

La visualisation de données avec ggplot2

LA VISUALISATION DES DONNÉES ET LES TABLEAUX DE BORD

uses non-data ink. Design elements!

The plotting space you are using

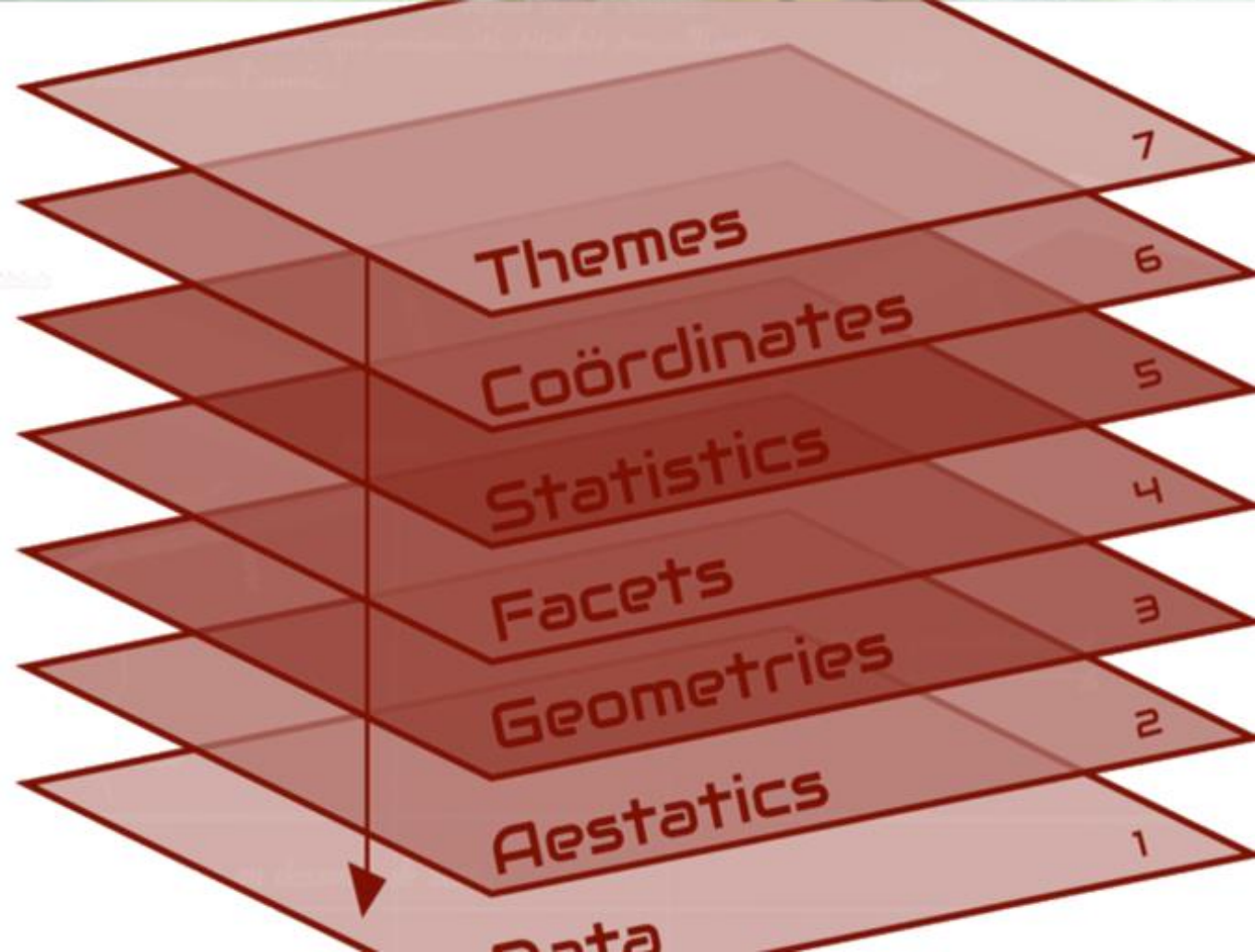
Statistical models & summaries

Rows and columns of sub-plots

Shapes used to represent your data

scales on which the data is mapped

The actual variables to be plotted



10. La grammaire des graphiques

La grammaire des graphiques

C'est une chose de reconnaître quand les graphiques sont **efficaces** et que leur esthétique les rend **faciles à lire**, et quand ils sont disposés dans un tableau de bord qui raconte une histoire visuelle **convaincante** (et quand ils ne le sont pas).

C'est une toute autre chose d'apprendre à **construire** de tels graphiques.

La **grammaire des graphiques** [Wilkinson, 1999 ; Wickham, 2009] fournit une voie fiable pour y parvenir.

La grammaire des graphiques

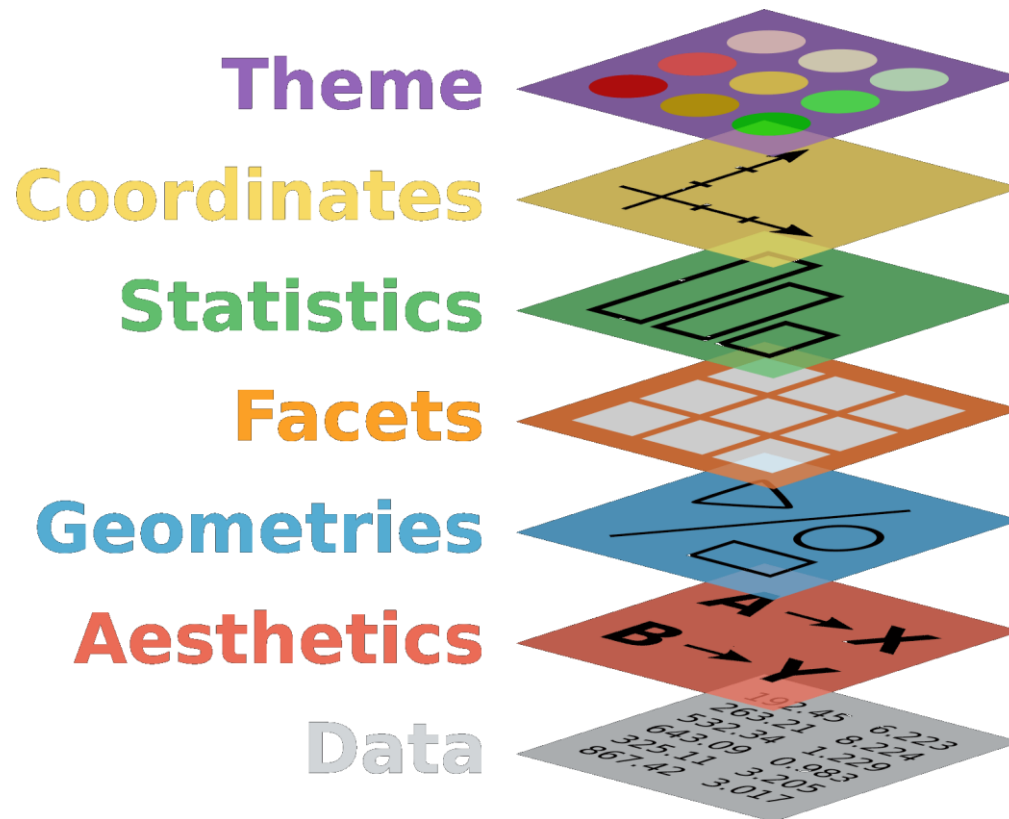
"Une **grammaire** est un ensemble de règles structurelles qui permet de définir et d'établir les **composantes d'une langue**.

Le système/structure d'une langue se compose généralement de la **syntaxe** (règles) et de la **sémantique** (signification).

Une grammaire graphique est un cadre qui suit une approche **en couches** pour décrire et construire des visualisations ou des graphiques de manière **structurée**.

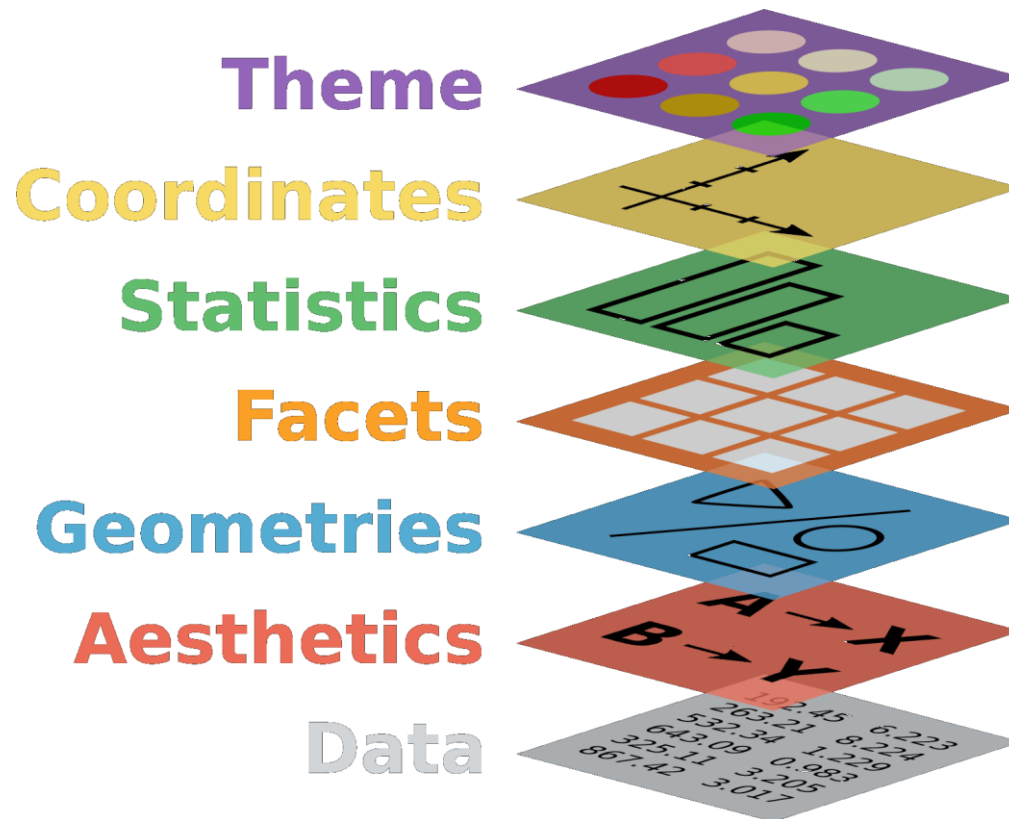
La grammaire **stratifiée** des graphiques utilise des composantes prédéfinies pour construire des diagrammes (au lieu d'essais et d'erreurs aléatoires)."

La grammaire stratifiées des graphiques



1. **Données** (obligatoire) : les observations se retrouvent en lignes, les variables en colonnes.
2. **Esthétique** (obligatoire) : le lien explicite de correspondance des variables aux échelles du graphique (position, forme, taille, couleur, etc.)
3. **Géométrie** (obligatoire) : le type de graphique sur lequel les données sont représentées (barres, lignes, points, etc.)
4. **Facettes** (facultatif) : les sous-ensembles de données représentés sur le graphique (niveaux)

La grammaire stratifiées des graphiques



5. **Statistiques** (facultatif) : les mesures qui pourraient fournir un contexte au graphique (centralité, dispersion, tendance, etc.)
6. **Coordonnées** (obligatoire) : l'espace de traçage du graphique (axes, échelle, etc.)
7. **Thèmes** (obligatoire) : les choix de design/conception utilisés pour créer une identité visuelle (polices, couleurs, etc.)

Exemples – Gapminder

L'ensemble de données Gapminder (<https://gapminder.org>) contient des informations socio-démographiques (plus de 500 variables) au sujet des nations de la Terre, pour des années allant de 1800 à 2020.

Nous déconstruisons 8 graphiques construits à partir de cet ensemble de données en utilisant la grammaire stratifiée des graphiques.



Données : Gapminder, 2009

Géométrie : diagramme de densité empilé

Esthétique :

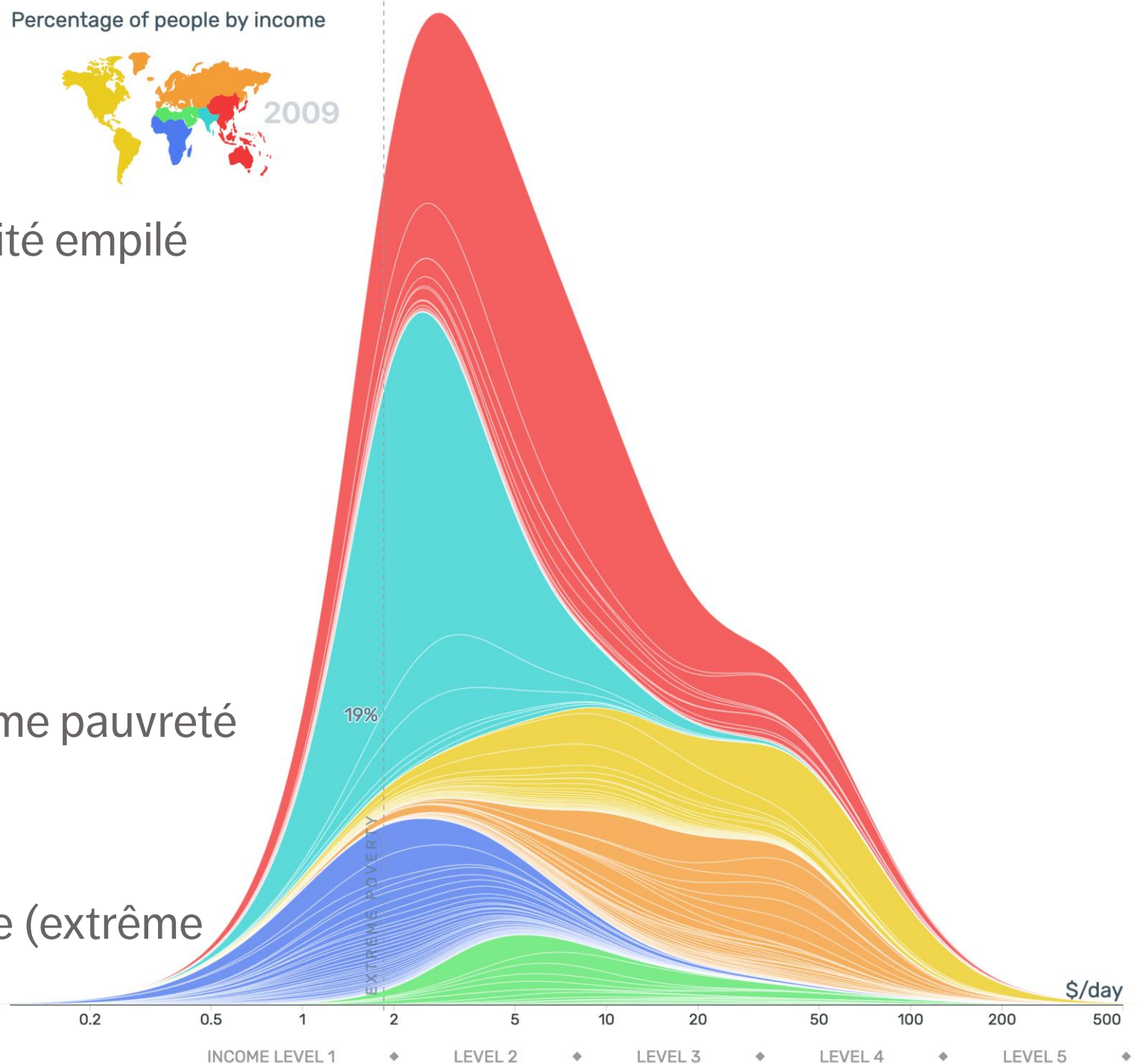
- x : revenu journalier
- (y : pourcentage par pays)
- “fill” : région

Facettes : aucune

Statistiques : proportion d'extrême pauvreté

Coordonnées : logarithmique (x)

Thème : Gapminder Tools; parure (extrême pauvreté)



Données : Gapminder, 2009

Géométrie : diagramme à bulles

Esthétique :

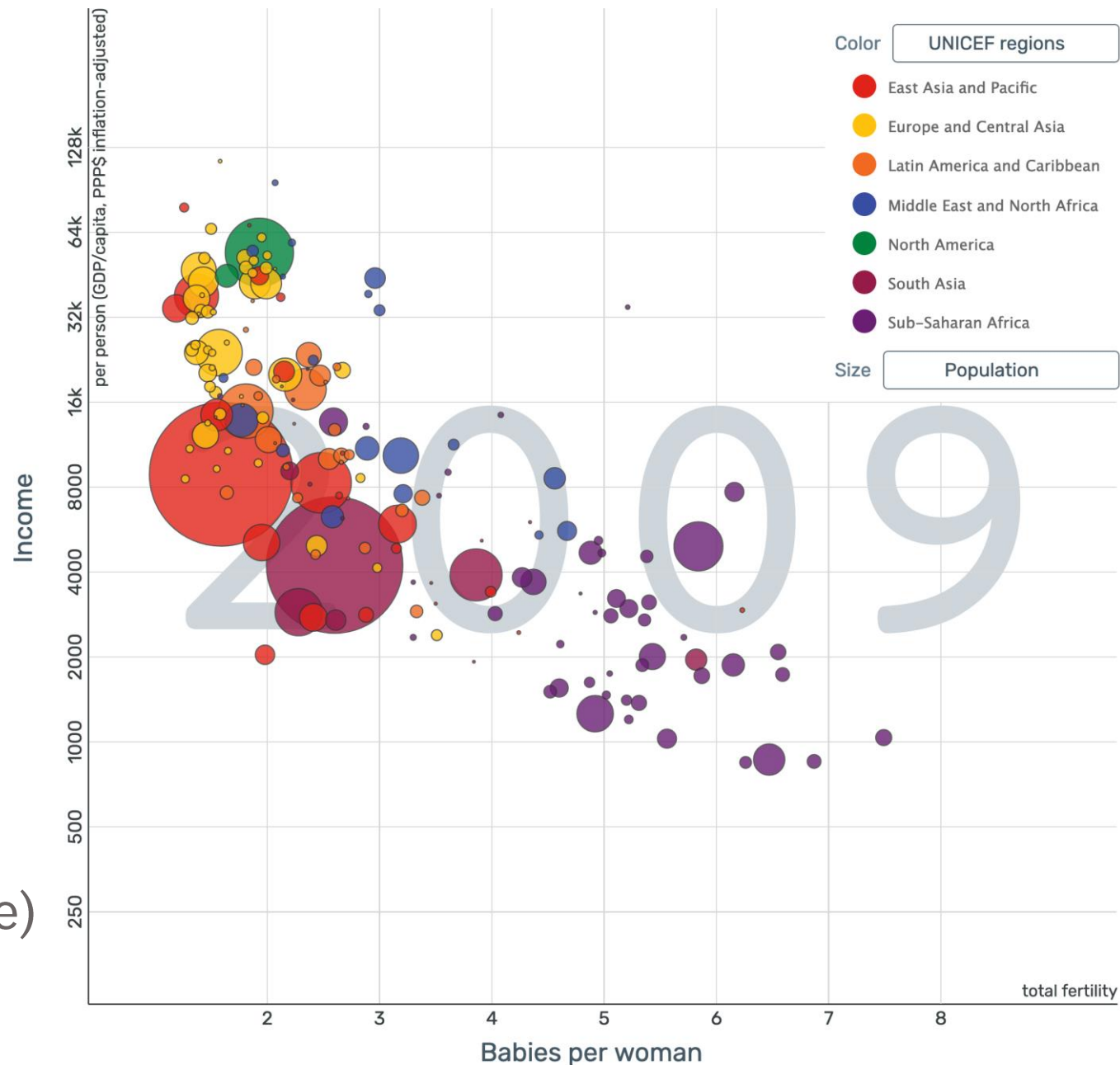
- x : fécondité totale
- y : revenu par personne
- “fill” : région de l'UNICEF
- taille : population

Facettes : aucune

Statistiques : aucune

Coordonnées : logarithmique (x, y, taille)

Thème : Gapminder Tools



Données : Gapminder, 2009

Géométrie : diagramme de dispersion

Esthétique :

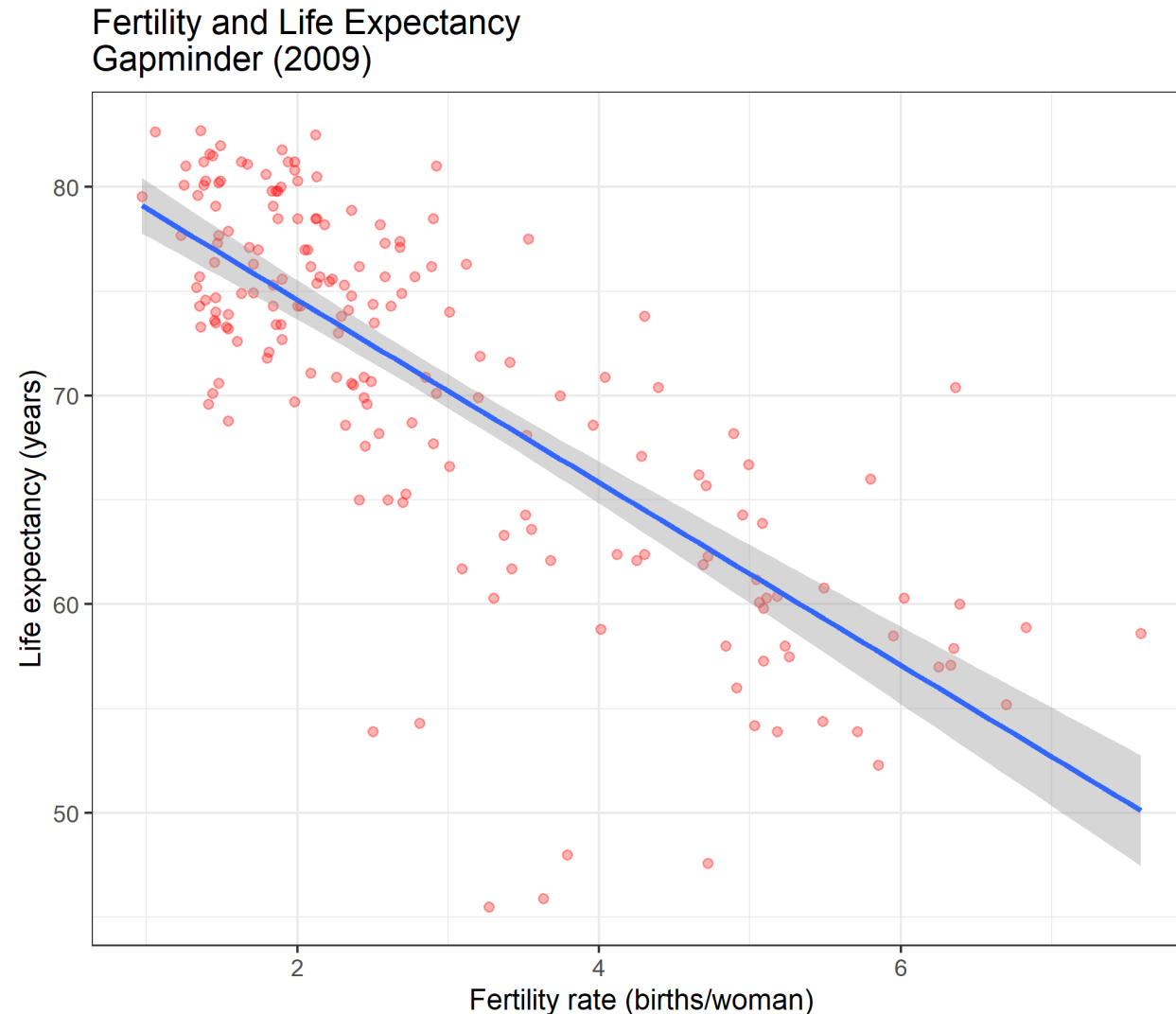
- x : fécondité totale
- y : espérance de vie

Facettes : aucune

Statistiques : ligne de meilleur ajustement, intervalle de confiance

Coordonnées : linéaire (x, y)

Thème : ggplot2, par défaut



Données : Gapminder, 2009

Facettes : continent

Géométrie : diagramme de densité

Statistiques : aucune

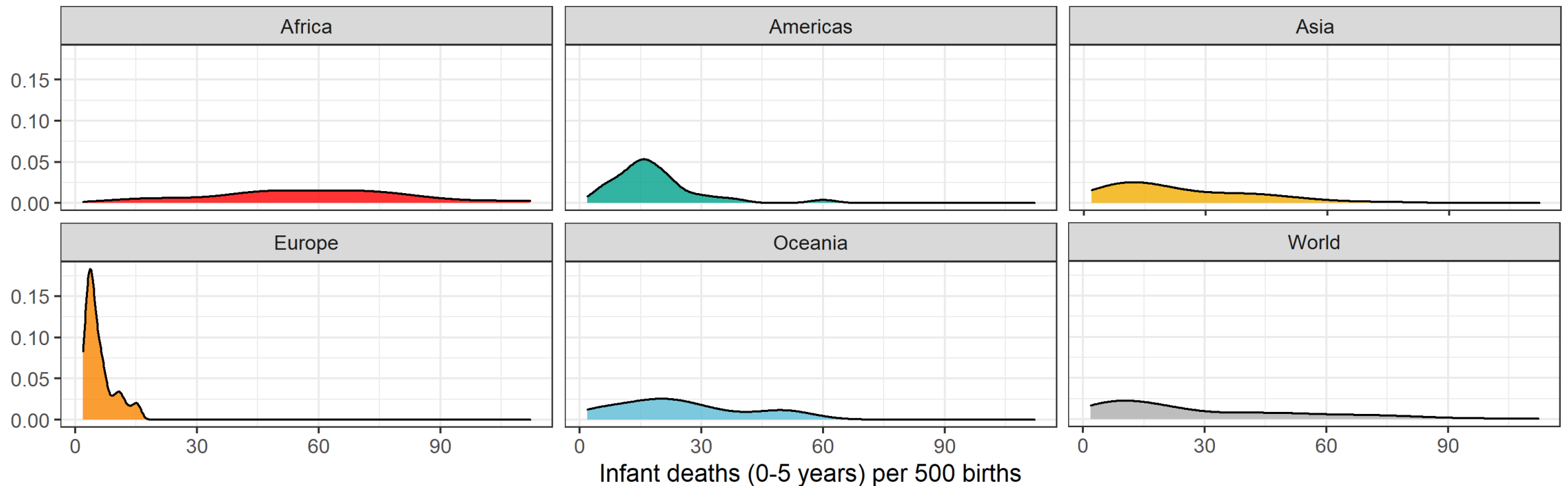
Esthétique :

- x : mortalité infantile
- "fill" : continent

Coordonnées : linéaire (x)

Thème : Darjeeling1

Infant Mortality by Continent Gapminder (2009)



Données : Gapminder, 2009

Géométrie : diagramme à bulles

Esthétique :

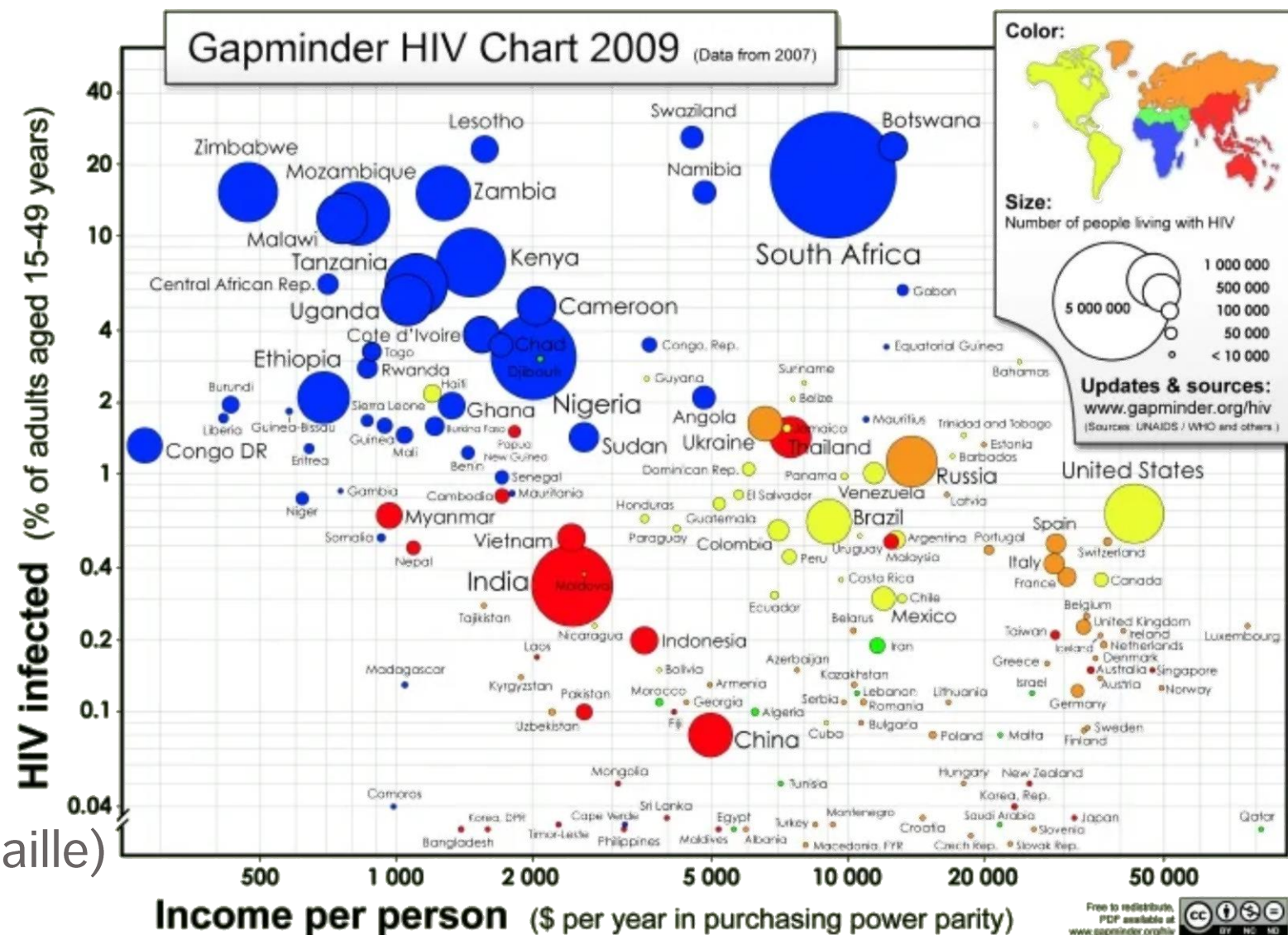
- x : revenu par personne
- y : taux d'infection par le VIH
- "fill" : région de l'OMS
- taille : population infectée par le VIH

Facettes : aucune

Statistiques : aucune

Coordonnées : logarithmique (x, y, taille)

Thème : ancient Gapminder Tools



Données : Gapminder, 2009

Géométrie : graphique boxplot

Esthétique :

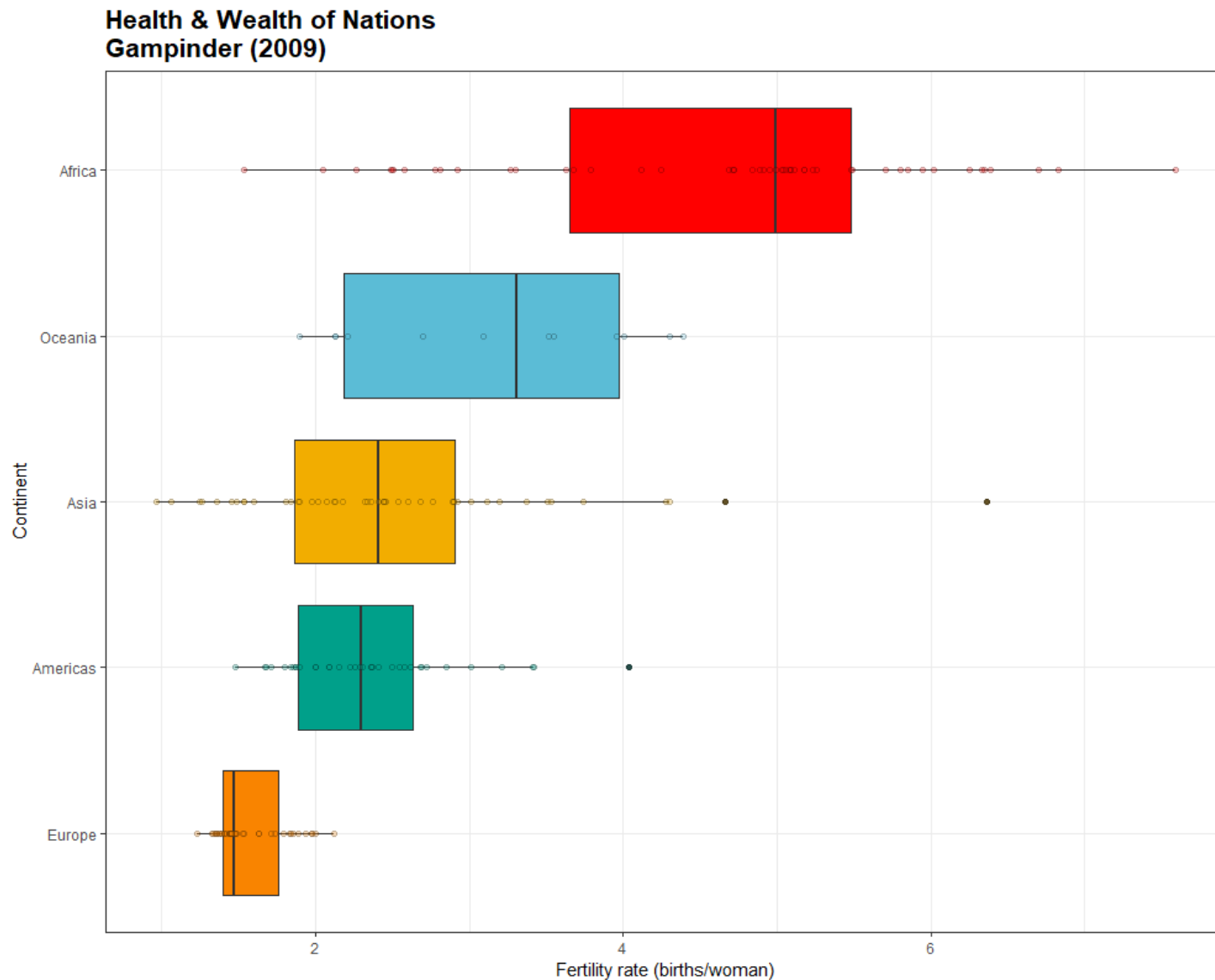
- x : taux de fécondité
- y : continent
- "fill" : continent

"Facettes" : continent

Statistiques : résumé à 5 points

Coordonnées : linéaire (x)

Thème : Darjeeling1



Données : Gapminder, 1960-2011

Géométrie : diagramme en lignes

Esthétique :

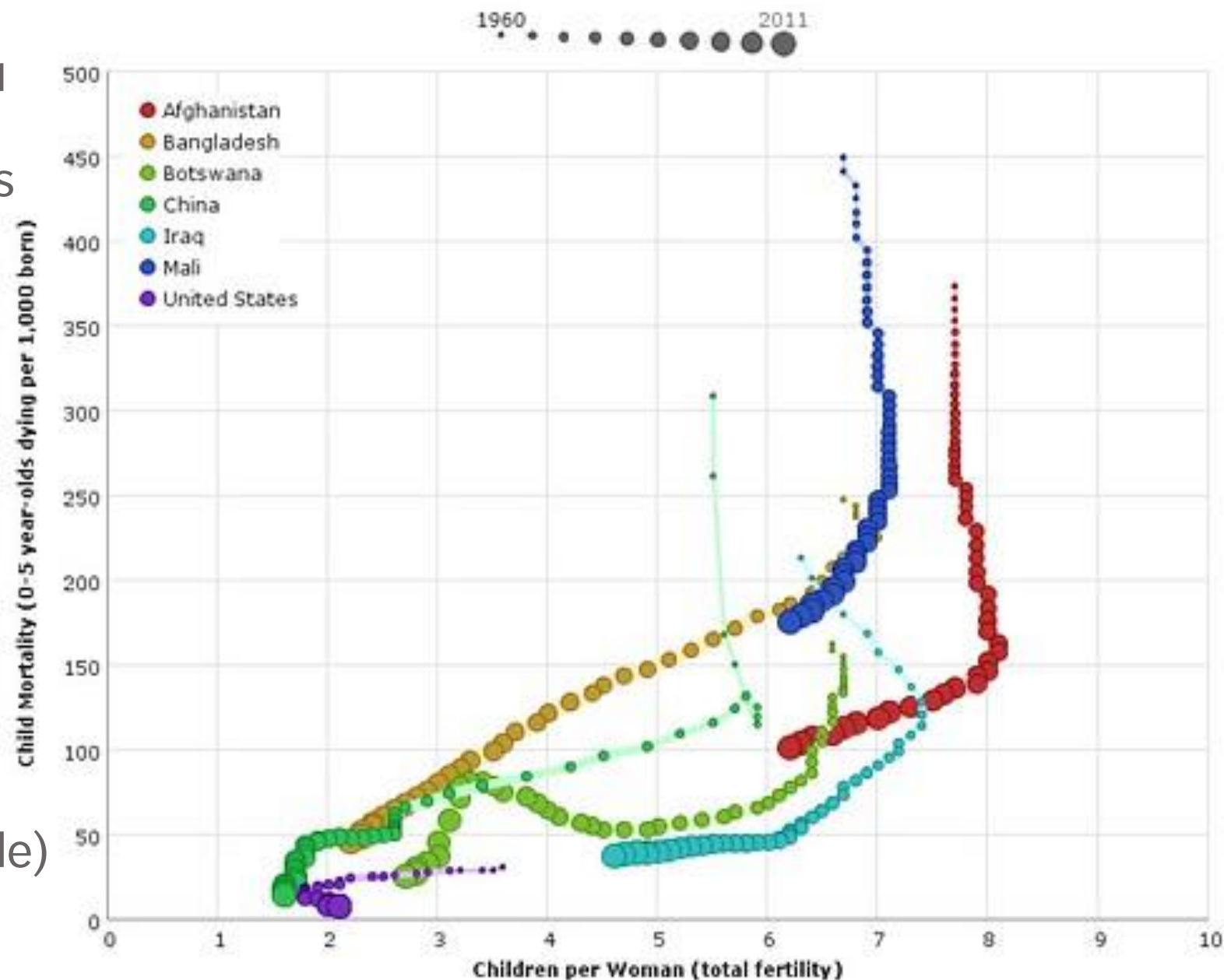
- x : fécondité totale
- y : pourcentage par pays
- couleur : pays
- taille : année

Facettes : aucune

Statistiques : aucune

Coordonnées : linéaires (x, y, taille)

Thème : personnalisé



Données : Gapminder, 2012

Géométrie : diagramme à bulles

Esthétique :

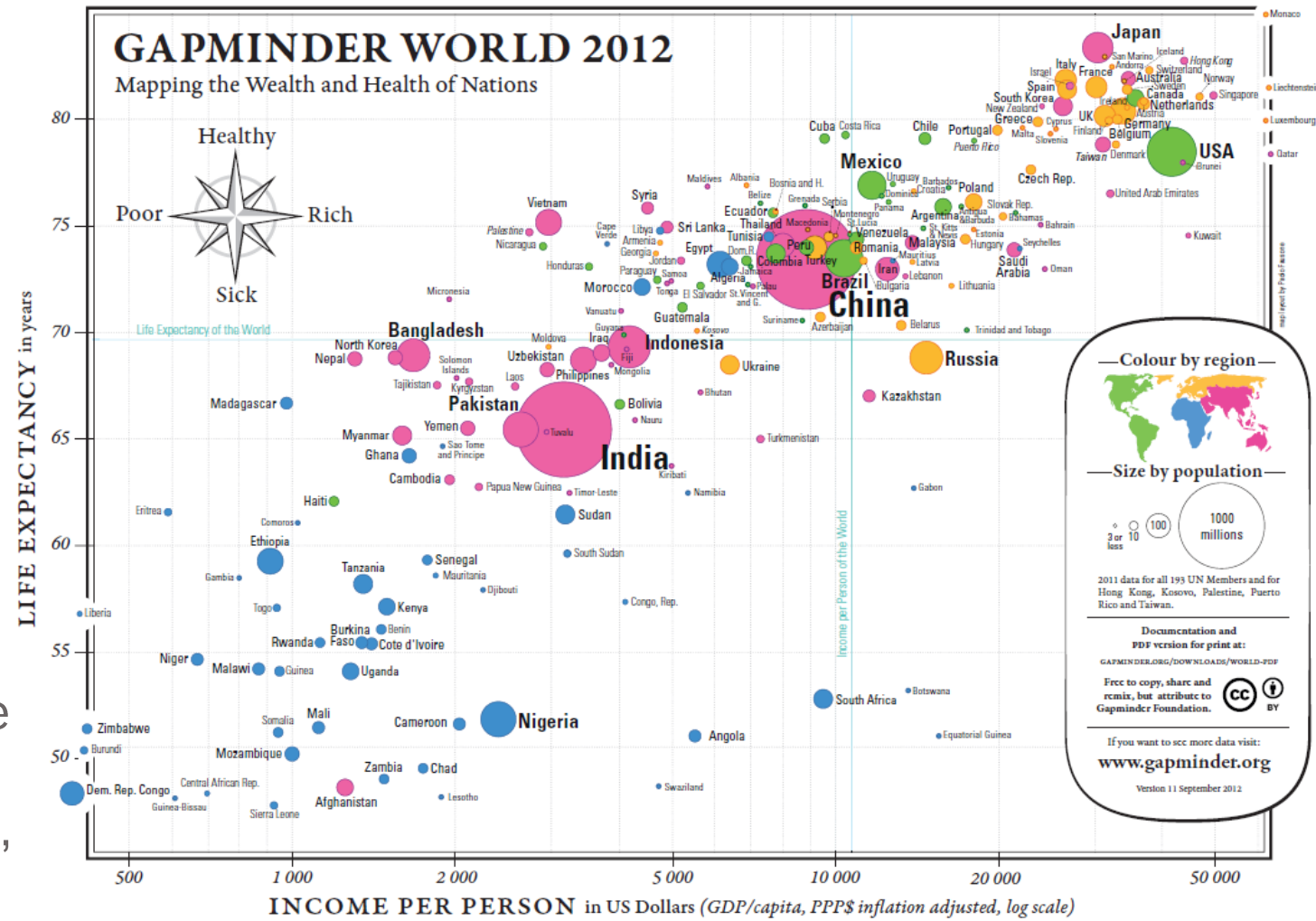
- x : revenu par personne
- y : espérance de vie
- "fill" : région
- taille : population

Facettes : aucune

Statistiques : espérance de vie mondiale, revenu mondial par personne

Coordonnées : logarithmique (x, taille), linéaire (y)

Thème : ancient Gapminder Tools



Lectures conseillées

La grammaire des graphiques

The Practice of Data Visualization

Part II: The Foundations of Visual Design

5. Visual Design and Data Charts

5.4 The Grammar of Graphics

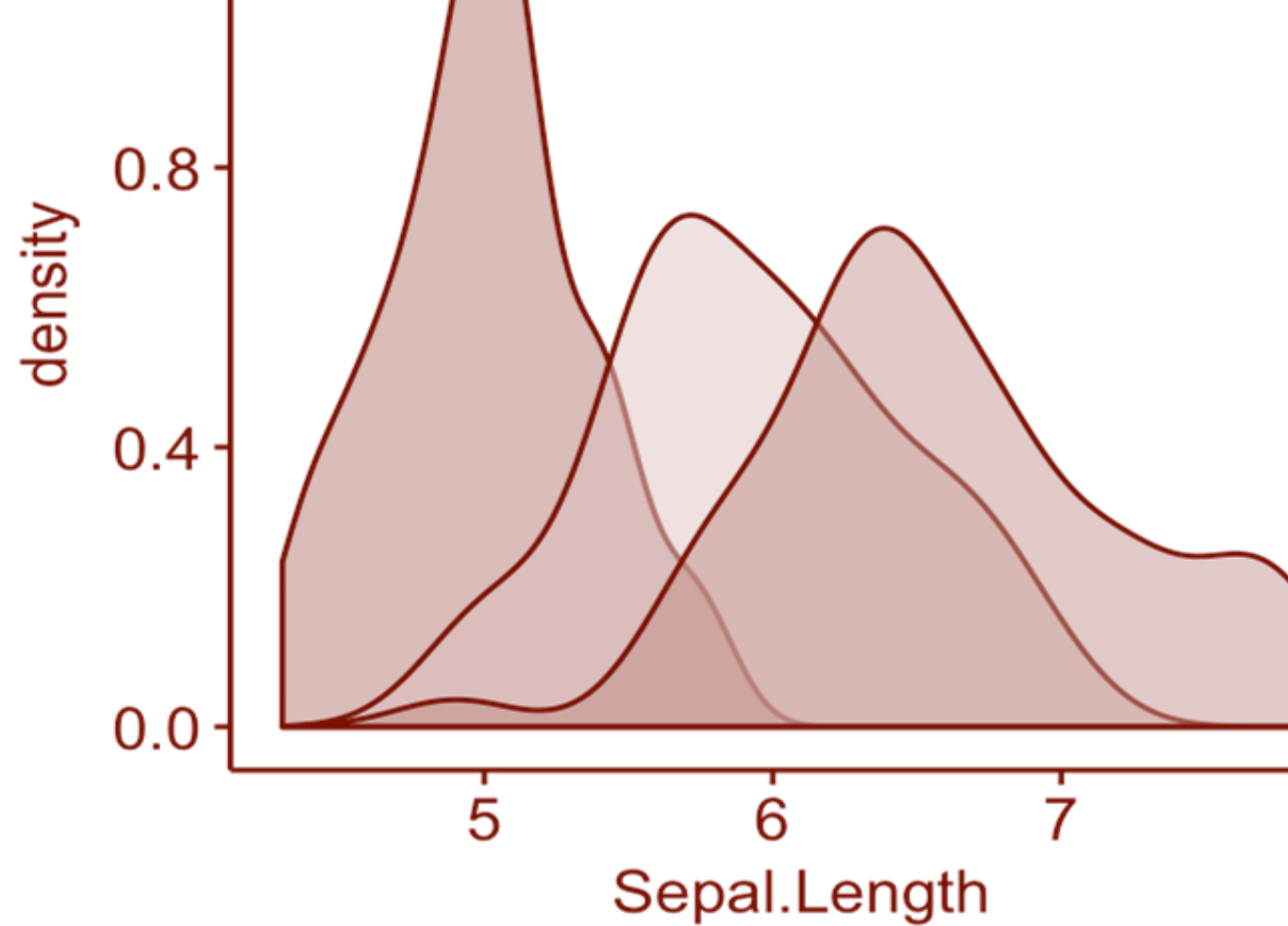
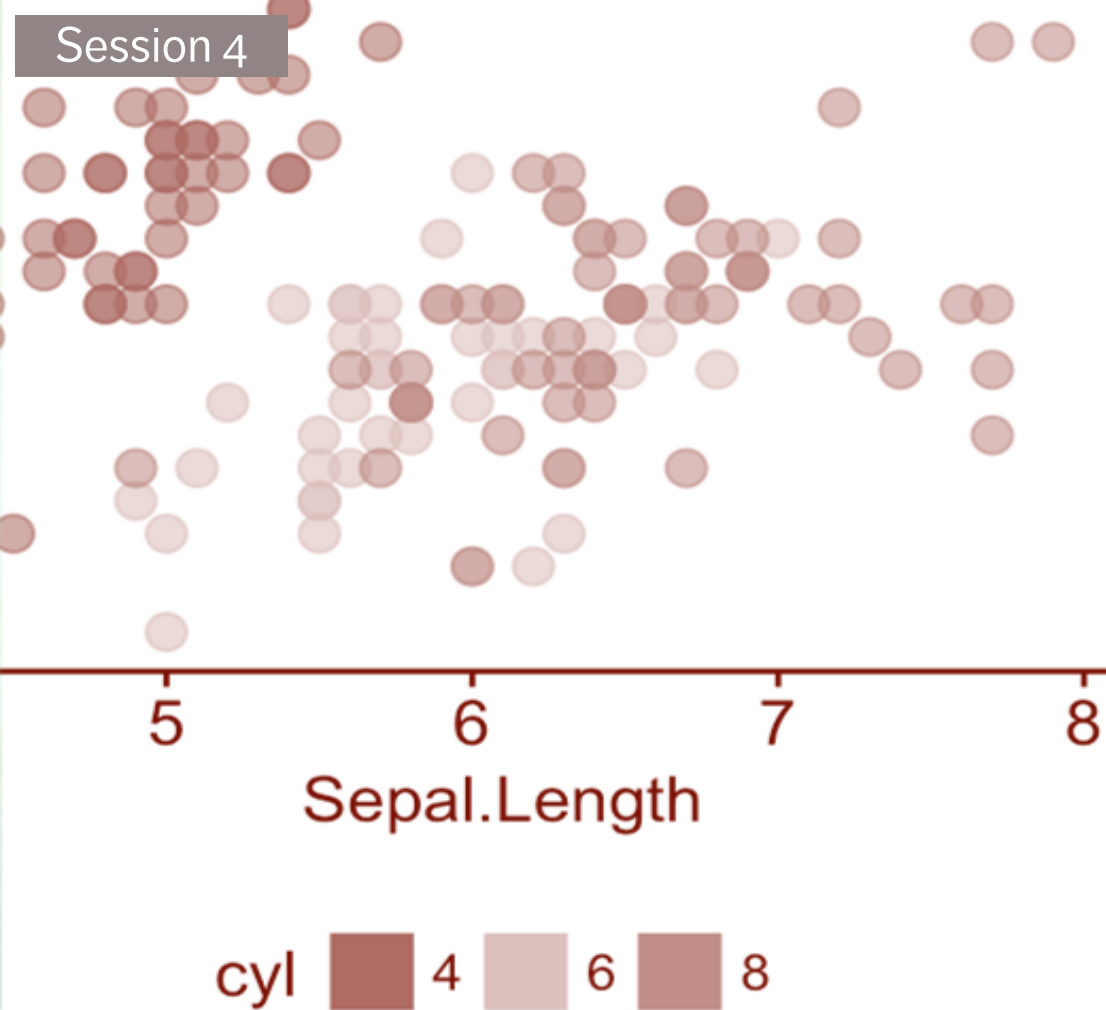
Exercices

La grammaire des graphiques

Déconstruisez les graphiques introduits dans les 9 premiers modules en termes de la grammaire des graphiques.

Qu'ont en commun certains des graphiques les plus efficaces ? Qu'en est-il de ceux qui sont les moins efficaces ?

Cela suggère-t-il une stratégie ?



11. Les bases de ggplot2

Le b-a-ba de ggplot2

La grammaire des graphiques est impléée dans `ggplot2`, un ensemble d'outils qui met en correspondance les données aux éléments d'affichage visuel et qui permettent aux utilisateurs de contrôler les détails de l'affichage du graphique.

L'aspect le plus important est que `ggplot2` peut être utilisé pour réfléchir à la **structure logique** des graphiques.

Un graphique `ggplot2` possède deux composantes principales :

- **mappings esthétiques** (`aes` – connexions entre les données et les éléments du graphique)
- **géométrie** (`geom` – spécifie le type de graphique)
- *facettes, *coordonnées, *échelles, *étiquettes, *guides, etc. (facultatifs)

La grammaire de ggplot2

1. Tidy Data

```
p <- ggplot(data = gapminder, ...
```

gdp	lifexp	pop	continent
340	65	31	Euro
227	51	200	Amer
909	81	80	Euro
126	40	20	Asia

2. Mapping

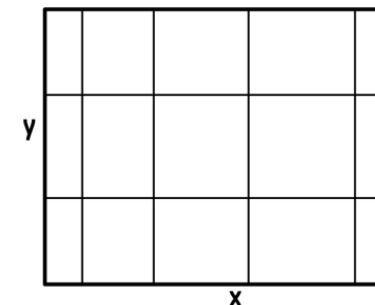
```
p <- ggplot(data = gapminder, mapping =  
  aes(x = gdp, y = lifexp, size = pop,  
      color = continent))
```

3. Geom

```
p + geom_point()
```

4. Co-Ordinates & Scales

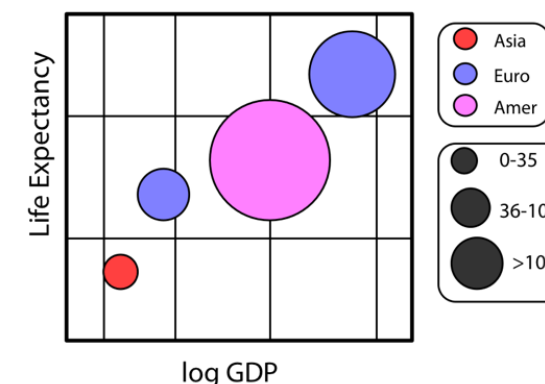
```
p + coord_cartesian() + scale_x_log10()
```



5. Labels & Guides

```
p + labs(x = "log GDP", y = "Life  
Expectancy", title = "A Gapminder Plot")
```

A Gapminder Plot



[Healey, K., *Data Visualization : A Practical Introduction*].

Les géométries de `ggplot2`

La source de données et les variables à tracer sont spécifiées via `ggplot()`.

Les différentes fonctions `geom()` précisent **comment** ces variables doivent être représentées visuellement :

- en utilisant des points, des barres, des lignes, des régions ombrées, etc.

Il y a actuellement plus de 37 géométries disponibles.

Function	Adds	Options
<code>geom_bar()</code>	bar chart	color, fill, alpha
<code>geom_boxplot()</code>	boxplot	color, fill, alpha, notch, width
<code>geom_density()</code>	density plot	color, fill, alpha, linetype
<code>geom_histogram()</code>	histogram	color, fill, alpha, linetype, binwidth
<code>geom_hline()</code>	horizontal lines	color, alpha, linetype, size
<code>geom_line()</code>	jittered points	color, size, alpha, shape
<code>geom_jitter()</code>	line graph	color, alpha, linetype, size
<code>geom_point()</code>	scatterplot	color, alpha, shape, size
<code>geom_rug()</code>	rug plot	color, side
<code>geom_smooth()</code>	fitted line	method, formula, color, fill, linetype, size
<code>geom_text()</code>	text annotations	many; see the help for this function
<code>geom_violin()</code>	violin plot	color, fill, alpha, linetype
<code>geom_vline()</code>	vertical lines	color, alpha, linetype, size

Option	Specifies
color	colour of points, lines, and borders around filled regions
fill	colour of filled areas such as bars and density regions
alpha	transparency of colors, ranging from 0 (fully transparent) to 1 (opaque)
linetype	pattern for lines (1 = solid, 2 = dashed, 3 = dotted, 4 = dotdash, 5 = longdash, 6 = twodash)
size	point size and line width
shape	point shapes (same as pch, with 0 = open square, 1 = open circle, 2 = open triangle, and so on)
position	position of plotted objects such as bars and points. For bars, <code>dodge''</code> places grouped bar charts side by side, <code>stacked''</code> vertically stacks grouped bar charts, and <code>fill''</code> vertically stacks grouped bar charts and standardizes their heights to be equal; for points, <code>jitter''</code> reduces point overlap
binwidth	bin width for histograms
notch	indicates whether box plots should be notched (TRUE/FALSE)
sides	placement of rug plots on the graph (<code>b''</code> = bottom, <code>l''</code> = left, <code>t''</code> = top, <code>r''</code> = right, <code>bl''</code> = both bottom and left, and so on)
width	width of box plots

Les geom() de ggplot2

```
library("ggplot2")
data(singer, package="lattice")
# Using data from the 1979 ed. of the
# New York Choral Society

# Histogram of heights
ggplot(singer, aes(x=height)) +
  geom_histogram()

# Boxplot of heights by voice part
ggplot(singer, aes(x=voice.part, y=height)) +
  geom_boxplot()
```

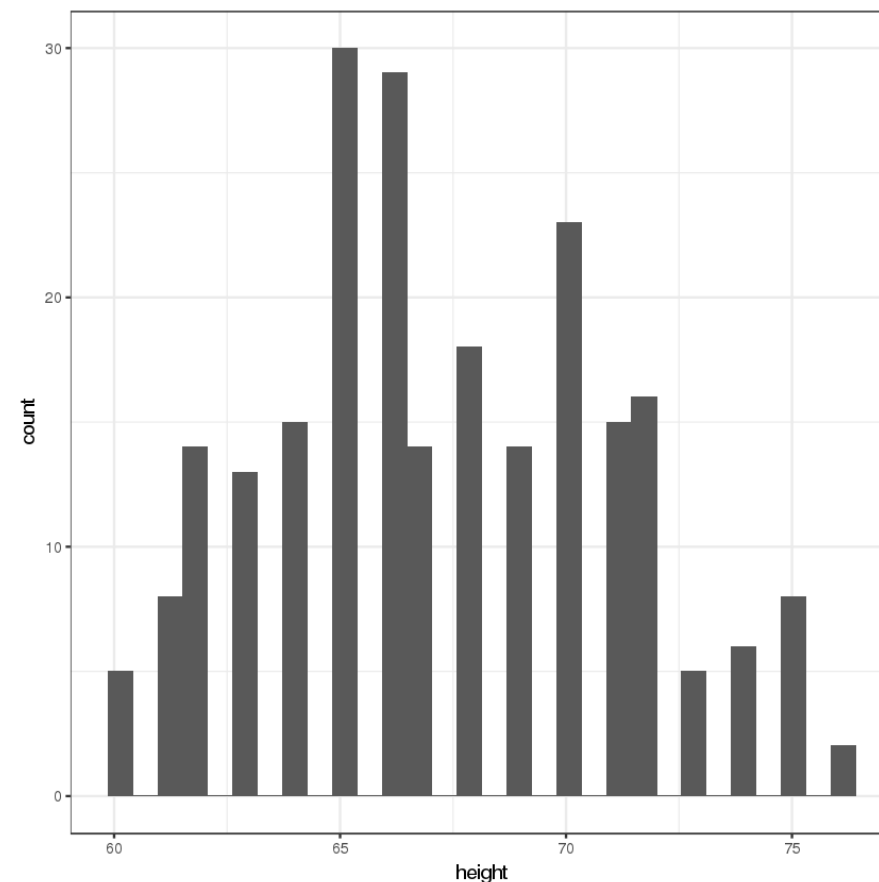
Quel est le résultat attendu ?

Les geom() de ggplot2

```
library("ggplot2")
data(singer, package="lattice")
# Using data from the 1979 ed. of the
# New York Choral Society

# Histogram of heights
ggplot(singer, aes(x=height)) +
  geom_histogram()

# Boxplot of heights by voice part
ggplot(singer, aes(x=voice.part, y=height)) +
  geom_boxplot()
```

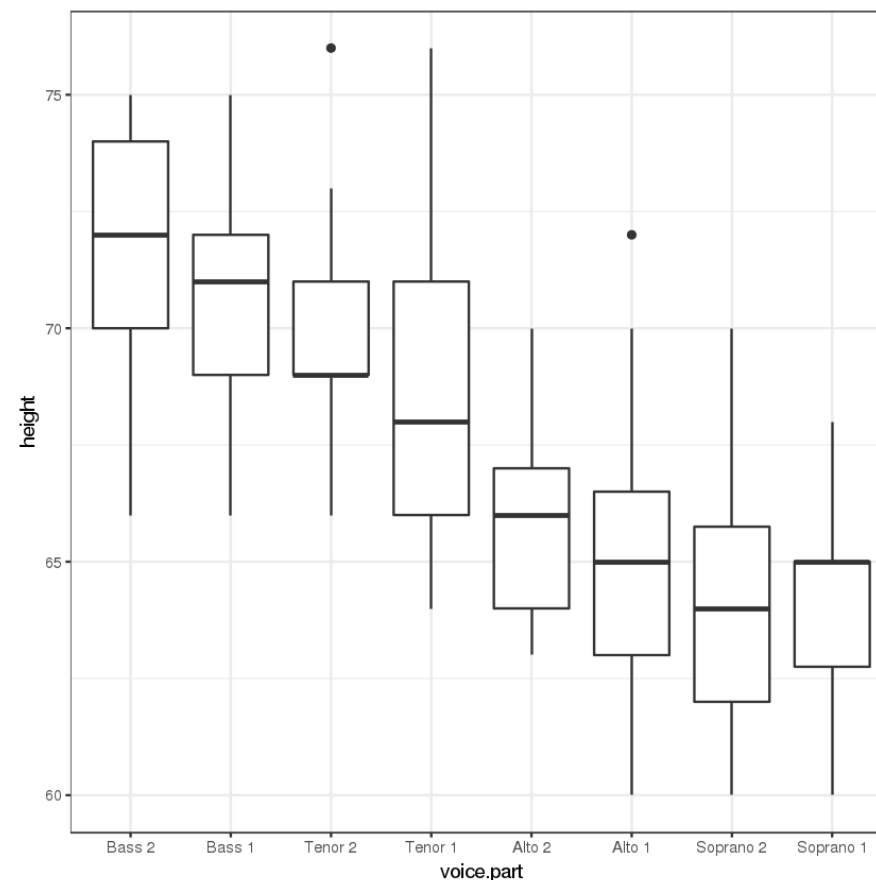


Les geom() de ggplot2

```
library("ggplot2")
data(singer, package="lattice")
# Using data from the 1979 ed. of the
# New York Choral Society

# Histogram of heights
ggplot(singer, aes(x=height)) +
  geom_histogram()

# Boxplot of heights by voice part
ggplot(singer, aes(x=voice.part, y=height)) +
  geom_boxplot()
```



Les geom() de ggplot2

```
library(ggplot2)
data(Salaries, package="car")
# Using data on salaries of a sample of
# US university professors (2018-2019)
# var: rank, sex, yrs.since.phd, yrs.service, salary

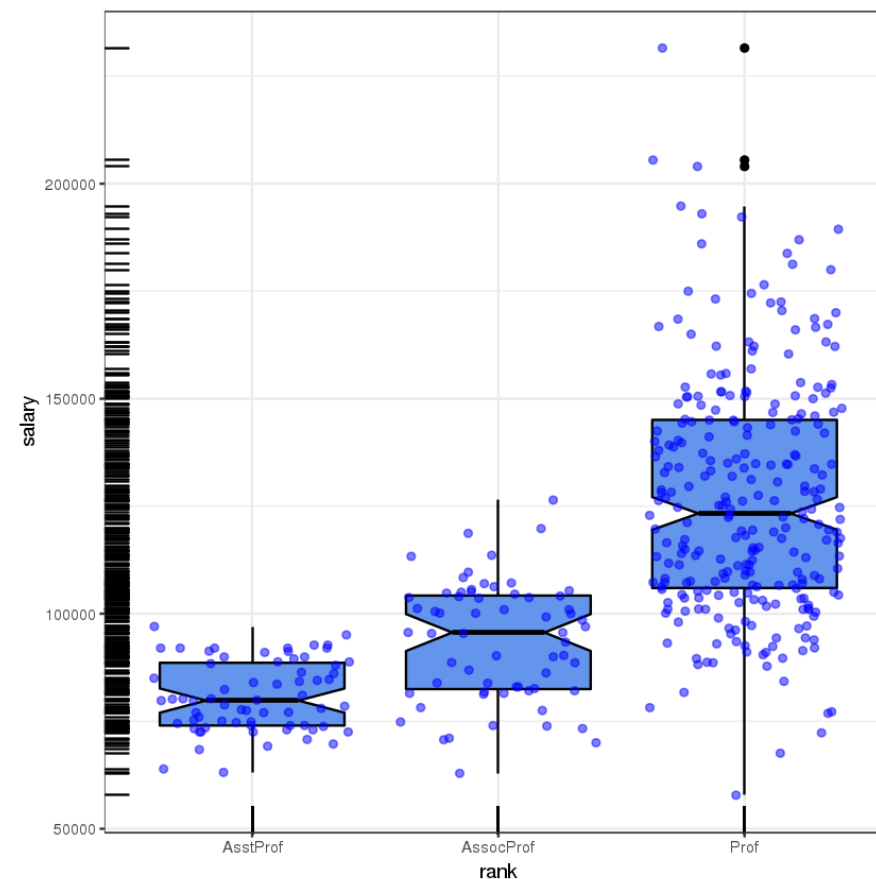
ggplot(Salaries, aes(x=rank, y=salary)) +
  geom_boxplot(fill="cornflowerblue", color="black", notch=TRUE) +
  geom_point(position="jitter", color="blue", alpha=.5) +
  geom_rug(side="l", color="black")
```

Quel est le résultat attendu ?

Les geom() de ggplot2

```
library(ggplot2)
data(Salaries, package="car")
# Using data on salaries of a sample of
# US university professors (2018-2019)
# var: rank, sex, yrs.since.phd, yrs.service, salary

ggplot(Salaries, aes(x=rank, y=salary)) +
  geom_boxplot(fill="cornflowerblue", color="black", notch=TRUE) +
  geom_point(position="jitter", color="blue", alpha=.5) +
  geom_rug(side="l", color="black")
```



L'esthétique de `ggplot2`

L'**esthétique** fait référence aux attributs affichés des données.

La **correspondance** associe les données à des attributs (tels que la taille ou la forme d'un marqueur) et génèrent une légende appropriée.

Les aspects esthétiques sont spécifiées avec la fonction `aes()`, soit dans l'appel de données, soit dans un appel à `geom()`. Si elles sont spécifiées dans `ggplot()`, elles s'appliquent à toutes les géométries spécifiées.

Les `aes()` de `ggplot2`

À titre d'exemple, les esthétiques disponibles pour `geom_point()` sont les suivantes :

- `x`, `y`, `alpha`, `colour`, `fill`, `shape`, `size`

Il y a une **différence importante** entre la spécification des caractéristiques (comme la couleur, la forme) à l'intérieur et à l'extérieur de l'appel à `aes()`

- à l'**intérieur** : couleur ou forme attribuée automatiquement en fonction des données
- à l'**extérieur** : les attributs ne sont pas mis en correspondance avec les données

Les aes() de ggplot2

```
library(ggplot2)
# Using the mpg dataset

# specifying characteristics inside aes()
ggplot(mpg, aes(cty, hwy)) +
  geom_point(aes(colour = class))

# specifying characteristics inside aes()
ggplot(mpg, aes(cty, hwy)) +
  geom_point(colour = "red")
```

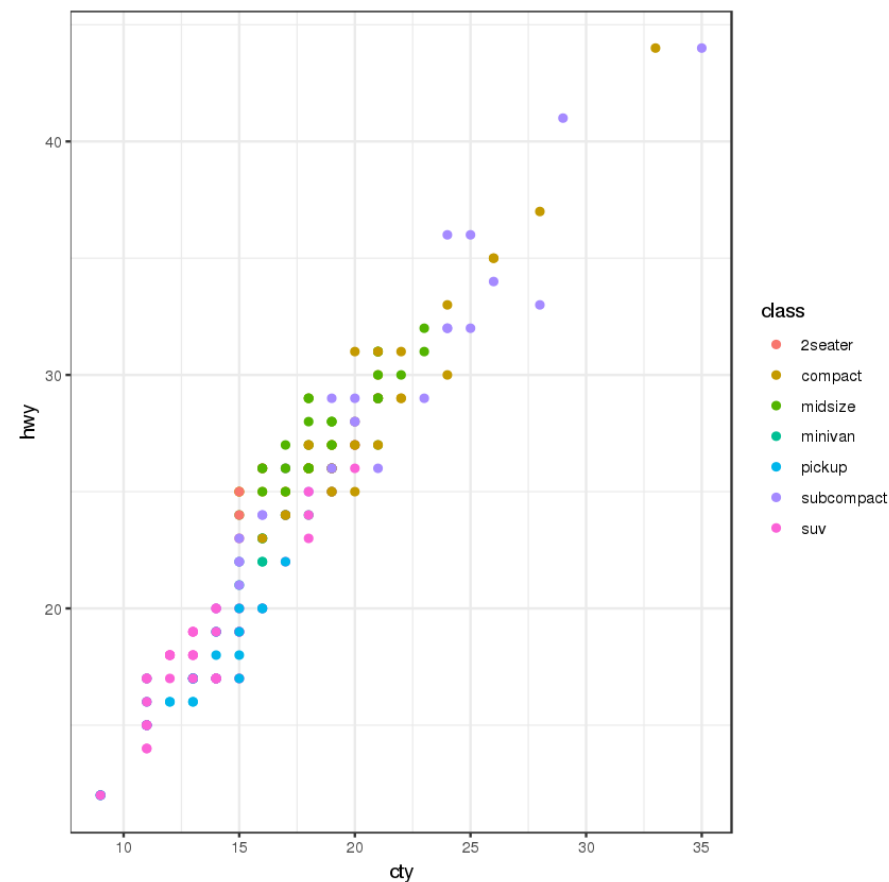
Quel est le résultat attendu ?

Les aes() de ggplot2

```
library(ggplot2)
# Using the mpg dataset

# specifying characteristics inside aes()
ggplot(mpg, aes(cty, hwy)) +
  geom_point(aes(colour = class))

# specifying characteristics outside aes()
ggplot(mpg, aes(cty, hwy)) +
  geom_point(colour = "red")
```

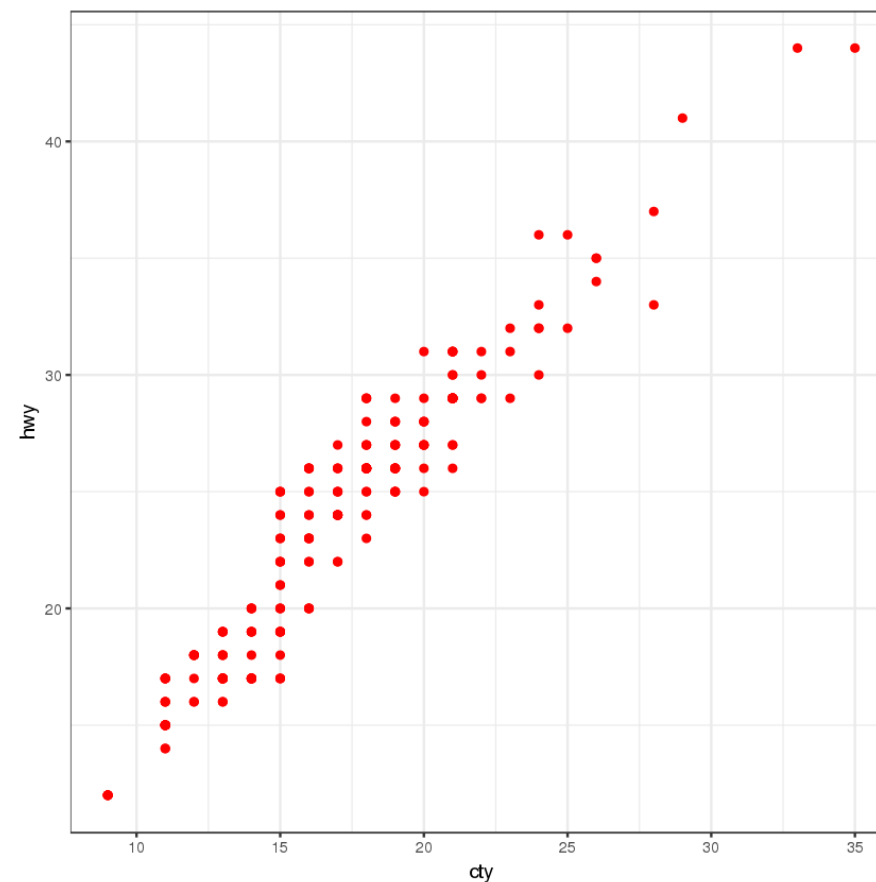


Les aes() de ggplot2

```
library(ggplot2)
# Using the mpg dataset

# specifying characteristics inside aes()
ggplot(mpg, aes(cty, hwy)) +
  geom_point(aes(colour = class))

# specifying characteristics inside aes()
ggplot(mpg, aes(cty, hwy)) +
  geom_point(colour = "red")
```



Les facettes de `ggplot2`

Dans le langage de `ggplot2`, les petits multiples sont appelés des facettes

- `facet_wrap()`, `facet_grid()`

Par défaut, tous les panneaux (un pour chaque facteur) partagent les mêmes axes (à l'échelle).

En séparant le graphique en une série de petits tracés côte-à-côte, il est plus facile de procéder à des comparaisons.

Les “wraps” n'affichent que les petits multiples pour lesquels il existe des données, les “grids” affichent tous les multiples, même ceux qui sont vides.

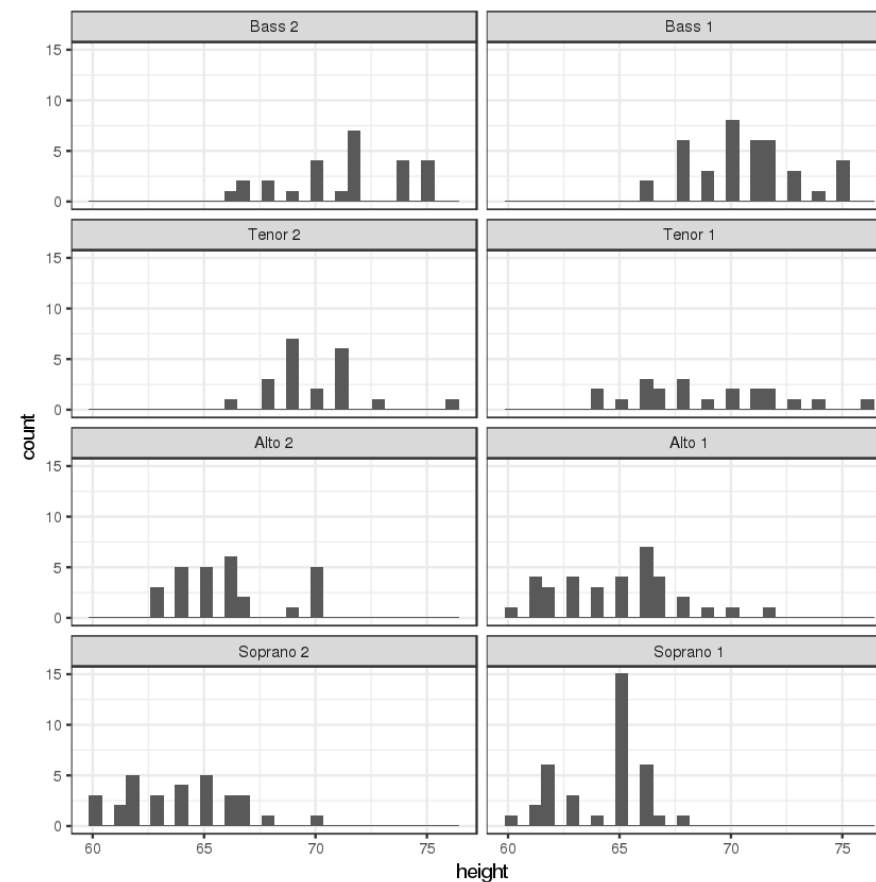
Les facet() de ggplot2

```
data(singer, package="lattice")  
library(ggplot2)  
ggplot(data=singer, aes(x=height)) +  
  geom_histogram() +  
  facet_wrap(~voice.part, nrow=4)
```

Quel est le résultat attendu ?

Les facet() de ggplot2

```
data(singer, package="lattice")
library(ggplot2)
ggplot(data=singer, aes(x=height)) +
  geom_histogram() +
  facet_wrap(~voice.part, nrow=4)
```



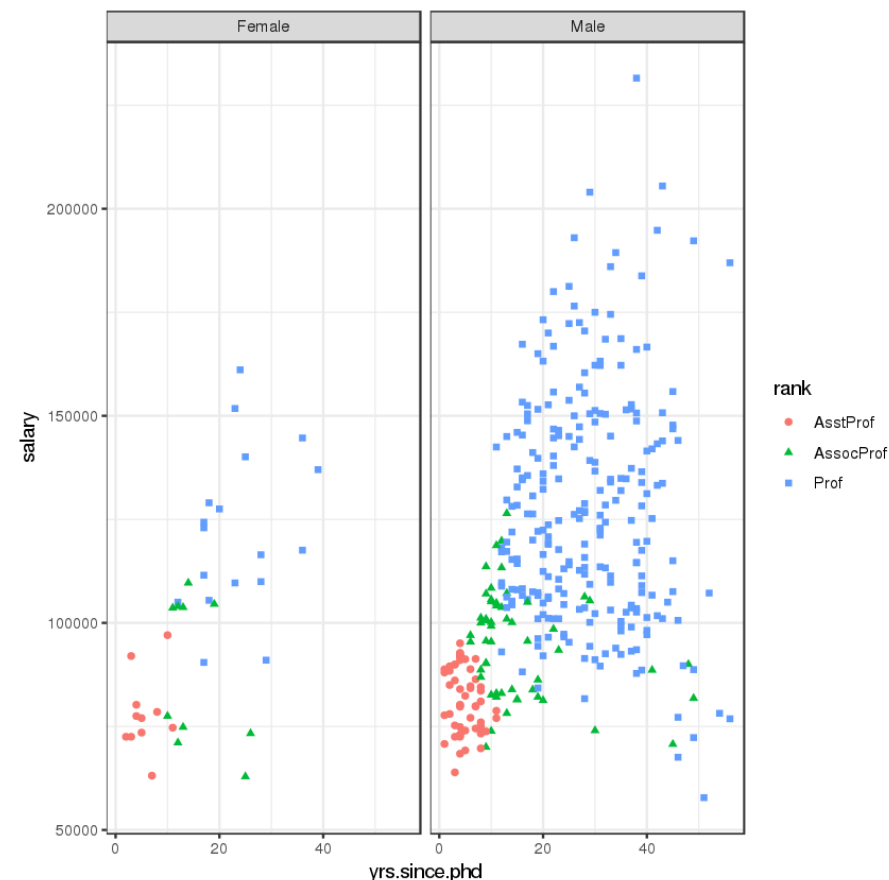
Les facet() de ggplot2

```
data(Salaries, package="car")
library(ggplot2)
ggplot(Salaries, aes(x=yrs.since.phd,
  y=salary, color=rank, shape=rank)) +
  geom_point() +
  facet_grid(~sex)
```

Quel est le résultat attendu ?

Les facet() de ggplot2

```
data(Salaries, package="car")
library(ggplot2)
ggplot(Salaries, aes(x=yrs.since.phd,
  y=salary, color=rank, shape=rank)) +
  geom_point() +
  facet_grid(~sex)
```



Lectures conseillées

Les bases de `ggplot2`

The Practice of Data Visualization

Part IV: Some Practical Aspects of Data Visualization

12. `ggplot2` Visualizations in R

12.1 Basics of `ggplot2`'s Grammar

12.2 `ggplot2` Miscellanea

Exercices

Les bases de `ggplot2`

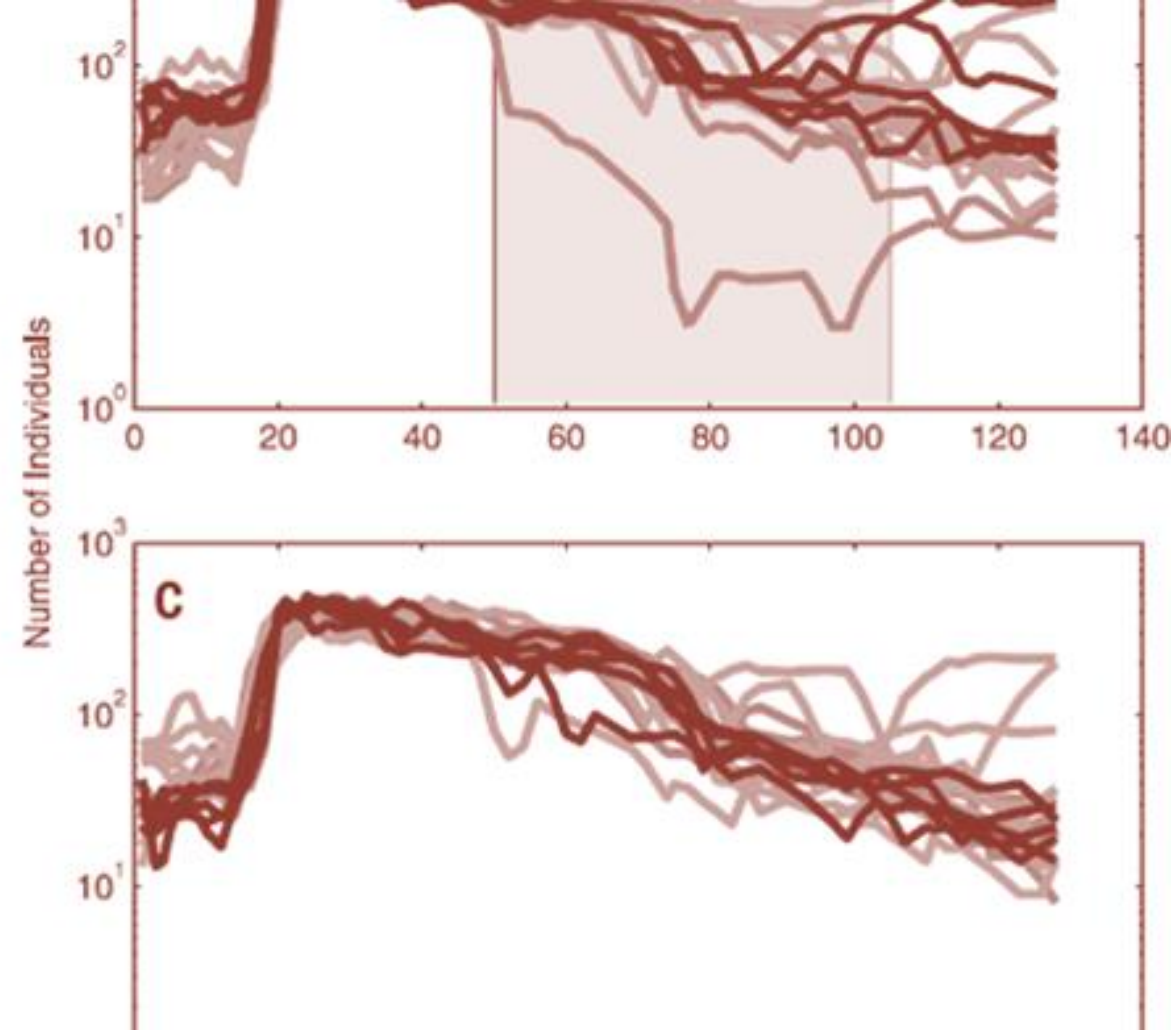
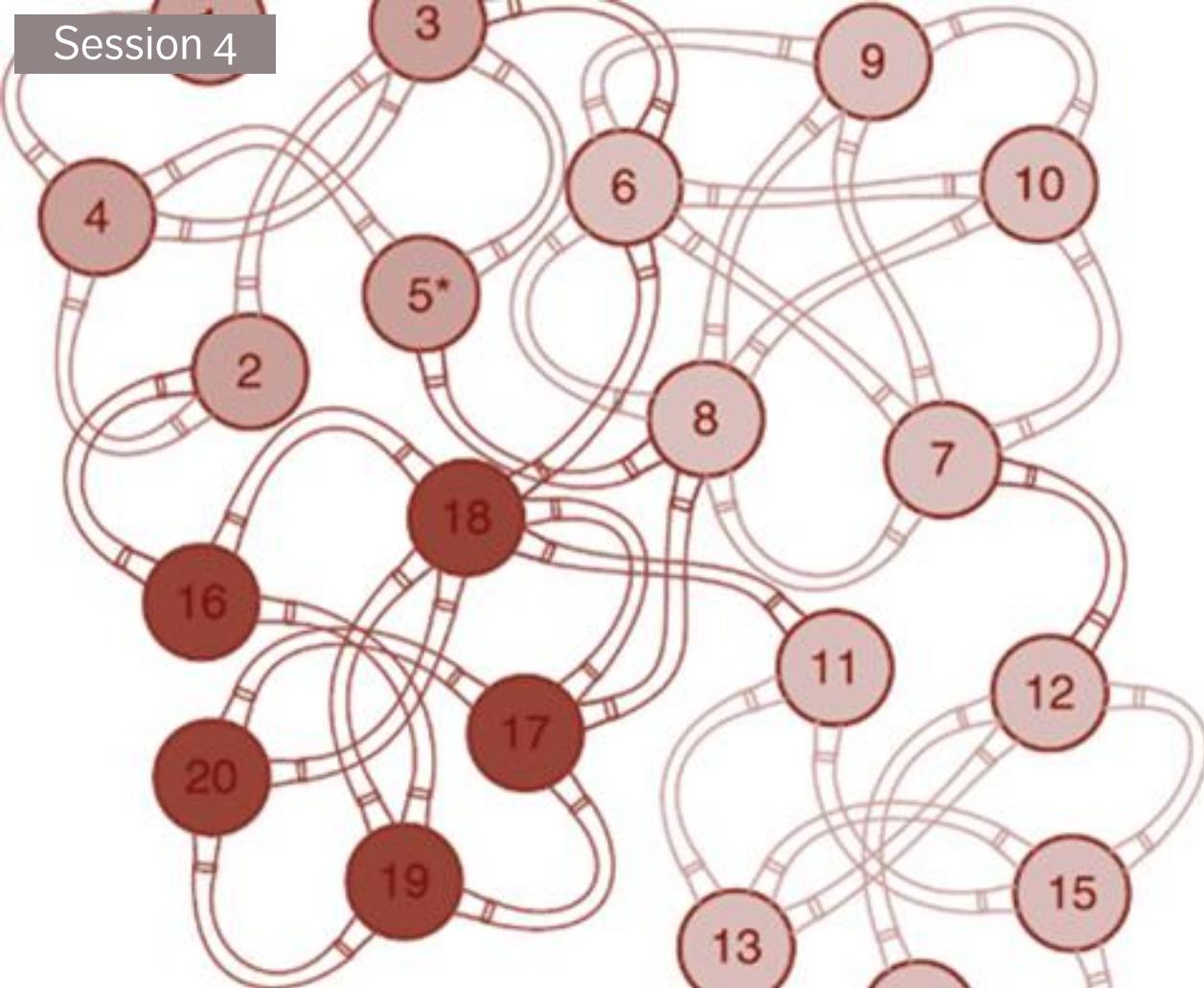
Créez quelques visualisations `ggplot2` simples avec les ensembles de données disponibles dans R. L'accent est mis sur la familiarisation avec diverses géométries, leur esthétique, et l'utilisation de facettes.

Vous pouvez utiliser les exemples trouvés dans ce module comme fondations de votre travail.

Les ensembles de données disponibles sont obtenus en exécutant

```
> données()
```

dans R.



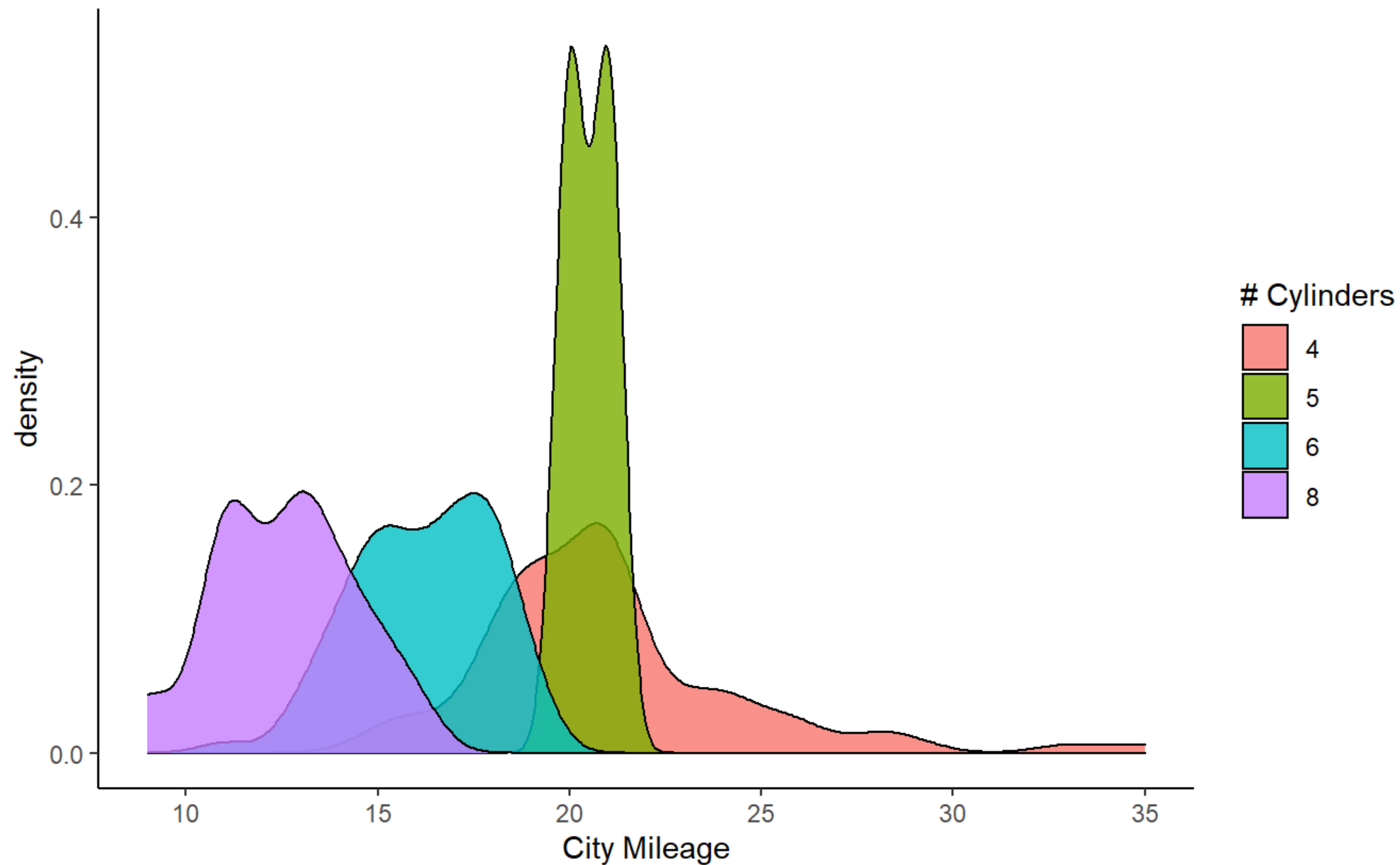
12. Exemples et miscellanea

Exemple 1

```
1 library(ggplot2)
2 theme_set(theme_classic())
3
4 # Plot
5 g <- ggplot(mpg, aes(cty))
6 g + geom_density(aes(fill=factor(cyl)), alpha=0.8) +
7   labs(title="Density Plot",
8         subtitle="City Mileage Grouped by Number of cylinders",
9         caption="Source: mpg",
10        x="City Mileage",
11        fill="# Cylinders")
```

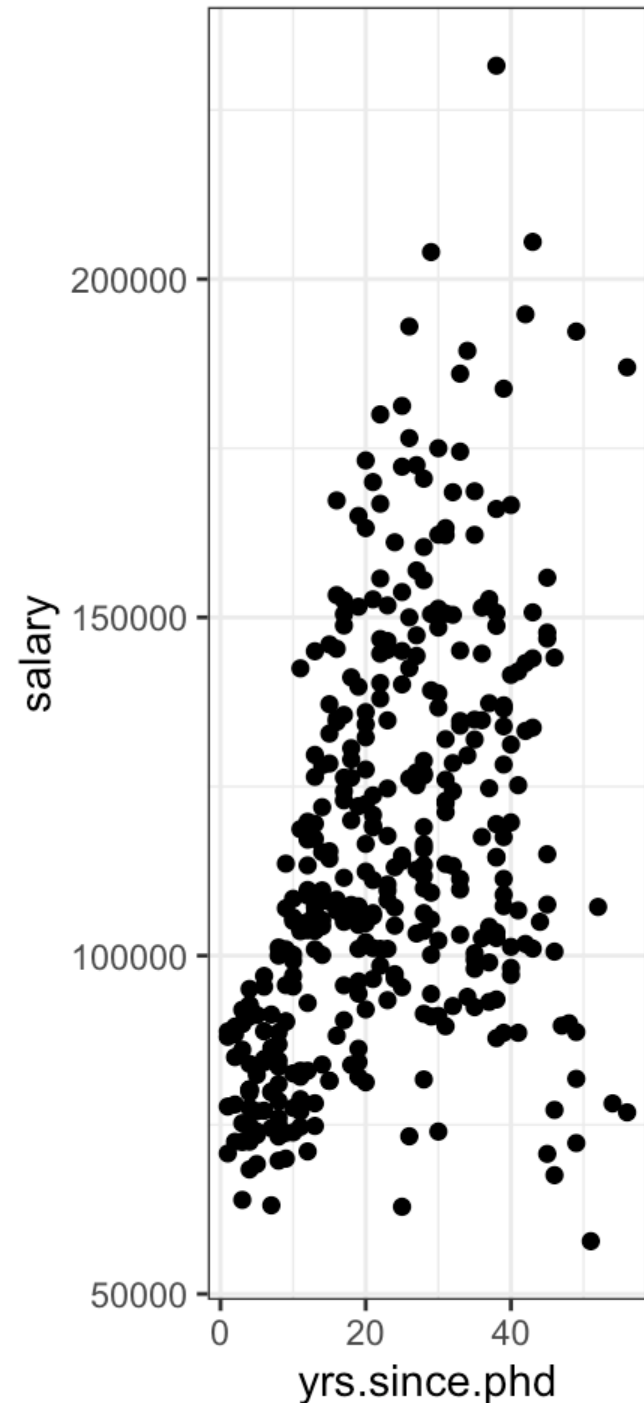
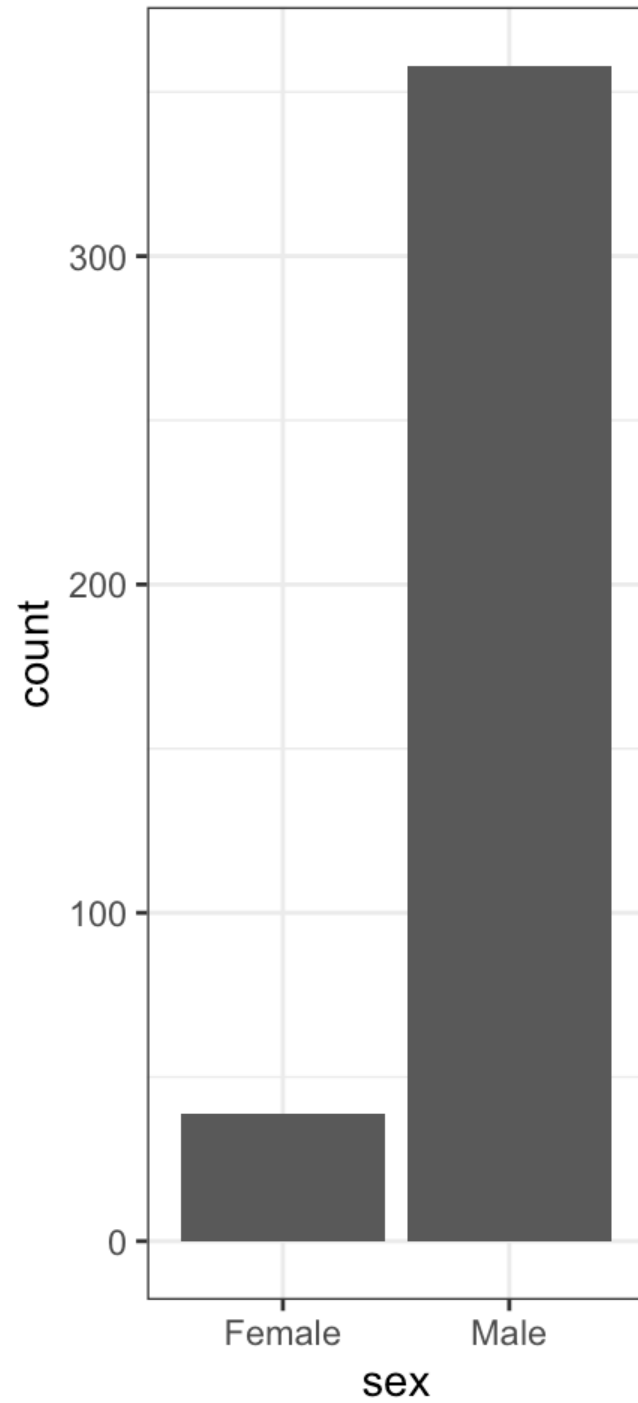
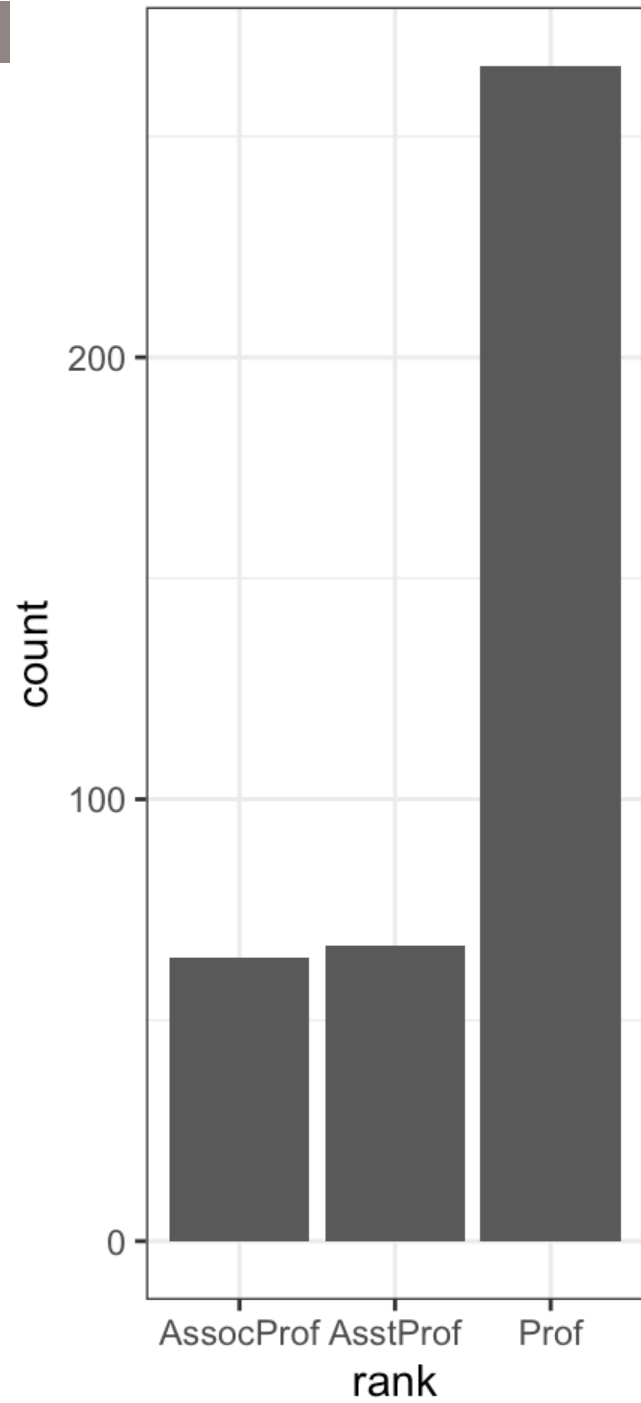
Density Plot

City Mileage Grouped by Number of cylinders



Exemple 2

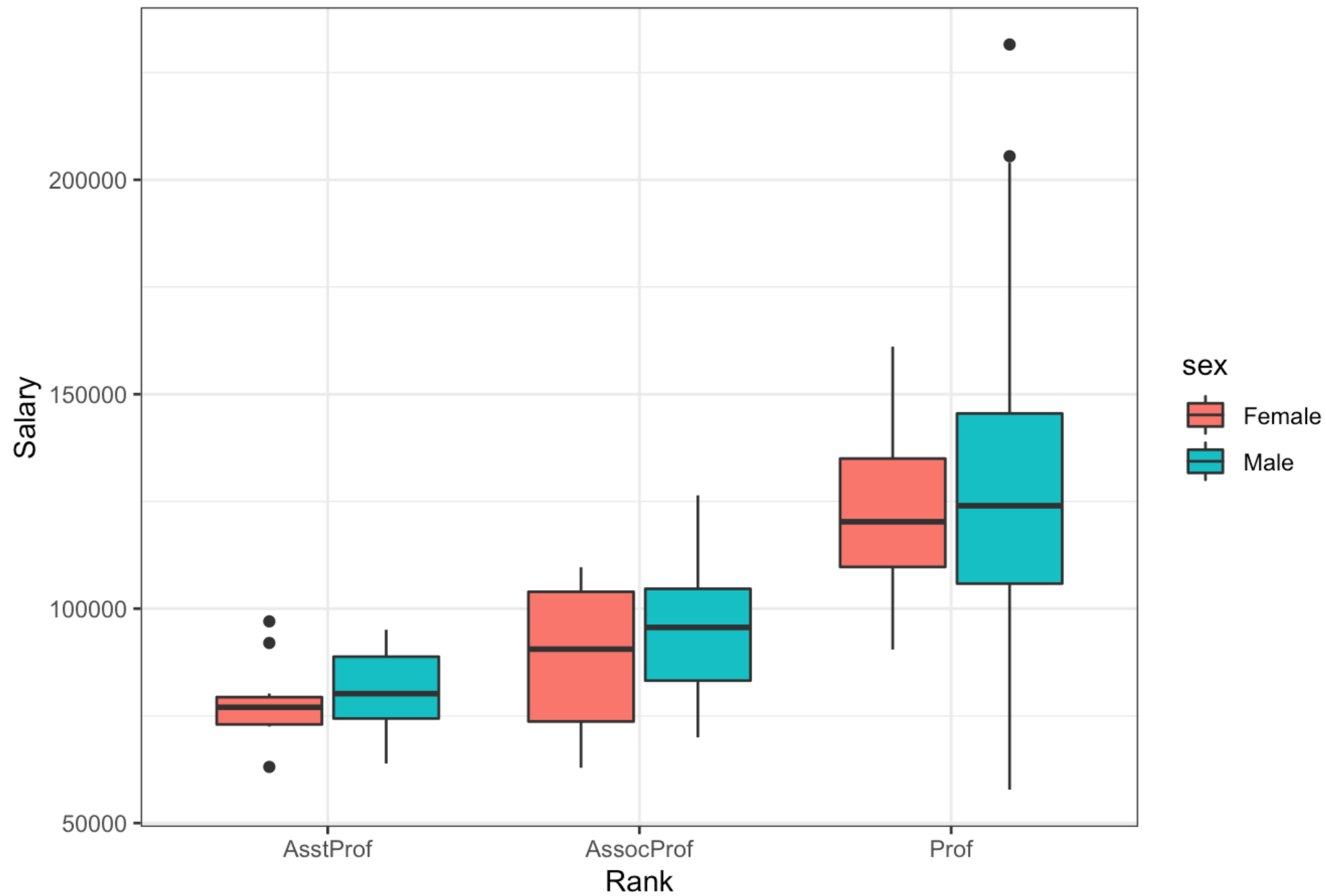
```
1 library(ggplot2)
2 p1 <- ggplot(data=Salaries, aes(x=rank)) + geom_bar()
3 p2 <- ggplot(data=Salaries, aes(x=sex)) + geom_bar()
4 p3 <- ggplot(data=Salaries, aes(x=yrs.since.phd, y=salary)) + geom_point()
5
6 library(gridExtra)
7 grid.arrange(p1, p2, p3, ncol=3)
```



Exemple 3

```
1 library(ggplot2)
2 mytheme <- theme(plot.title=element_text(face="bold", size=14, color="brown"),
3                   axis.title=element_text(size=10, color="brown"),
4                   axis.text=element_text(size=9, color="black"),
5                   panel.background=element_rect(fill="white",color="black"),
6                   panel.grid.major.y=element_line(color="grey", linetype=1),
7                   panel.grid.minor.y=element_line(color="grey", linetype=2),
8                   panel.grid.minor.x=element_blank(),
9                   legend.position="top")
10
11 ggplot(Salaries, aes(x=reorder(rank,salary), y=salary, fill=sex)) +
12   geom_boxplot() +
13   labs(title="Salary by Rank and Sex", x="Rank", y="Salary")
```

Salary by Rank and Sex



Exemple 4

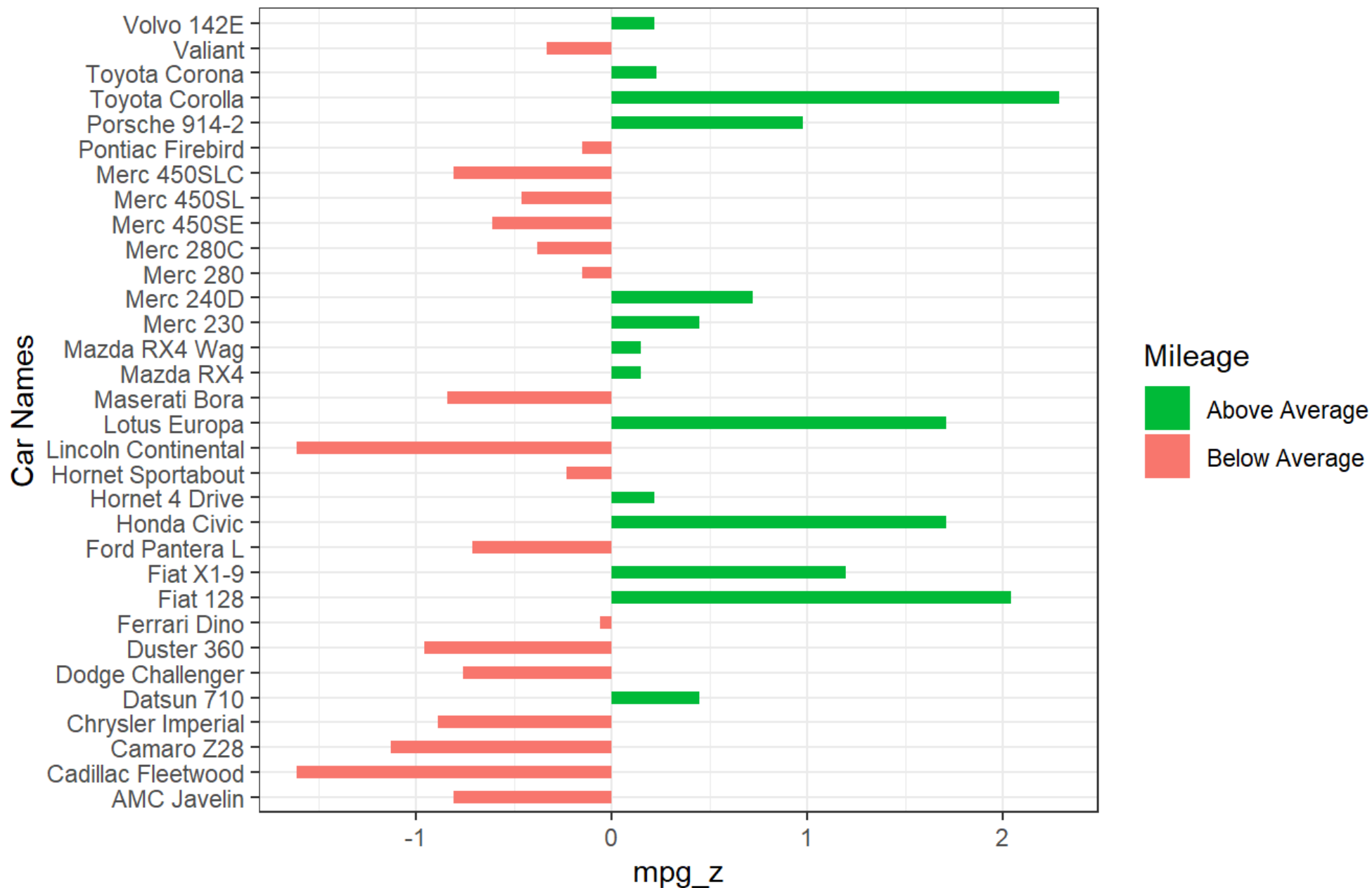
```

1 library(ggplot2)
2 theme_set(theme_bw())
3
4 mtcars_new <- mtcars |>
5   tibble::rownames_to_column(var = "car_name") |> # convert row names to a new column
6   dplyr::mutate(car_name = as.factor(car_name), # convert to factor to retain sorted order
7   mpg_z = round(scale(mpg), 2), # compute normalized mpg
8   mpg_type = ifelse(mpg_z < 0, "below", "above") # above / below avg flag
9   ) |>
10  dplyr::arrange(mpg_z) # sort
11
12 # Diverging Barcharts
13 ggplot(mtcars_new, aes(x= car_name, y=mpg_z, label=mpg_z)) +
14   geom_bar(stat='identity', aes(fill=mpg_type), width=.5) +
15   scale_fill_manual(name="Mileage",
16                     labels = c("Above Average", "Below Average"),
17                     values = c("above"="#00ba38", "below"="#f8766d")) +
18   labs(subtitle="Normalised mileage from 'mtcars'",
19        title= "Diverging Bars",
20        x = "Car Names") +
21   coord_flip()

```

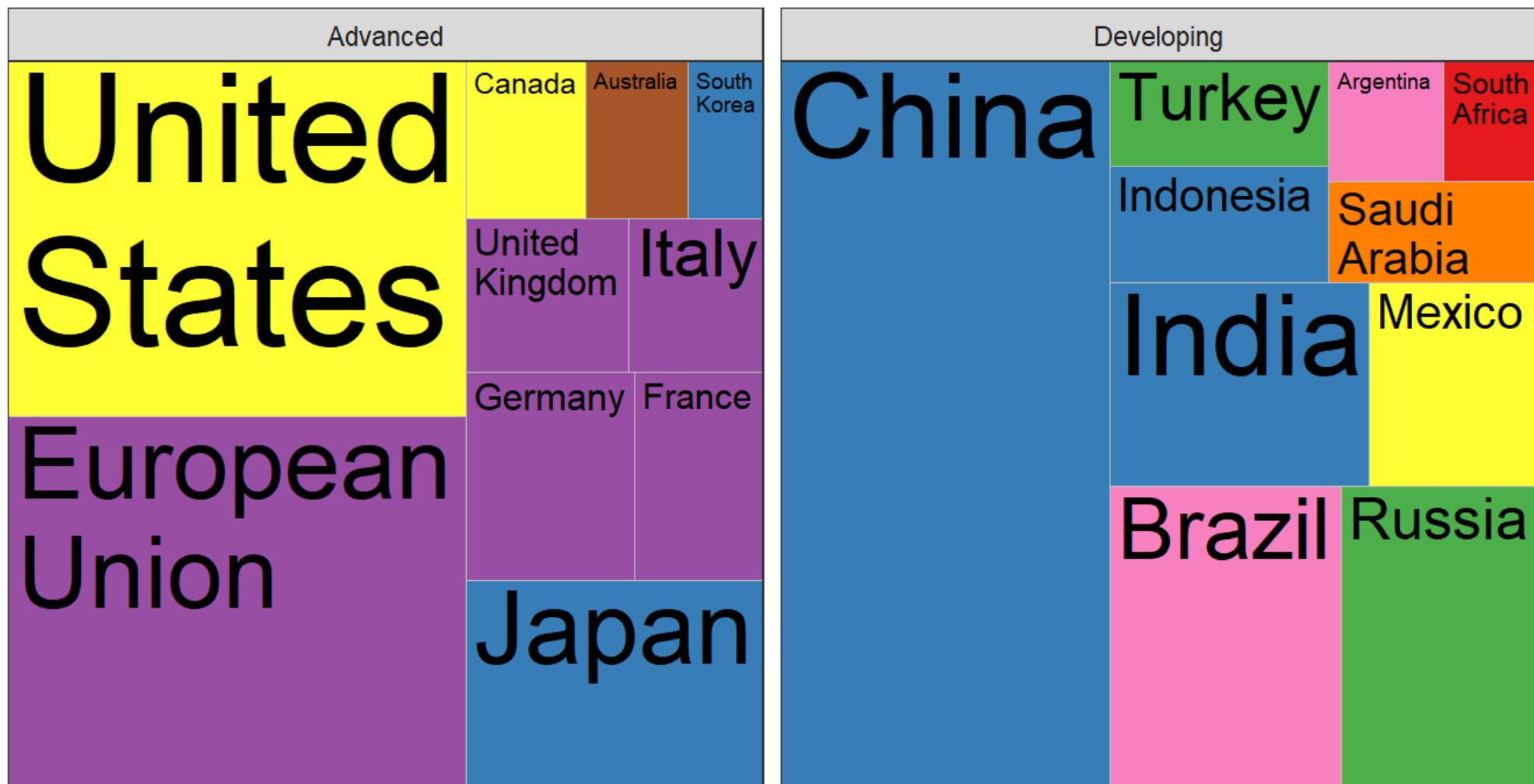
Diverging Bars

Normalised mileage from 'mtcars'



Exemple 5

```
1 library(devtools)
2 #devtools::install_github("wilkoj/treemapify")
3 library(treemapify)
4 library(ggplot2)
5 data(G20)
6 head(G20)
7
8 ggplot(G20, aes(area = gdp_mil_usd, fill = region, label = country)) +
9   geom_treemap() +
10   geom_treemap_text(grow = T, reflow = T, colour = "black") +
11   facet_wrap( ~ econ_classification) +
12   scale_fill_brewer(palette = "Set1") +
13   theme(legend.position = "bottom") +
14   labs(
15     title = "The G-20 major economies",
16     caption = "The area of each country is proportional to its relative GDP
17     within the economic group (advanced or developing)",
18     fill = "Region"
19   )
```



The area of each country is proportional to its relative GDP within the economic group (advanced or developing)

Exemple 6

```

1 data(economics_long, package = "ggplot2")
2 head(economics_long)
3
4 library(ggplot2)
5 library(lubridate)
6 theme_set(theme_bw())
7 df <- economics_long[economics_long$variable %in% c("psavert", "uempmed"), ]
8 df <- df[lubridate::year(df$date) %in% c(1967:1981), ]
9
10 # Labels and breaks for X axis text
11 brks <- df$date[seq(1, length(df$date), 12)]
12 lbls <- lubridate::year(brks)

```

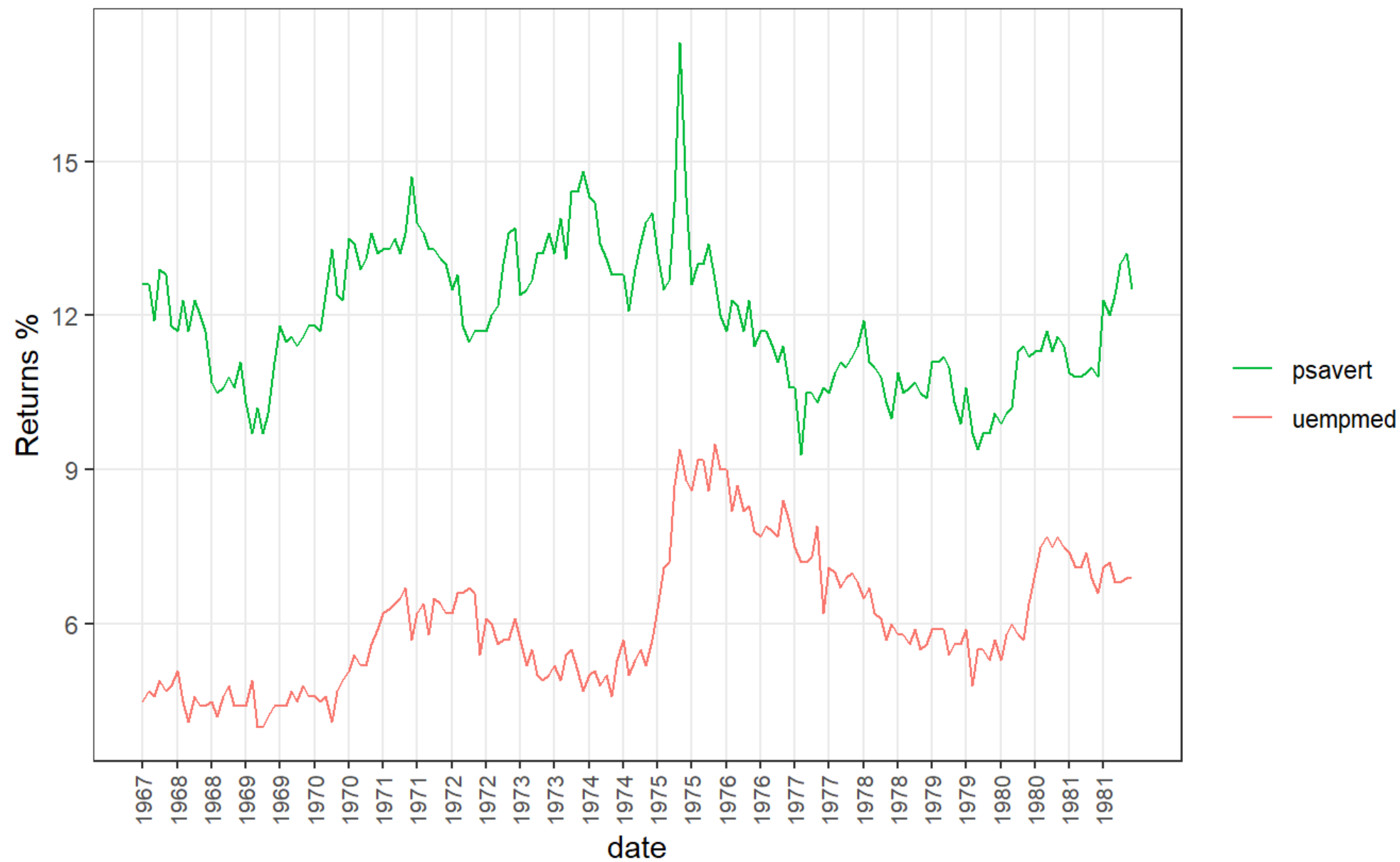
```

15 ggplot(df, aes(x=date)) +
16   geom_line(aes(y=value, col=variable)) +
17   labs(title="Time Series of Returns Percentage",
18         subtitle="Drawn from Long Data format",
19         caption="Source: Economics",
20         y="Returns %",
21         color=NULL) + # title and caption
22   scale_x_date(labels = lbls, breaks = brks) + # change to monthly ticks and
23   scale_color_manual(labels = c("psavert", "uempmed"),
24                      values = c("psavert"="#00ba38", "uempmed"="#f8766d")) +
25   theme(axis.text.x = element_text(angle = 90, vjust=0.5, size = 8), # rotate
26         panel.grid.minor = element_blank()) # turn off minor grid

```

Time Series of Returns Percentage

Drawn from Long Data format



Exemple 7

```

1 library(ggplot2)
2 var <- mpg$class # the categorical data
3 ## Prep data (nothing to change here)
4 nrows <- 10
5 df <- expand.grid(y = 1:nrows, x = 1:nrows)
6 categ_table <- round(table(var) * ((nrows*nrows)/(length(var))))
7 categ_table
8
9 df$category <- factor(rep(names(categ_table), categ_table))
13 ggplot(df, aes(x = x, y = y, fill = category)) +
14   geom_tile(color = "black", size = 0.5) +
15   scale_x_continuous(expand = c(0, 0)) +
16   scale_y_continuous(expand = c(0, 0), trans = 'reverse') +
17   scale_fill_brewer(palette = "Set3") +
18   labs(title="Waffle Chart", subtitle="'Class' of vehicles",
19         caption="Source: mpg") +

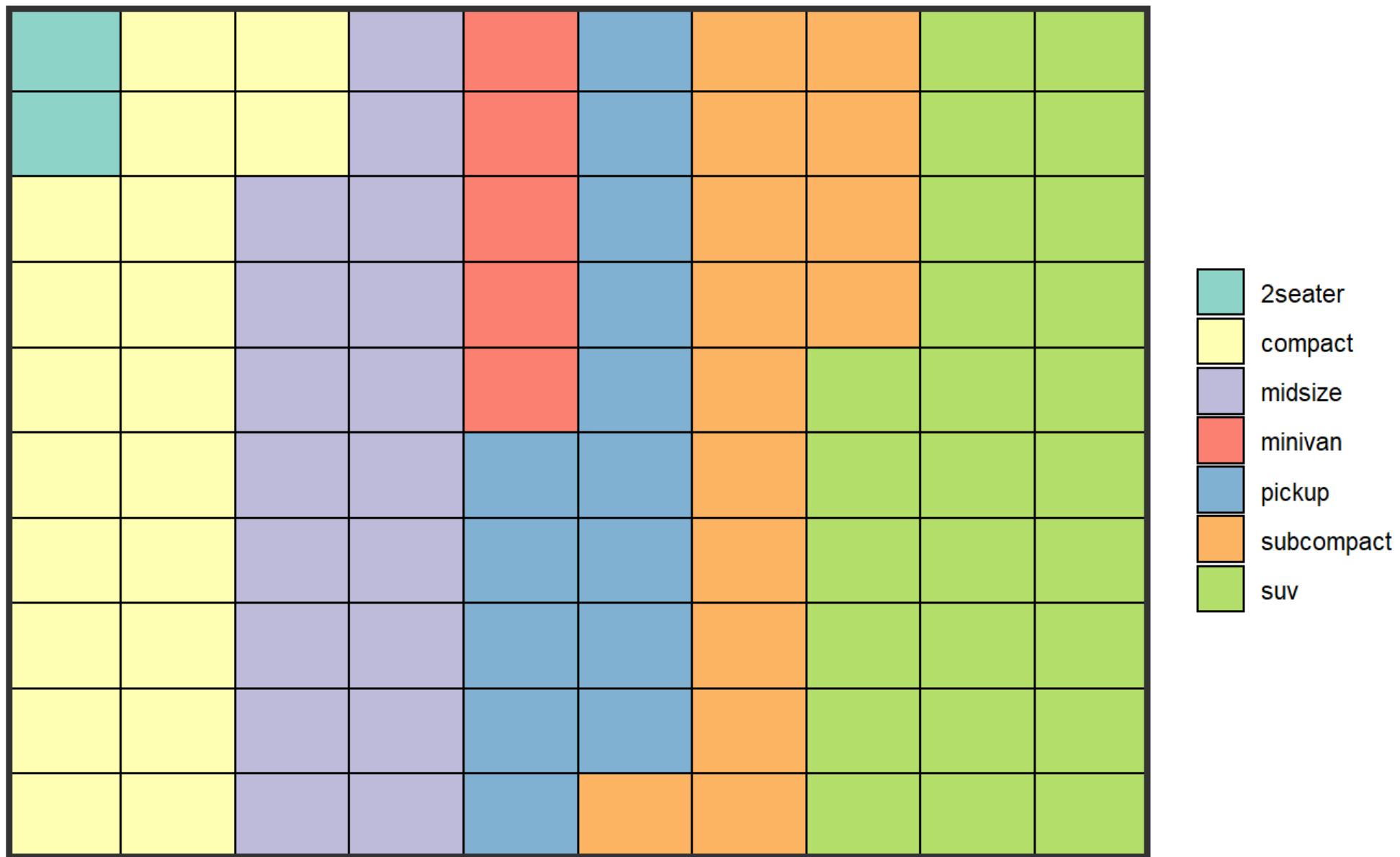
```

```

20   theme(panel.border = element_rect(size = 2),
21         plot.title = element_text(size = rel(1.2)),
22         axis.text = element_blank(),
23         axis.title = element_blank(),
24         axis.ticks = element_blank(),
25         legend.title = element_blank(),
26         legend.position = "right")

```

'Class' of vehicles



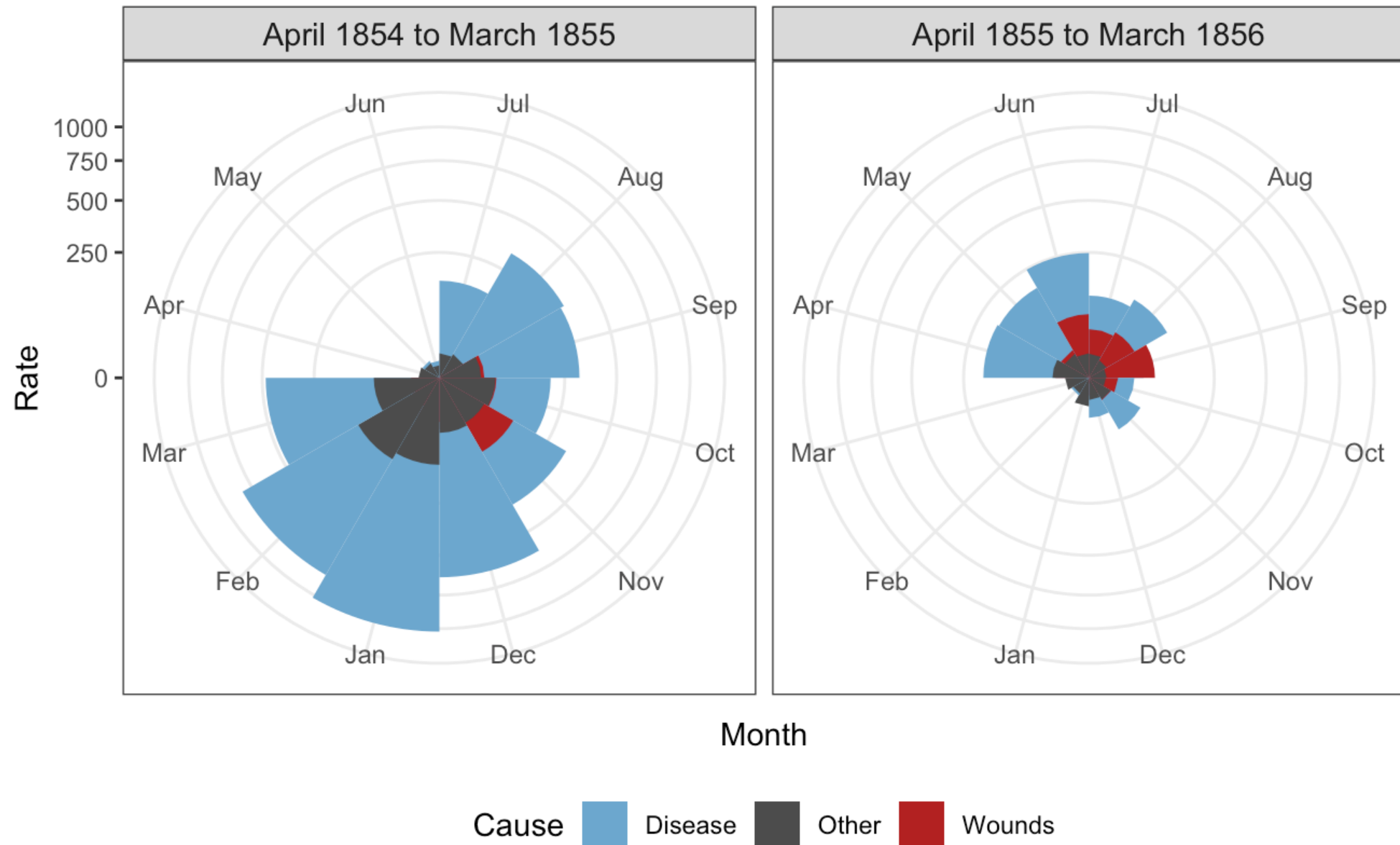
Exemple 8

```

1 library(tidyverse)
2 library(HistData)
3
4 Nightingale %>%
5   select(Date, Month, Year, contains("rate")) %>%
6   pivot_longer(cols = 4:6, names_to = "Cause", values_to = "Rate") %>%
7   mutate(Cause = gsub(".rate", "", Cause),
8          period = ifelse(Date <= as.Date("1855-03-01"), "April 1854 to March 1855", "April 1855 to March 1856"),
9          Month = fct_relevel(Month, "Jul", "Aug", "Sep", "Oct", "Nov", "Dec", "Jan", "Feb", "Mar", "Apr", "May", "Jun")) %>%
10  ggplot(aes(Month, Rate)) +
11    geom_col(aes(fill = Cause), width = 1, position = "identity") +
12    coord_polar() +
13    facet_wrap(~period) +
14    scale_fill_manual(values = c("skyblue3", "grey30", "firebrick")) +
15    scale_y_sqrt() +
16    theme_bw() +
17    theme(axis.text.x = element_text(size = 9),
18          strip.text = element_text(size = 11),
19          legend.position = "bottom",
20          plot.margin = unit(c(10, 10, 10, 10), "pt"),
21          plot.title = element_text(vjust = 5)) +
22    ggtitle("Diagram of the Causes of Mortality in the Army in the East")

```

Diagram of the Causes of Mortality in the Army in the East



Exemple 9

```

1 library(tidyverse)
2 gapminder_ds <- dslabs::gapminder # for the dataset
3 library(wesanderson) # for the colour palette
4 library(ggrepel) # for country names on chart
5 yr <- 2009
6 chart_title <- paste("Health & Wealth of Nations \nGapminder (",yr,""),sep="")
7 # sort the countries by inverse population
8 gapminder_ds <- gapminder_ds |>
9   dplyr::arrange(year, dplyr::desc(population))
10
11 # select which countries will have their names labelled
12 num_countries <- 20
13
14 # tidyverse + ggplot2
15 filtered_gapminder <- gapminder_ds %>%
16   filter(year==yr) %>%
17   mutate(pop_m = population/1e6, gdppc=gdp/population)
18
19 weights <- filtered_gapminder$pop_m/sum(filtered_gapminder$pop_m)
20 p <- sample(nrow(filtered_gapminder), num_countries, prob = weights)
21
22 filtered_gapminder$country_display <- ifelse(ifelse(1:185 %in% p, TRUE,FALSE),as.character(filtered_gapminder$country),"")

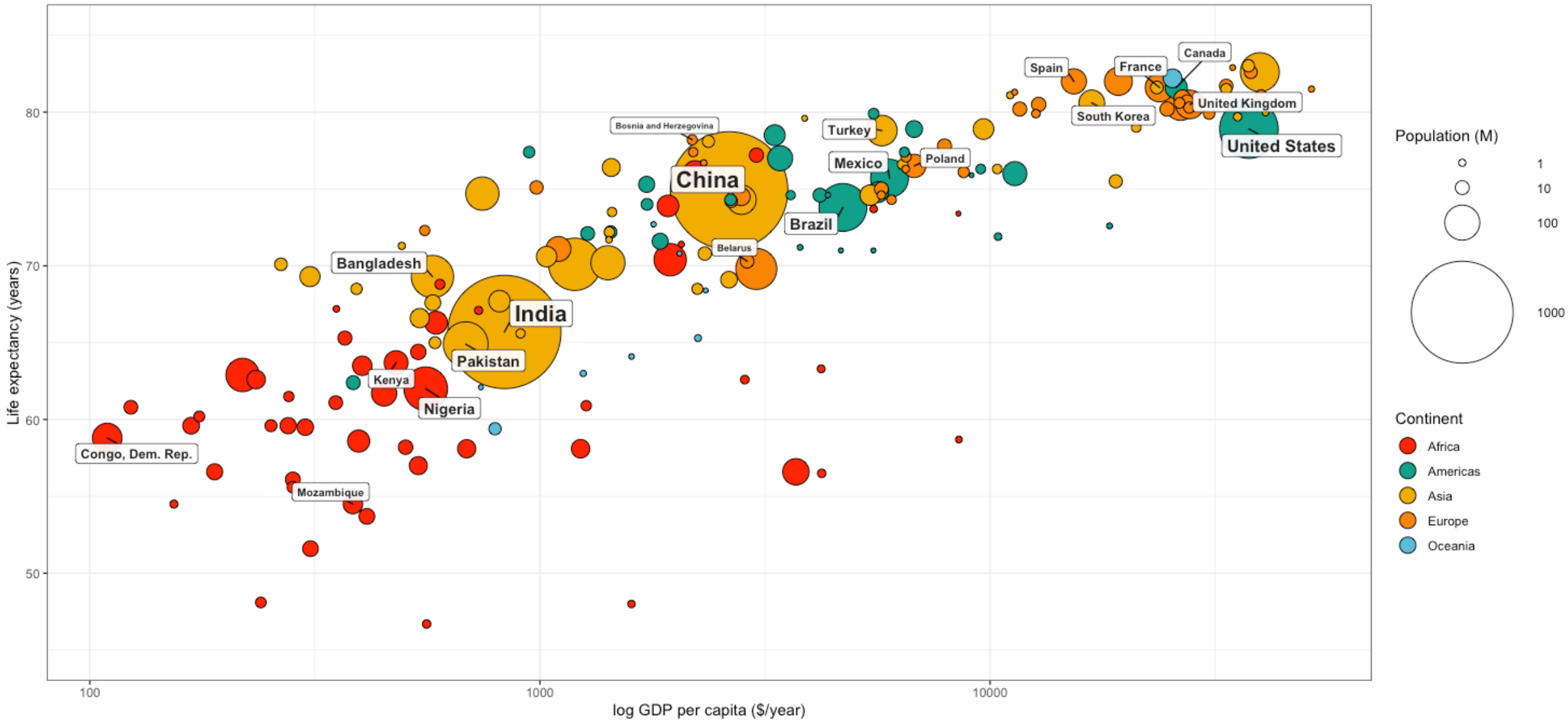
```

```

1 filtered_gapminder %>%
2   ggplot(aes(x = gdppc,
3             y=life_expectancy,
4             size=pop_m)) +
5   geom_point(aes(fill=continent), pch=21) +
6   scale_fill_manual(values=wes_palette(n=5, name="Darjeeling1")) +
7   scale_x_log10() +
8   geom_label_repel(aes(label=country_display, size=sqrt(pop_m/pi)),
9                   alpha=0.9,
10                  fontface="bold",
11                  min.segment.length = unit(0, 'lines'),
12                  show.legend=FALSE) +
13   ggtitle(chart_title) +
14   theme(plot.title = element_text(size=14, face="bold")) +
15   xlab('log GDP per capita ($/year)') +
16   ylab('Life expectancy (years)') +
17   ylim(45,85) +
18   scale_size_continuous(range=c(1,40),
19                         breaks = c(1,10,100,1000),
20                         limits = c(0, 1500),
21                         labels = c("1","10","100","1000")) +
22   guides(fill = guide_legend(override.aes = list(size = 5))) +
23   labs(fill="Continent", size="Population (M)") +
24   theme_bw() +
25   theme(plot.title = element_text(size=16, face="bold"))

```

Health & Wealth of Nations Gampinder (2011)



Lectures conseillées

Exemples et curiosités de `ggplot2`

The Practice of Data Visualization

Part IV: Some Practical Aspects of Data Visualization

12. `ggplot2` Visualizations in R

12.3 Examples

Exercices

Exemples et curiosités de `ggplot2`

(Essayez cet exercice sans consulter au préalable l'exemple des données PUMS du recensement des États-Unis dans DUDADS).

Le fichier `custdata.tsv` est dérivé des données PUMS du recensement des États-Unis. Cet ensemble de données synthétiques contient des informations sur certains clients dont le statut d'assurance maladie est connu.

Recréez les graphiques `ggplot2` suivants (ignorez les valeurs manquantes pour l'instant).

Session 4

