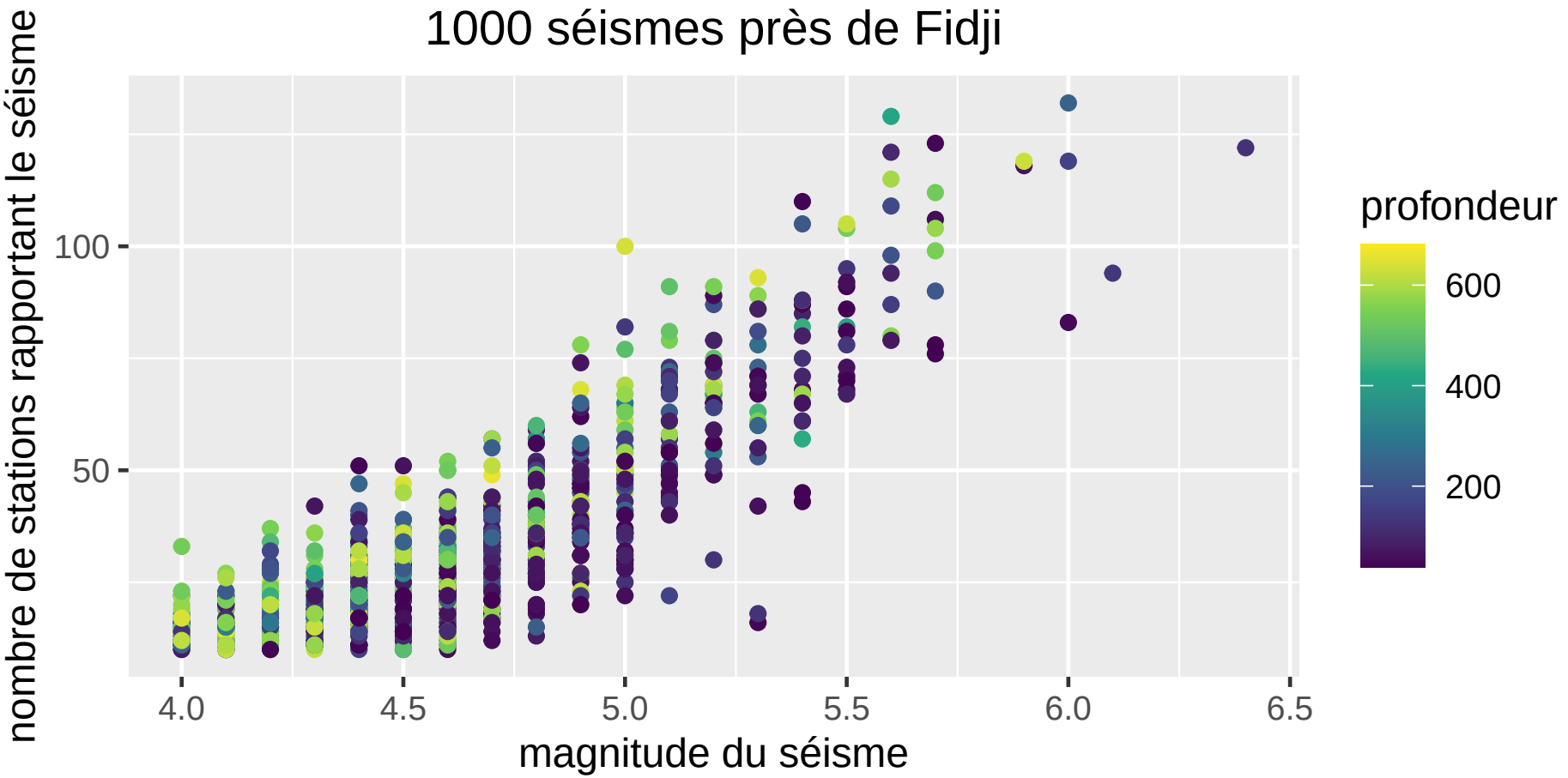
V.Démarche de création d'un graphique



V.Démarche de création d'un graphique

```
scatterplot <- ggplot(data = quakes) +
  geom_point(mapping = aes(
    x = mag,
    y = stations,
    colour = depth  # permet de faire varier la couleur des po
  )) +
  labs(
    title = "1000 séismes près de Fidji",
    x = "magnitude du séisme",
    y = "nombre de stations rapportant le séisme",
    colour = "profondeur"  # permet de modifier le titre de le
  ) +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_colour_viridis_c()       # permet d'utiliser la palet
```

V.Démarche de création d'un graphique



V.Démarche de création d'un graphique

- Echelles de couleurs - fonctions de type scale-colour-* et scale-fill-*

Deux propriétés visuelles (aesthetics) permettent de spécifier des couleurs :

- colour = couleur de points et de lignes,
- fill = couleur de remplissage.

Les fonctions permettant de contrôler les palettes de couleurs utilisées pour ses propriétés visuelles ont toutes un nom débutant par scale_* suivi du nom de la propriété visuelle concernée et de _*. Le nom de la fonction peut se terminer par :

- viridis_d ou viridis_c : utiliser une palette de couleur offerte dans le package R viridisLite, soit discrète (_d) ou continue (_c);
- brewer ou distiller : utiliser une palette de couleur de ColorBrewer;
- grey : utiliser un dégradé de gris,
- manual : utiliser une palette discrète spécifiée manuellement;
- gradient, gradient2 ou gradientn : utiliser dégradé continu spécifié

VI.quelques exemples

1.Diagramme en barres: fonctions geom-bar et geom-col

La fonction geom_bar permet de produire un graphique en barre (barplot). On lui passe en x la variable qualitative dont on souhaite représenter l'effectif de chaque modalité.
En effet,la fonction geom_bar calcule les fréquences des niveaux de facteurs et produit des diagrammes à barres. Avec cette fonction, pas besoin d'utiliser table ou une aute fonction similaire pour calculer les fréquences.

- SYNTAXE

```
ggplot(data = my_data, aes(x = my_variable, y = my_value)) +
  geom_bar(stat = "identity")
```

VI.quelques exemples

2.Diagramme en secteurs: coordonnées polaires avec coord-polar

Les auteurs de ggplot2 n'offrent pas de fonction conviviale pour la création de diagrammes en secteurs, probablement parce qu'ils ne recommandent pas leur utilisation. Malgré tout, tenter de tracer un diagramme en secteurs avec ggplot2 aide à comprendre davantage les possibilités du package. Pour produire un diagramme en secteurs avec ggplot2, il faut d'abord produire un diagramme à barres empilées, puis demander l'utilisation d'un système de coordonnées polaires par un appel à la fonction coord_polar.

VI.quelques exemples

3.Courbes de densité :fonction geom-density

geom_density permet d'afficher l'estimation de densité d'une variable numérique. Son usage est similaire à celui de geom_histogram.
Ainsi, si on veut afficher la densité de la répartition de la part des cadres dans les communes de notre jeu de données :

- SYNTAXE

```
ggplot(rp) + geom_density(aes(x = cadres))
```

VI.quelques exemples

4.Diagrammes en violons : fonction geom-violin

Un autre outil, dérivé des courbes de densités à noyau, permettant d'explorer l'association potentielle entre une variable numérique et une variable catégorique est le diagramme en violon (violin plot).
Il faut noter aussi que geom_violin est très semblable à geom_boxplot, mais utilise des graphes en violon à la place des boîtes à moustache.sommes toutes on peut dire que les graphiques geom-violon sont dérivés des courbes de densité et des boites à moustache.on peut alors supperposer plusieurs graphiques geom_violon comme les boxplots aussi.

```
ggplot(rp) + geom_violin(aes(x = departement, y = maison))
```

VI.quelques exemples

5.Diagrammes en boîtes : fonction geom-boxplot

geom_boxplot permet de représenter des boîtes à moustaches. On lui passe en y la variable numérique dont on veut étudier la répartition, et en x la variable qualitative contenant les classes qu'on souhaite comparer.

```
# Graphique en boîtes de l'âge par rapport à la région
ggplot(donnees, aes(x = region, y = age)) +
  geom_boxplot()
```

VI.quelques exemples

6.La fonction geom-text

geom_text permet d'afficher des étiquettes de texte. On doit lui fournir trois paramètres dans aes : x et y pour la position des étiquettes, et label pour leur texte.
Par exemple, si on souhaite représenter le nuage croisant la part des diplômés du supérieur et la part de cadres, mais en affichant le nom de la commune (variable commune) plutôt qu'un simple point, on peut faire

```
data(rp)
ggplot(rp) +
  geom_text(
    aes(x = dipl_sup, y = cadres, label = commune)
```

VI.quelques exemples

6.La fonction geom-label

geom_label est identique à geom_text, mais avec une présentation un peu différente.

7.La fonction geom-line

geom_line trace des lignes connectant les différentes observations entre elles. Il est notamment utilisé pour la représentation de séries temporelles. On passe à geom_line deux paramètres : x et y. Les observations sont alors connectées selon l'ordre des valeurs passées en x.
Comme il n'y a pas de données adaptées pour ce type de représentation dans notre jeu de données d'exemple, on va utiliser ici le jeu de données economics inclus dans ggplot2 et représenter l'évolution du taux de chômage aux États-Unis (variable unemploy) dans le temps (variable date) :

```
ggplot(economics) + geom_line(aes(x = date, y = unemploy))
```

VI.quelques exemples

8.La fonction geom-hex et geom-bin2d

Lorsque le nombre de points est important, la représentation sous forme de nuage peut vite devenir illisible : la superposition des données empêche de voir précisément leur répartition.

- geom_bin2d, crée une grille sur toute la zone du graphique et colorier chaque carré selon le nombre de points qu'il contient.
- geom_hex, quant à elle crée une grille constituée d'hexagones.

VI.quelques exemples

9.Nuage de point:La fonction geom-point

La fonction geom_point permet de créer un nuage de point et a d'autre fonctionalité telle que ajouté des points à un graphique.

- SYNTAXE

```
# Nuage de points de l'âge par rapport au revenu
ggplot(donnees, aes(x = revenu, y = age)) +
  geom_point()
```

VII.representation de plusieurs geom

On peut représenter plusieurs geom simultanément sur un même graphique, il suffit de les ajouter à tour de rôle avec l'opérateur $+$.

par exemple combiner un boxplot et un ensemble de point pour mieux voir la concentration des individus.pour cela on utilise la fonction geom_boxplot et la fonction geom_point.

Autre exemple, on peut vouloir ajouter à un nuage de points une ligne de régression linéaire à l'aide de geom_smooth.

VIII.Mapping

Un mappage, dans ggplot2, est une mise en relation entre un attribut graphique du geom (position, couleur, taille...) et une variable du tableau de données.

Ces mappages sont passés aux différents geom via la fonction aes() (abbréviation d'aesthetic). Les arguments de cette fonction spécifient comment les données doivent être représentées graphiquement.

IX.Scales

On a vu qu'avec ggplot2 on définit des mappages entre des attributs graphiques (position, taille, couleur, etc.) et des variables d'un tableau de données. Ces mappages sont définis, pour chaque geom, via la fonction aes().

Les scales dans ggplot2 permettent de modifier la manière dont un attribut graphique va être relié aux valeurs d'une variable, et dont la légende correspondante va être affichée. Par exemple, pour l'attribut color, on pourra définir la palette de couleur utilisée. Pour size, les tailles minimales et maximales, etc.

Pour modifier une scale existante, on ajoute un nouvel élément à notre objet ggplot2 avec l'opérateur $+$. Cet élément prend la forme scale___.

X.Thèmes

Les thèmes permettent de contrôler l'affichage de tous les éléments du graphique qui ne sont pas reliés aux données : titres, grilles, fonds, etc.

Il existe un certain nombre de thèmes préexistants à savoir Le thème "Excel","theme_minimal" ,le"theme_classic",le "theme_bw".... Par exemple le thème theme_bw qui propose un style noir et blanc qui convient bien pour des graphiques avec beaucoup de détails.

XI.faceting

Le faceting permet d'effectuer plusieurs fois le même graphique selon les valeurs d'une ou plusieurs variables qualitatives.

Son but est de permettre les comparaisons entre variable.

XII.Packages souvent utilisés avec ggplot2

Le package ggplot2 est souvent utilisé conjointement à d'autres packages du tidyverse, notamment :

- dplyr pour la manipulation / le prétraitement de données;
- magrittr qui offre l'opérateur « pipe » %>%;
- forcats pour la manipulation de facteurs.

XIII.Cartes géographiques

Comme avec le système graphique de base, le package maps permet d'ajouter une carte en arrière-plan d'un graphique ggplot.

NB:L'extension ggmap permet également d'intégrer une carte Google Maps (requière un compte sur https://cloud.google.com/maps-platform/ pour avoir accès aux cartes) ou Stamen Maps à un graphique produit avec ggplot2.

XIV.Comparaison entre ggplot2 et le système graphique R de

Lorsque le nombre de variables à représenter dans un graphique est grand (Number of dimensions élevé), utiliser ggplot2 peut potentiellement permettre de sauver beaucoup de temps. Les représentations multivariées sont plus faciles à produire avec ggplot2 qu'avec le système graphique R de base. En grande partie pour cette raison, la production de graphiques pour de l'analyse exploratoire de données est souvent plus rapide à réaliser avec ggplot2 qu'avec le système graphique de base. Malgré tout, il existe encore une situation dans laquelle le système de base surpasse ggplot2 : la production de graphiques à mises en formes spécifiques pour des publications scientifiques. Les configurations graphiques sont souvent plus simples à réaliser avec le système de base qu'avec ggplot2.

Conclusion

Il ressort que, la bibliothèque ggplot2 est un outil puissant pour la visualisation de données en R. Elle offre une grande variété de graphiques et une flexibilité dans la personnalisation de ces graphiques. Avec ggplot2, vous pouvez facilement créer des graphiques attrayants et informatifs à partir de données brutes.