Optimiser ses graphiques avec R

Jérôme Sueur

MNHN – Systématique et Evolution
UMR CNRS 7205 – OSEB
sueur@mnhn.fr

28 Avril 2011

Outline

Typologie – Quels graphiques ?

- simples et composés
- 2D ou 3D
- statiques, interactifs, animés

Typologie – Familles

Quatre familles de graphiques issus de la base ou de packages spécifiques :

- base (Ross Ihaka)
- package `grids` (Paul Murrel)
- package `lattice` (Deepayan Sarkar)
- package `ggplot2` (Hadley Wickman)

Outline

Base – Principes

Cinq grands types de fonctions graphiques

- haut-niveau (high-level)
- bas-niveau (low-level)
- paramétrage (parameters)
- division (treillis, faceting)
- impression (vectorielle, matricielle)

Base – Fonctions de haut-niveau

- nuages de points
- profils
- graphiques en bâtons
- histogrammes
- boîtes à moustaches
- graphiques en violons
- camemberts
- radars
- contours
- cartes de densité
- cartes géographiques
- images
- etc

Base – Fonctions de bas-niveau

- points, traits, surfaces (rectangles, polygones, cercles)
- lignes, courbes, segments, flèches
- titres, légendes, étiquettes

Base – Fonctions de paramétrage

- couleurs
- taille, police, orientation
- échelles
- marges
- superposition
- etc

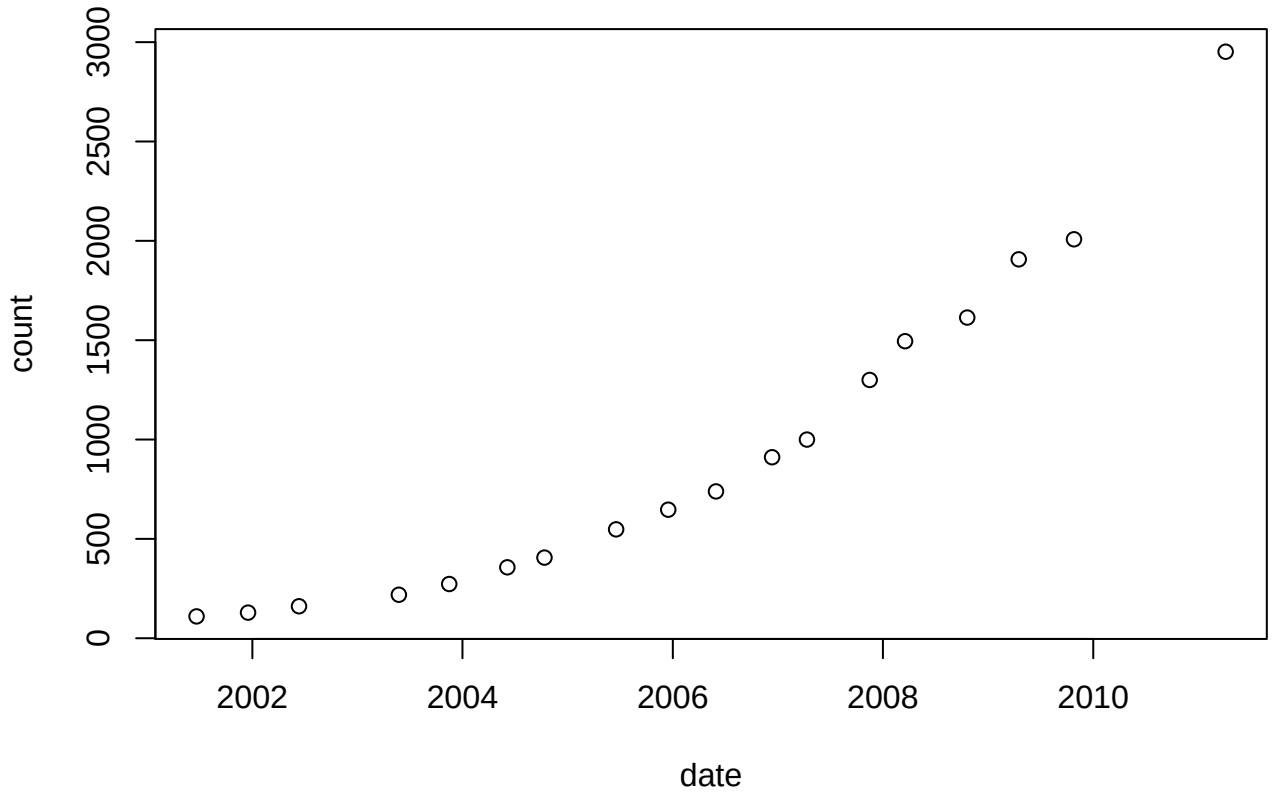
Base – Nombre de packages

```
data
```

	vers	count	date
1	1.30	110	2001-06-21
2	1.40	129	2001-12-17
3	1.50	161	2002-06-12
4	1.70	219	2003-05-25
5	1.80	273	2003-11-16
6	1.90	357	2004-06-05
7	2.00	406	2004-10-12
8	2.10	548	2005-06-18
9	2.20	647	2005-12-16
10	2.30	739	2006-05-31
11	2.40	911	2006-12-12
12	2.50	1000	2007-04-12
13	2.60	1300	2007-11-16
14	2.70	1495	2008-03-18
15	2.80	1614	2008-10-20
16	2.90	1907	2009-04-17
17	2.10	2008	2009-10-26
18	2.12	2952	2011-04-06

Base – Exemple

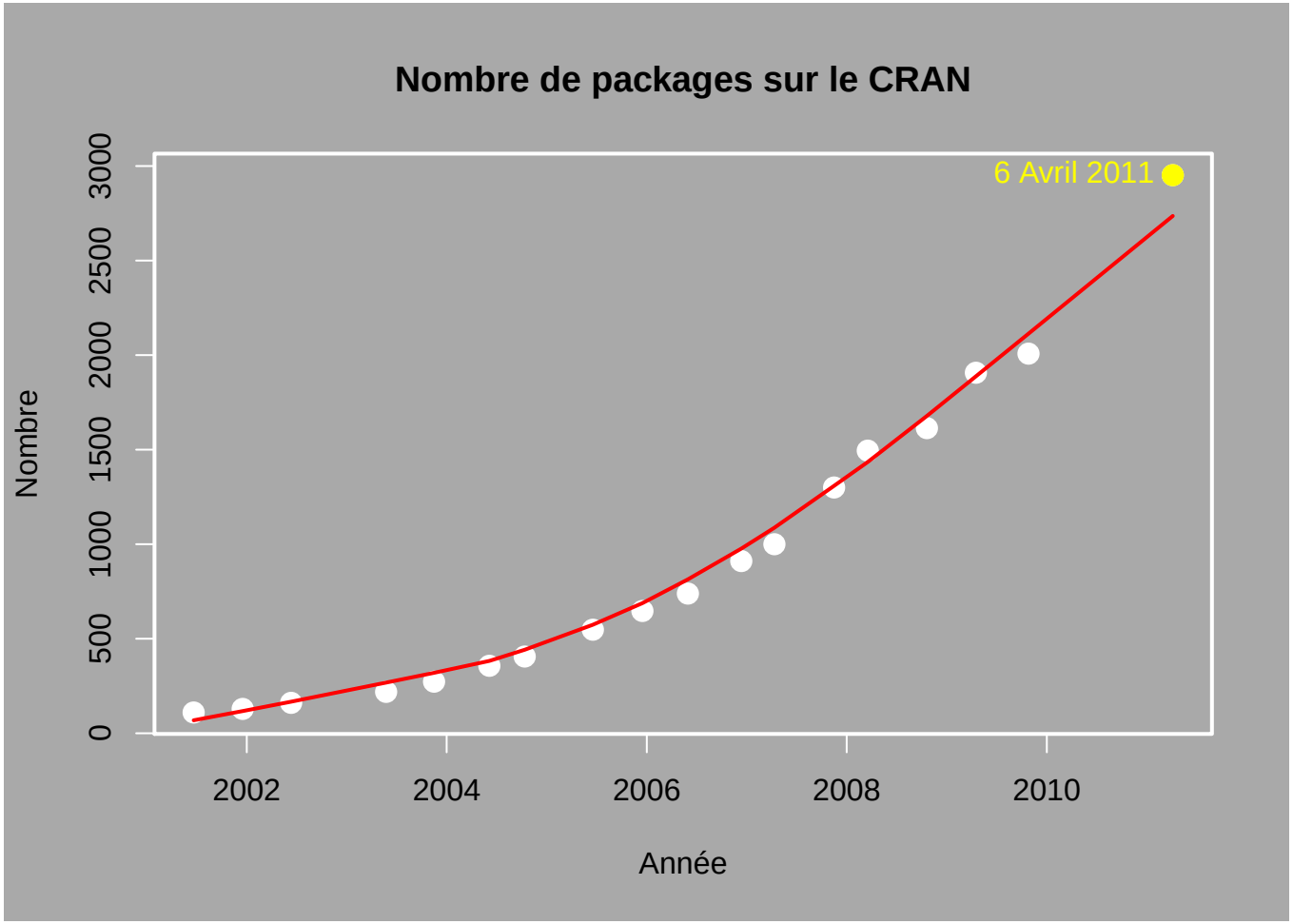
```
plot(date, count)
```



Base – Exemple

```
par(bg = "darkgrey", fg = "white", lwd = 2)
plot(date, count, pch = 20, cex = 2, col = "white", xlab = "Année",
    ylab = "Nombre", main = "Nombre de packages sur le CRAN")
points(date[18], count[18], pch = 20, col = "yellow",
    cex = 2)
text(date[18], count[18], pos = 2, label = "6 Avril 2011",
    col = "yellow")
lines(lowess(date, count), col = 2, lwd = 2)
```

Base – Exemple



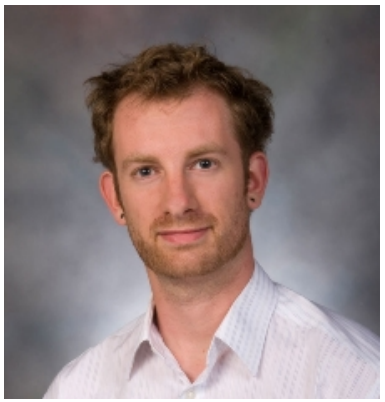
Base – Des exemples par centaines

http ://addictedtor.free.fr/graphiques/index.php

Outline

ggplot2 – Historique

- développé par Hadley Wickham (Rice University, Houston, USA)
- dépend principalement des packages `reshape` et `plyr`
- première version : Juin 2007



ggplot2 – Principes généraux

- obéit à de nouveaux principes esthétiques

ggplot2 – Principes généraux

- obéit à de nouveaux principes esthétiques
- obéit à une construction particulière suivant la grammaire graphique développée par Wilkinson en 2005

ggplot2 – Principes généraux

- obéit à de nouveaux principes esthétiques
- obéit à une construction particulière suivant la grammaire graphique développée par Wilkinson en 2005
- permet une construction rapide de graphiques simples

ggplot2 – Principes généraux

- obéit à de nouveaux principes esthétiques
- obéit à une construction particulière suivant la grammaire graphique développée par Wilkinson en 2005
- permet une construction rapide de graphiques simples
- réduction (considérable) de la longueur des codes

ggplot2 – Principes généraux

- obéit à de nouveaux principes esthétiques
- obéit à une construction particulière suivant la grammaire graphique développée par Wilkinson en 2005
- permet une construction rapide de graphiques simples
- réduction (considérable) de la longueur des codes
- permet la création de nouvelles familles de graphiques

ggplot2 – Principes généraux

- obéit à de nouveaux principes esthétiques
- obéit à une construction particulière suivant la grammaire graphique développée par Wilkinson en 2005
- permet une construction rapide de graphiques simples
- réduction (considérable) de la longueur des codes
- permet la création de nouvelles familles de graphiques
- **Nouveau dialecte à apprendre**

ggplot2 – Les calques ou layers

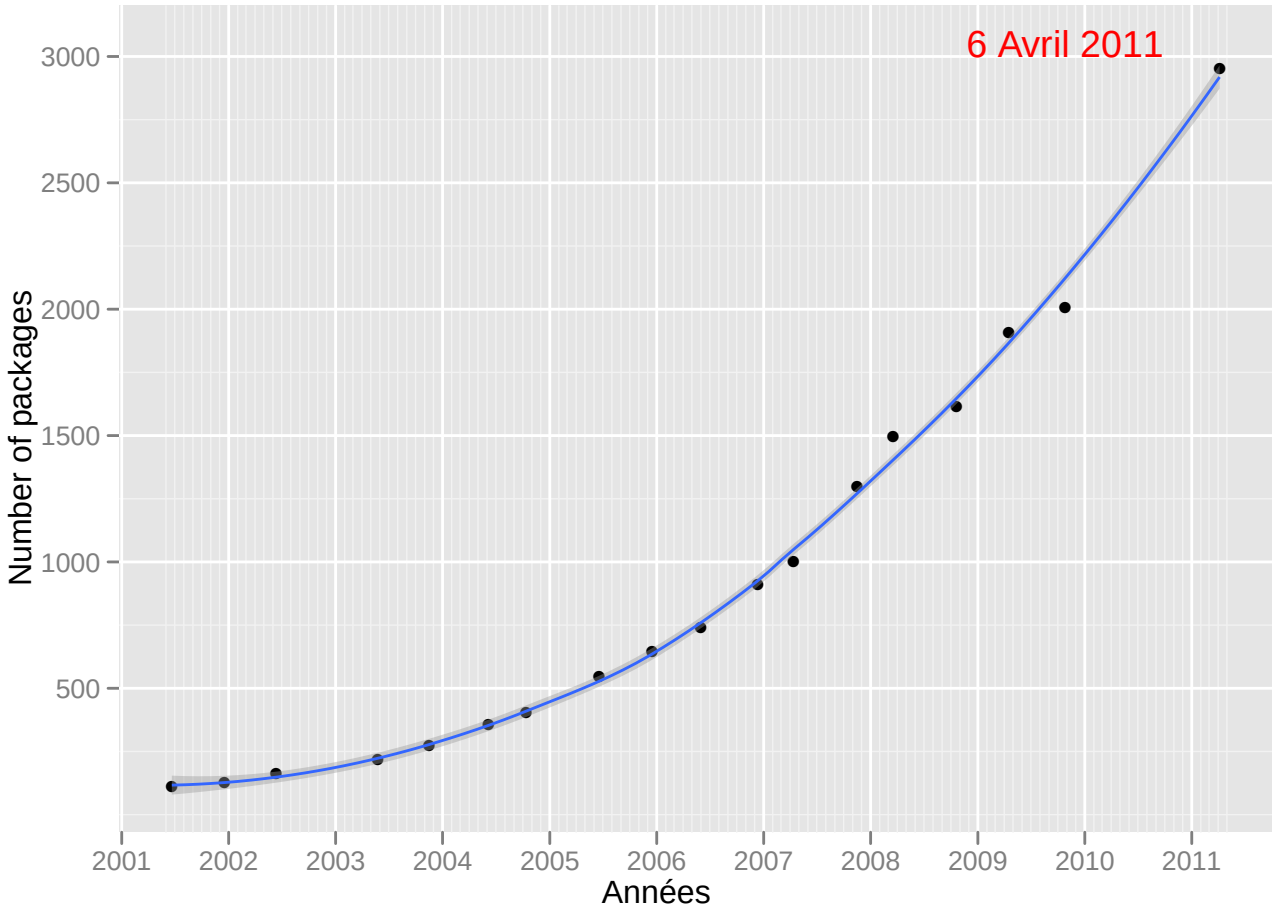
`ggplot2` créé des **layers** (calques) qui peuvent s'utiliser comme des objets (assignation possible).

Voici les principaux `layers` :

data	→	données brutes
mapping	→	projection graphique
geom	→	objets géométriques (points, lignes, polygones, etc.)
stat	→	transformation statistique (histogramme, modèle, etc.)
scale	→	espace esthétique (couleurs, formes, tailes, axes, légendes)
coord	→	système de coordonnées (axes, grilles)
facet	→	division

ggplot2 – Nombre de packages

```
plot <- qplot(date, count, data = data, geom = c("point",
    "smooth"))
plot + xlab("Années") + ylab("Number of packages") +
    annotate("text", x = data[17, 3], y = data[18, 2] +
        100, label = "6 Avril 2011", colour = "red")
```



ggplot2 – Fonctions de base

`ggplot2` a deux fonctions graphiques de base :

- `qplot()` pour **q**uick **plot**
  - rapide mais simple (pour un seul jeu de données et une seule méthode esthétique)
  - principe :

```
qplot(x, y, data=data)
```

- `ggplot()`
  - lent mais plus puissant, ajout de `layers` avec `+`
  - principe :

```
gplot(data, aes(x, y)) + layers
```

Remarque

On peut obtenir les mêmes résultats avec les deux fonctions (équivalence)

ggplot2 – Objets géométriques (geom)

point	→	scatterplot (défaut si 2 dimensions)
smooth	→	ajoute une courbe de tendance (lissage) à un scatterplot
boxplot	→	boîte à moustaches
jitter	→	gigue
path	→	chemins entre des points ($\forall$ direction)
line	→	lignes entre des points (de gauche à droite)
histogram	→	histogramme (défaut si 1 dimension)
freqpoly	→	polygone fréquentiel
density	→	courbe de densité
bar	→	bâtons

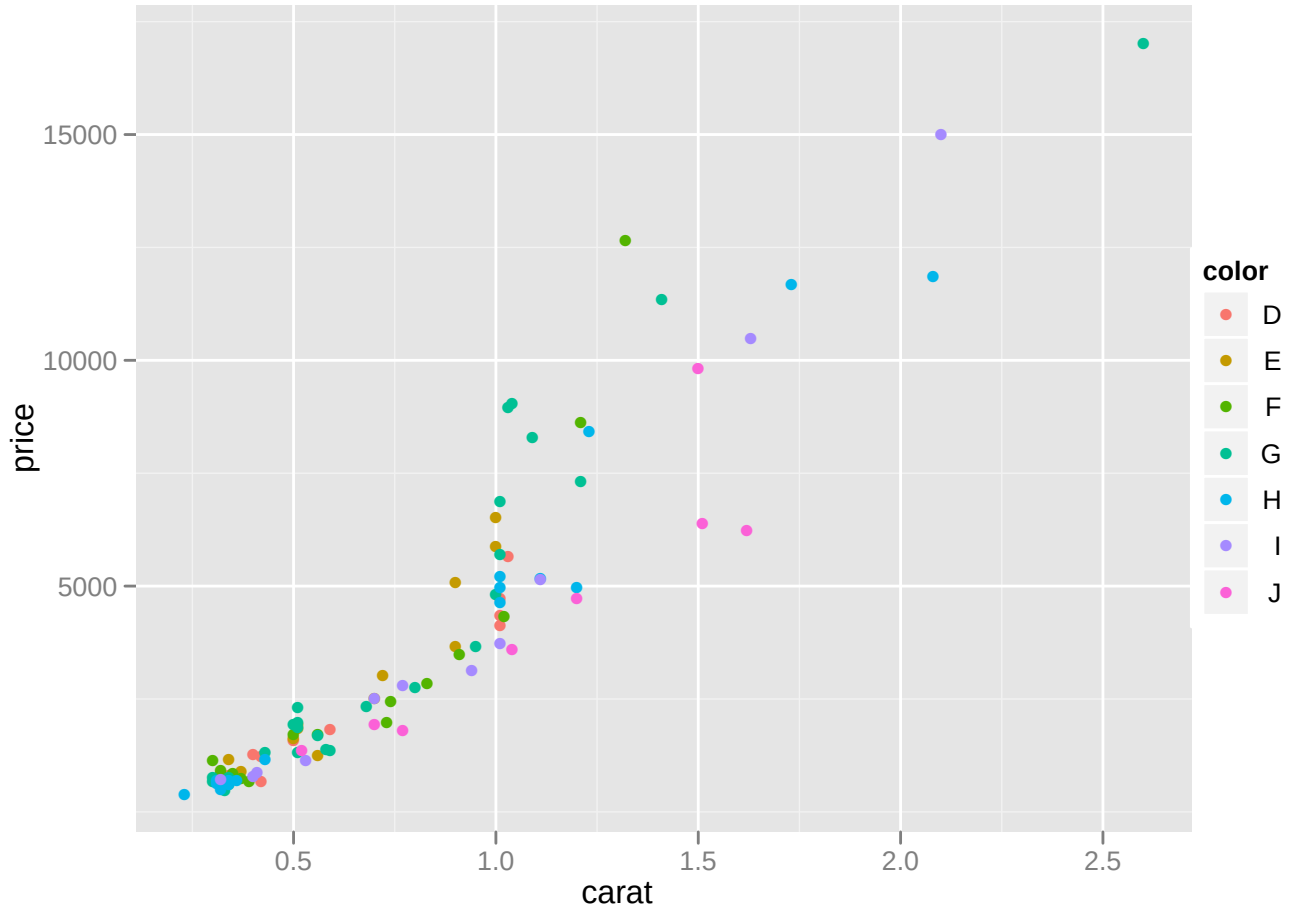
ggplot2 – Catégorisation par la couleur

```
head(diamonds)
```

	carat	cut	color	clarity	depth	table	price	x	y	z
13998	1.01	Ideal	G	SI1	62.6	56	5698	6.37	6.41	4.00
301	0.77	Ideal	I	VS1	61.5	59	2798	5.87	5.91	3.62
6816	1.01	Fair	D	SI1	66.3	55	4118	6.22	6.17	4.11
48416	0.73	Premium	F	SI2	60.2	59	1971	5.87	5.82	3.52
20812	1.04	Premium	G	IF	61.3	58	9039	6.52	6.43	3.97
36091	0.32	Very Good	F	VVS1	61.9	59	926	4.34	4.39	2.70

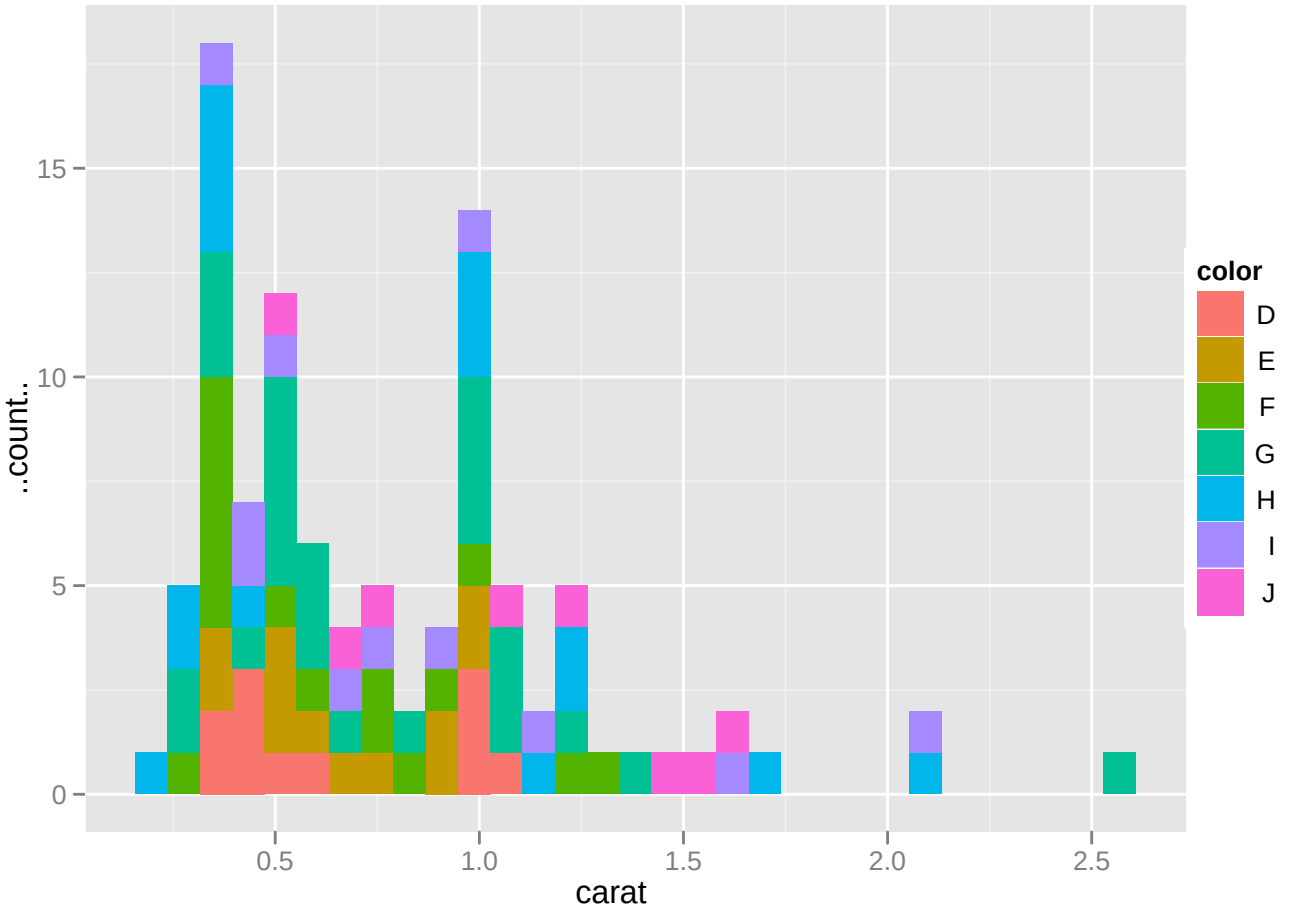
ggplot2 – Catégorisation par la couleur

```
qplot(carat, price, data = diamonds, colour = color)
```



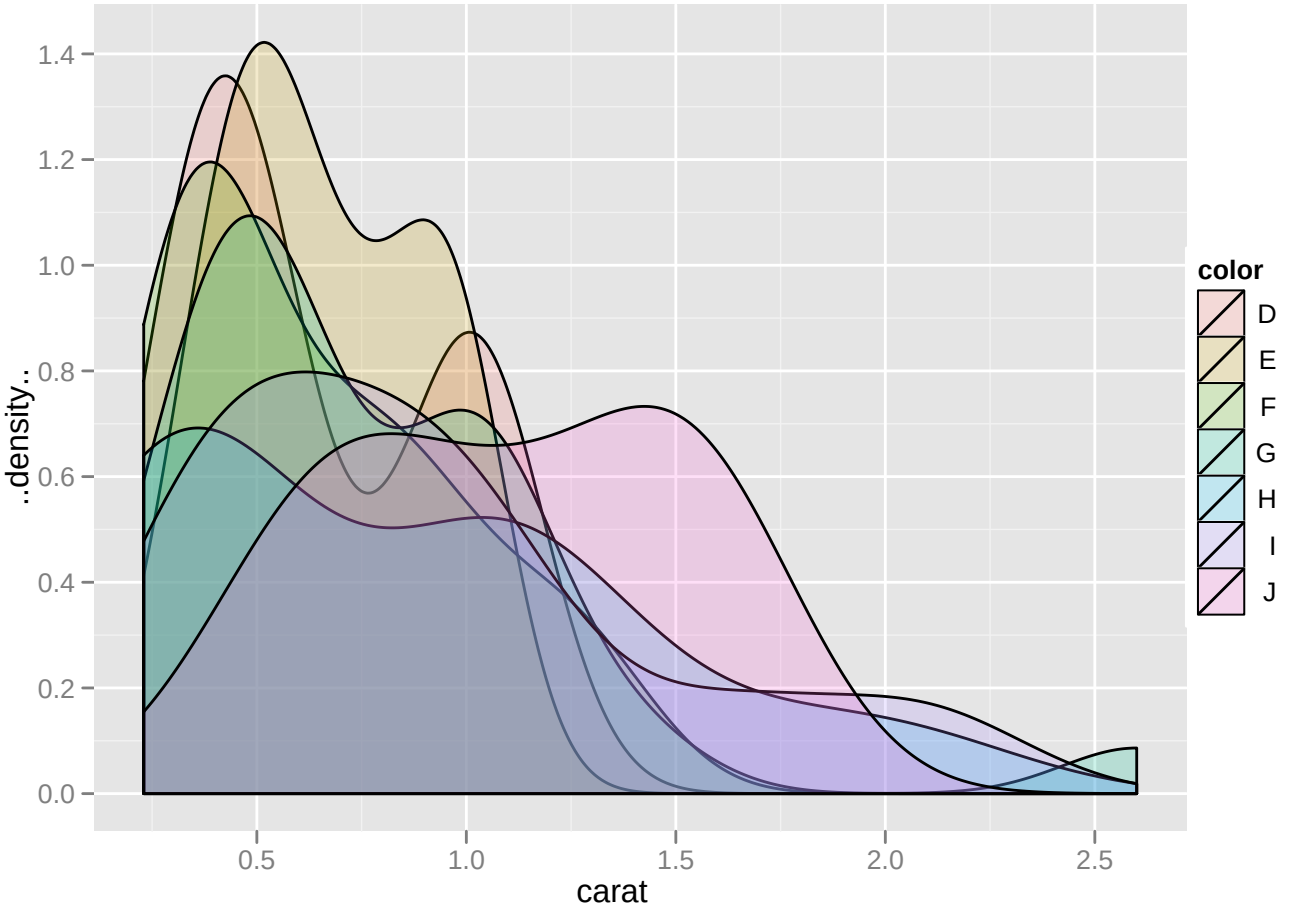
ggplot2 – Catégorisation par la couleur

```
qplot(carat, data = diamonds, geom = "histogram", fill = color)
```



ggplot2 – Catégorisation par la couleur

```
qplot(carat, data = diamonds, geom = "density", fill = color,
    alpha = I(1/5))
```

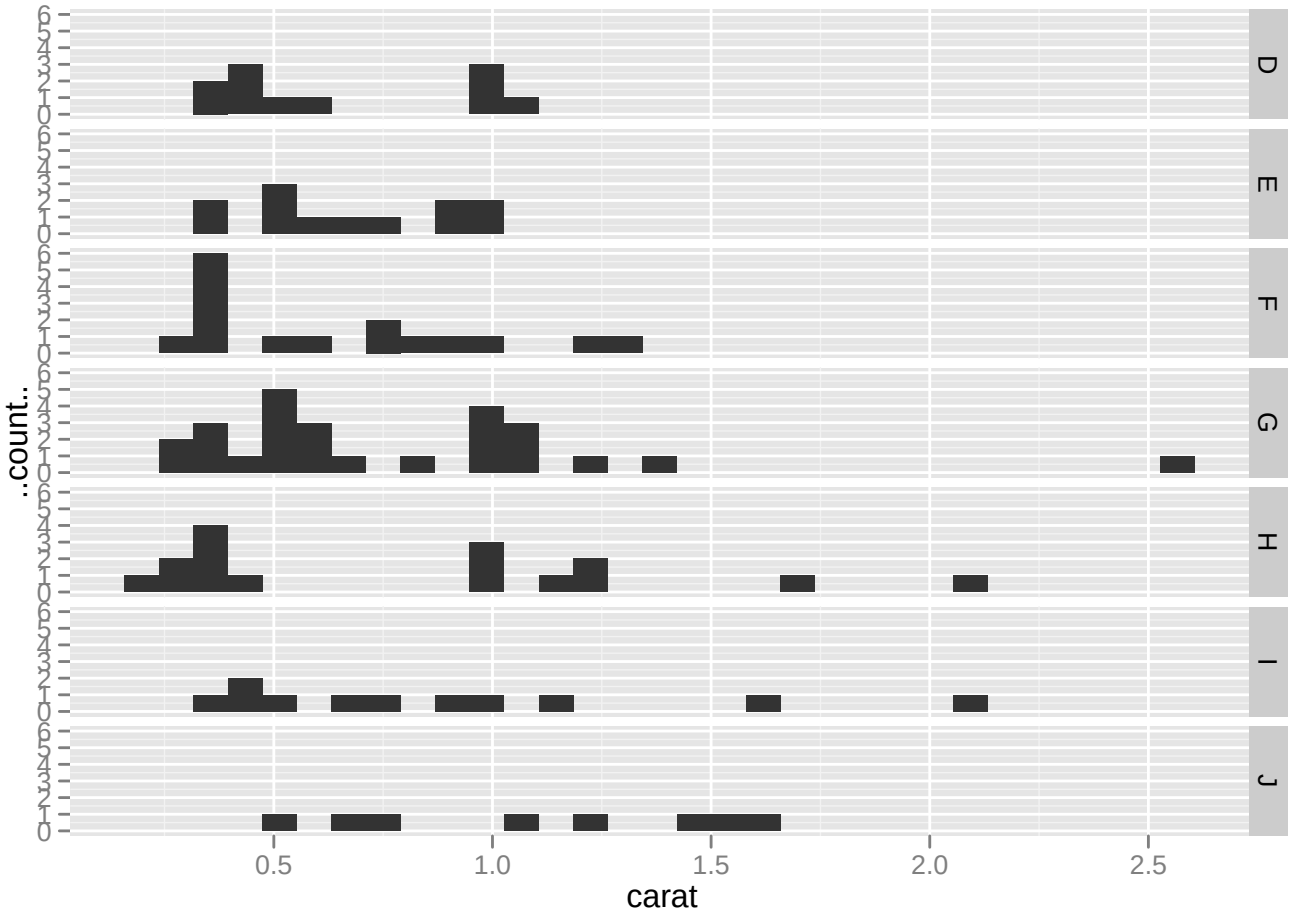


ggplot2 – Division

- **Principe** : diviser le jeu de données en sous-ensembles et créer un graphique similaire pour chacun sous-ensemble.
- **Syntaxe** : argument `facets` de `qplot` selon facteur en ligne $\sim$ facteur en colonne
  - `facets = facteur ~ .` produira un graphique multiple en colonnes
  - `facets = . ~ facteur` produira un graphique multiple en lignes

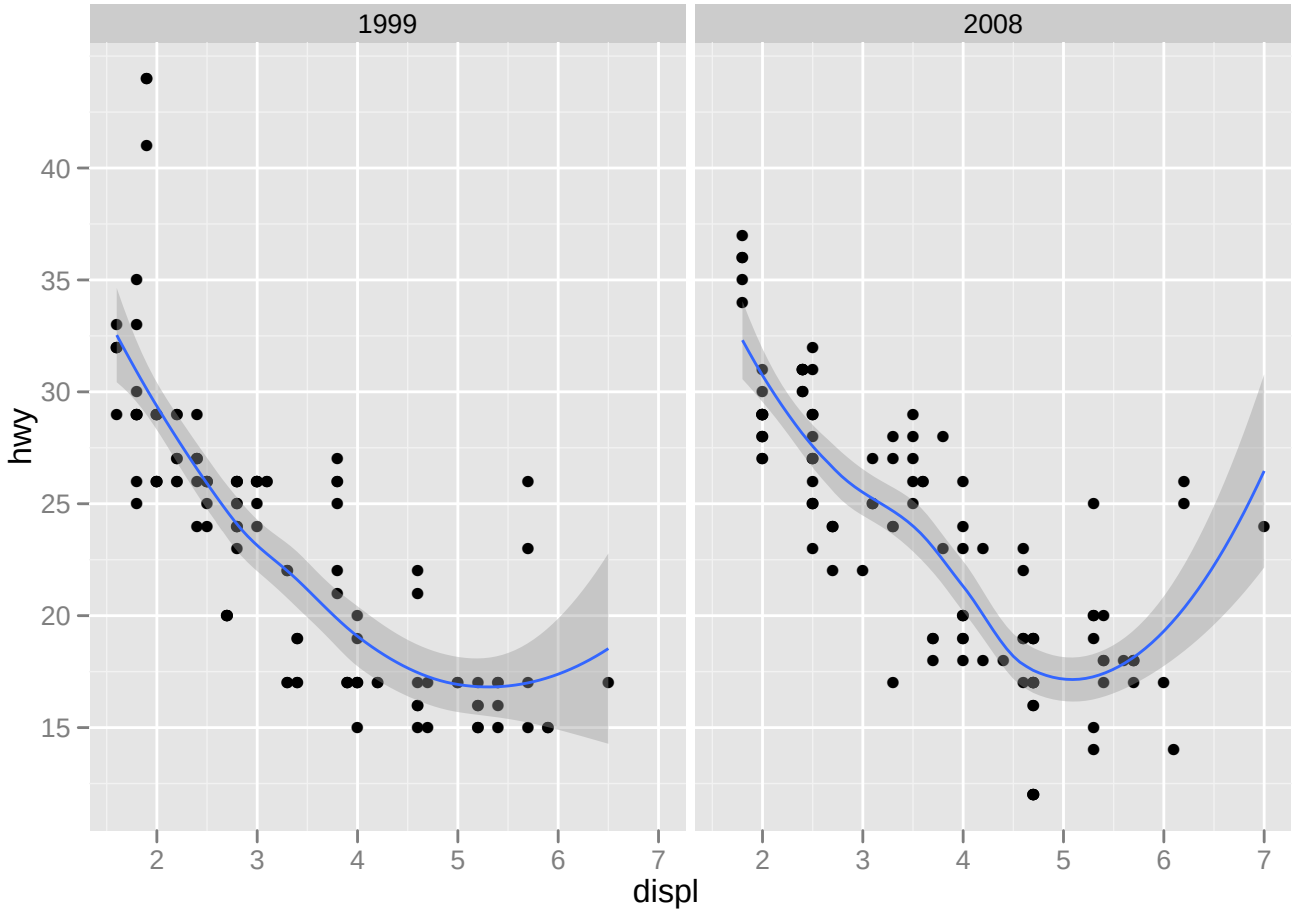
ggplot2 – Division

```
qplot(carat, data = diamonds, facets = color ~ ., geom = "histogram")
```



ggplot2 – Division

```
data(mpg)
qplot(displ, hwy, data = mpg, geom = c("point", "smooth"),
    facets = . ~ year)
```



ggplot2 – Arguments

`ggplot` permet, en association avec de produire des graphes plus complexes.

`ggplot` a deux arguments :

- **data** (données dans un `data.frame`)
- **aesthetic** mapping (paramètres esthétiques de la projection, doivent être inclus dans la fonction `aes()`)

ggplot2 – Calques

Voici les principaux `layers` :

- `geom_XXX()` (par example : `geom_point()`, `geom_histogramm()`, etc.)
- `stat_XXX()` (par exemple : `stat_bin()`, `stat_smooth()`, etc.)
- `scale_XXX()` (par exemple : `scale_x_continuous()`, `scale_x_log10()`, etc.)
- `facet_XXX()` (par exemple : `facet_grid()`, `facet_wrap()`, etc.)
- `coord_XXX()` (par exemple : `coord_flip()`, `coord_trans()`, etc.)

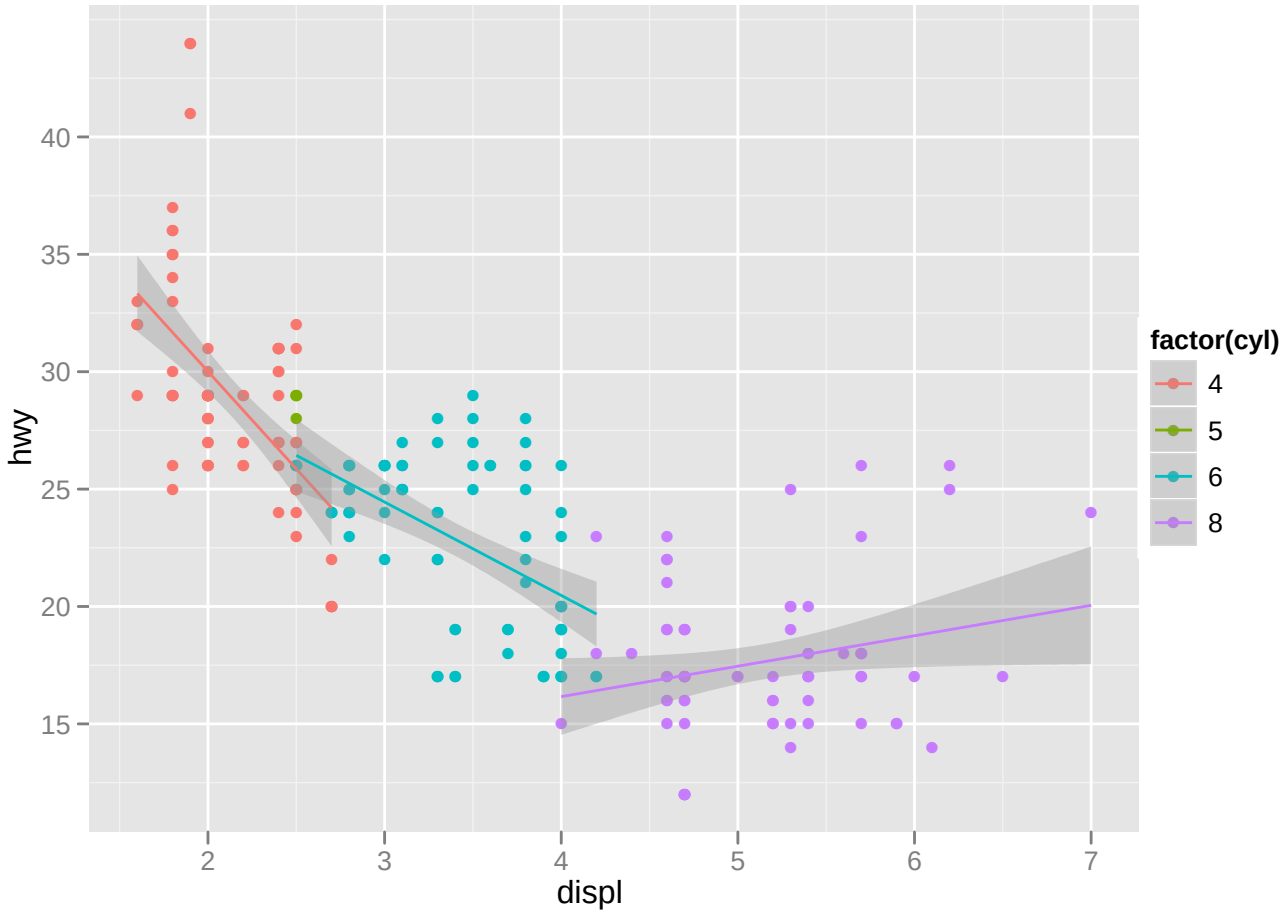
ggplot2 – Premier calque

```
p <- ggplot(mpg, aes(displ, hwy, colour = factor(cyl)))
summary(p)
```

```
data: manufacturer, model, displ, year, cyl, trans, drv, cty,
  hwy, fl, class [234x11]
mapping:  x = displ, y = hwy, colour = factor(cyl)
faceting: facet_grid(. ~ ., FALSE)
```

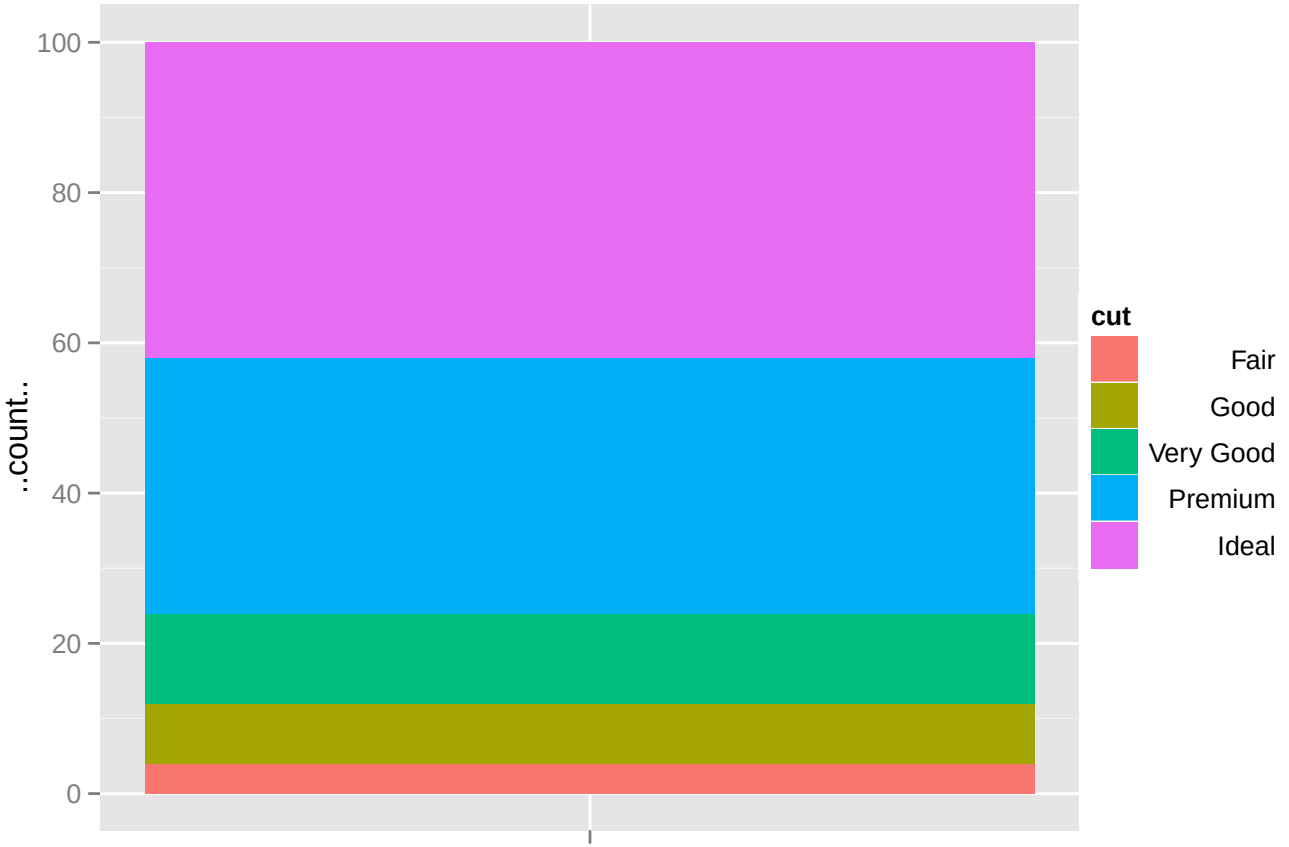
ggplot2 – Exemple d'objet géométrique

```
p + geom_point() + geom_smooth(method = "lm")
```



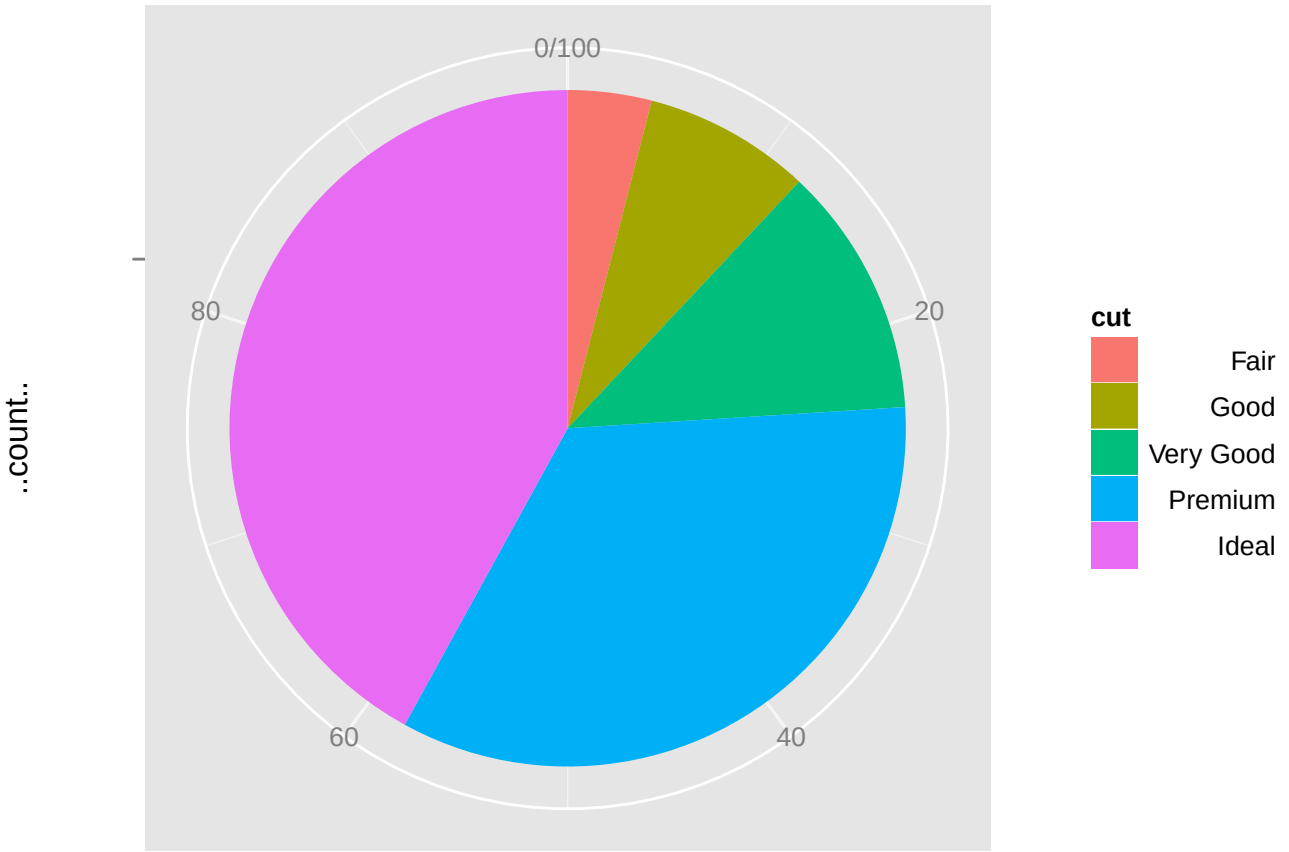
ggplot2 – Systèmes de coordonnées

```
p <- ggplot(diamonds, aes(x = "", fill = cut)) + geom_bar(width = 1)
p
```



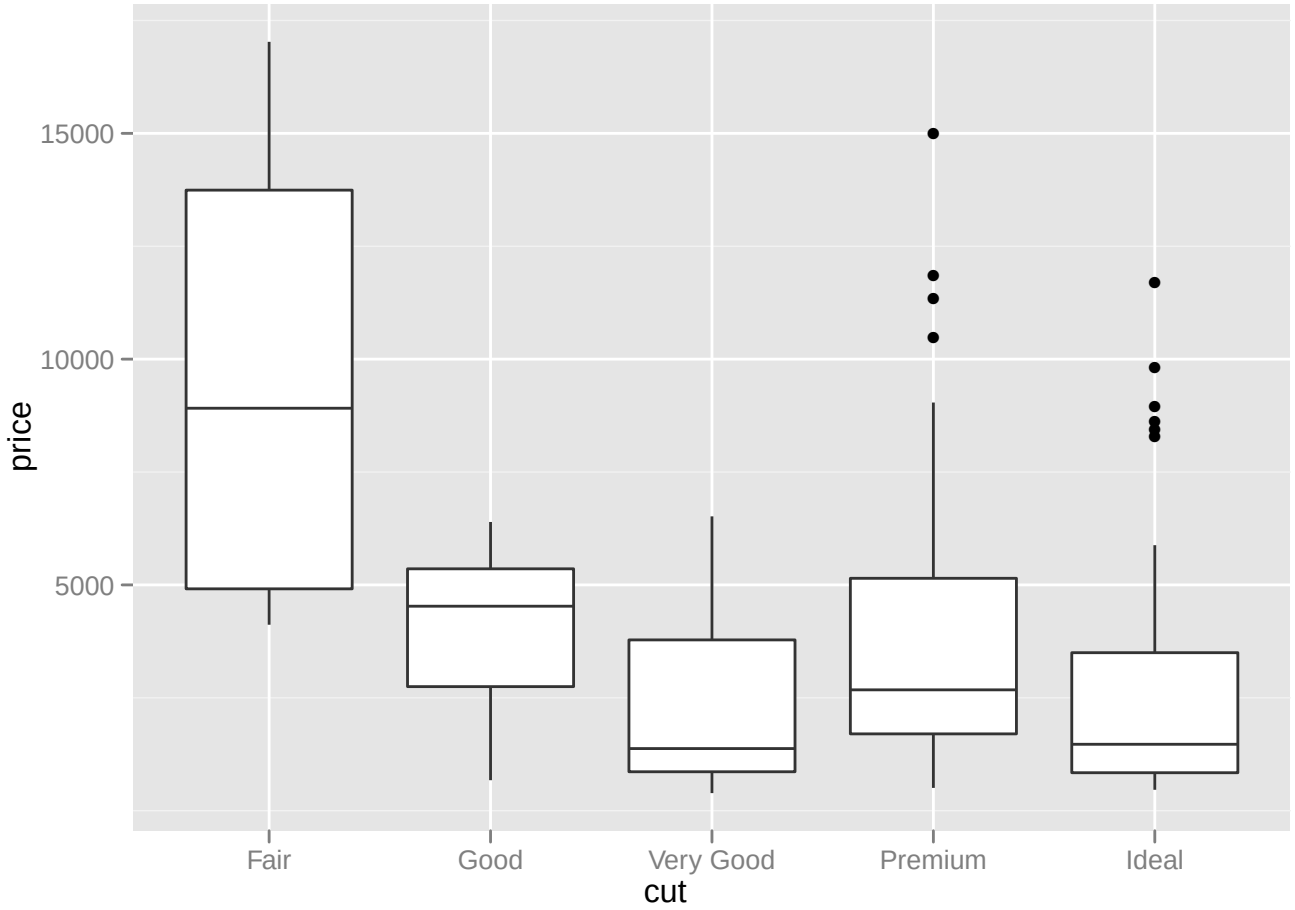
ggplot2 – Systèmes de coordonnées

```
p + coord_polar(theta = "y")
```



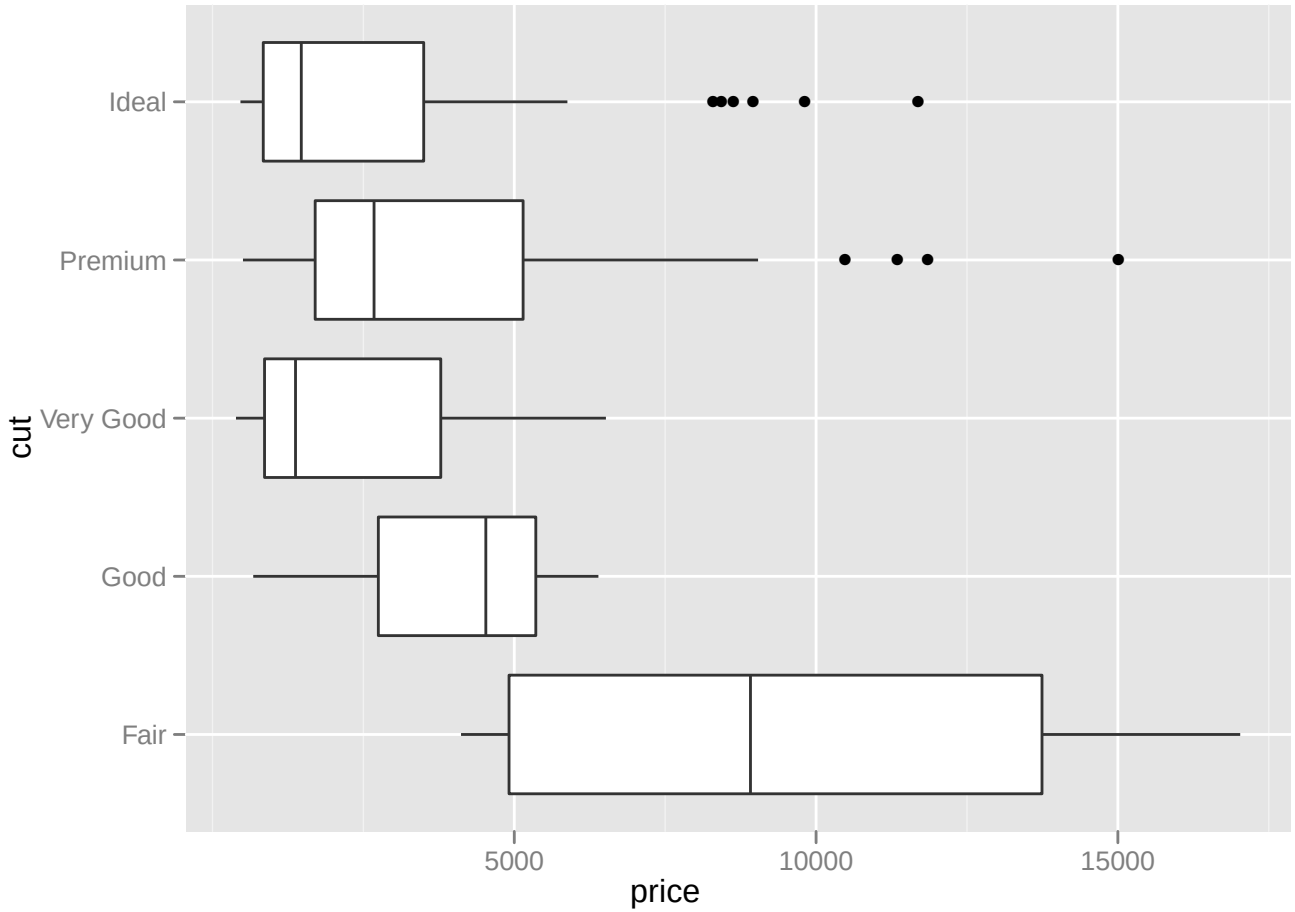
ggplot2 – Systèmes de coordonnées

```
p <- qplot(cut, price, data = diamonds, geom = "boxplot")
p
```



ggplot2 – Systèmes de coordonnées

```
p + coord_flip()
```



ggplot2 – Thèmes

Le **thème** contrôle l'apparence globale (police, couleurs).

Il y a deux thèmes par défaut :

- `theme_gray()` : fond gris, grille blanche
- `theme_bw()` : fond blanc, grille grise

ggplot2 – Thèmes

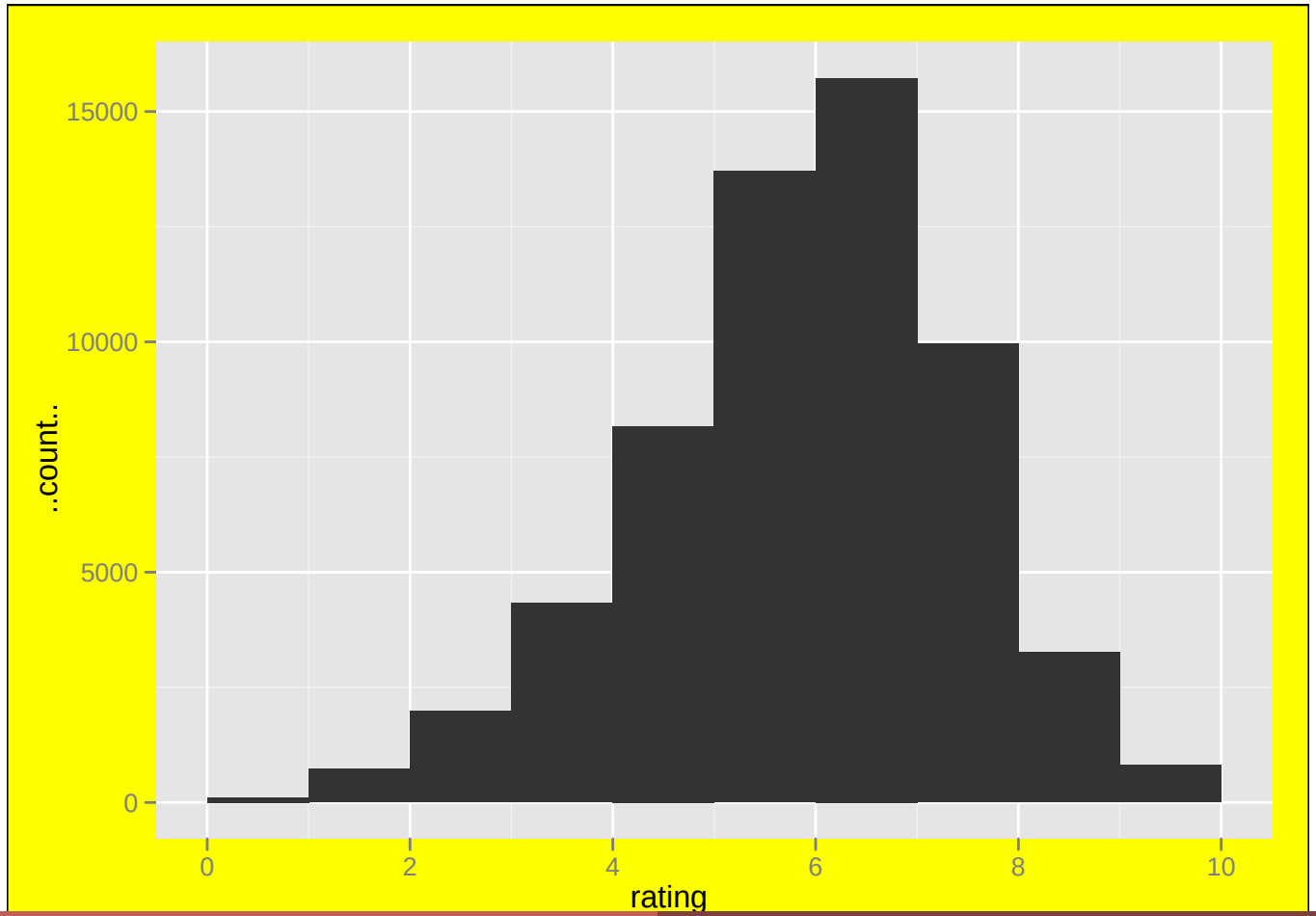
Il y a deux manières de choisir le thème par défaut

- globalement avec `theme_set(theme_XXX)`.
- localement comme un `layer` avec également `theme_set(theme_XXX)`

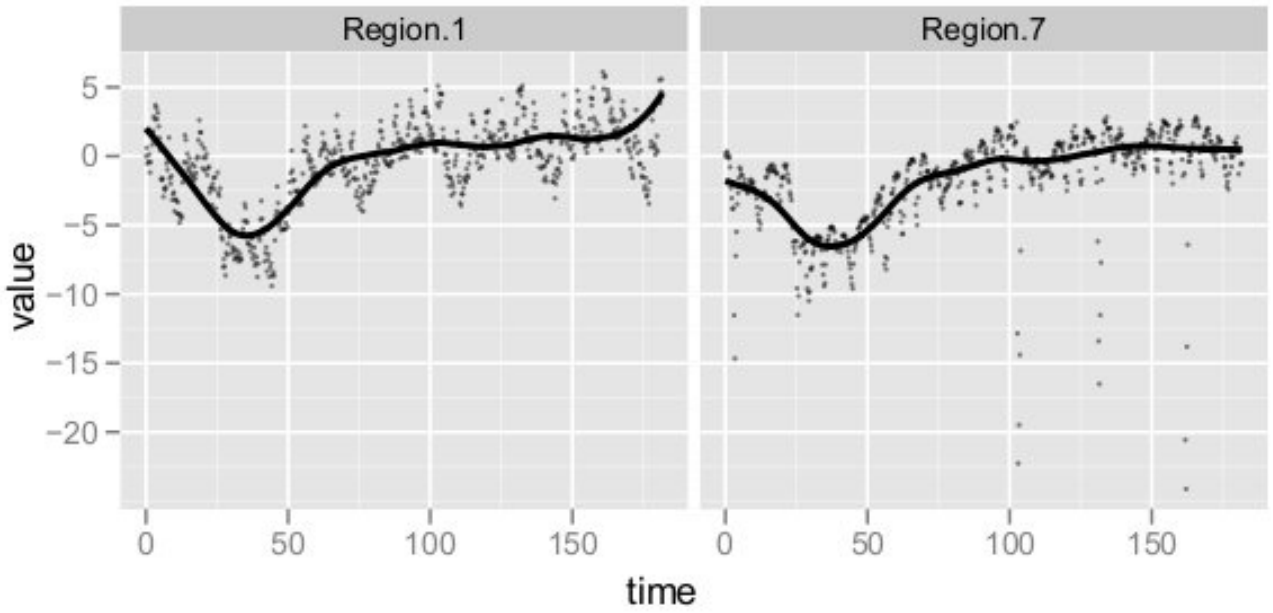
ggplot2 – Thèmes

Modification globale pour tous les graphiques suivants en utilisant `theme_update` :

```
theme_update(plot.background = theme_rect(fill = "yellow"))
data(movies)
qplot(rating, data = movies, binwidth = 1)
```

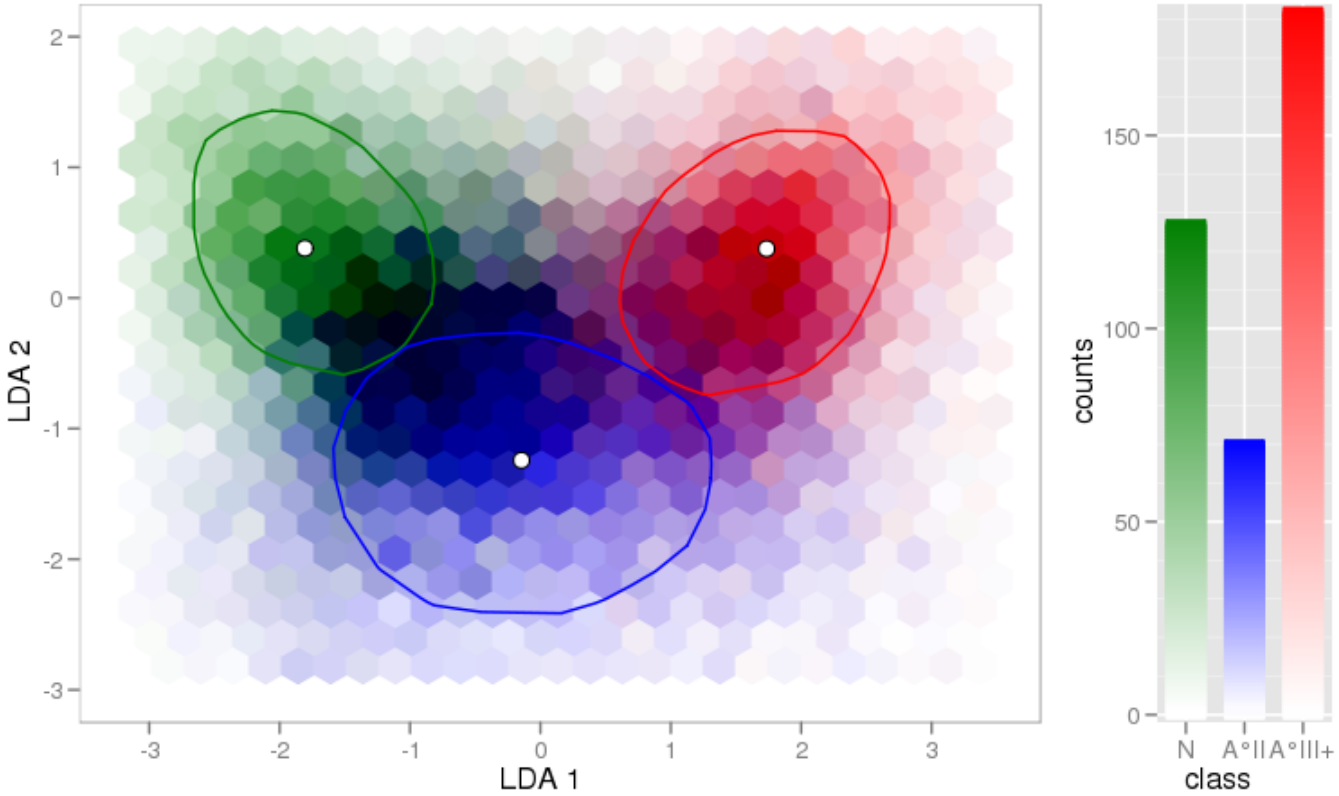


ggplot2 – Exemples



par Michael Lavine (UMass Amherst)
https ://github.com/hadley/ggplot2/wiki

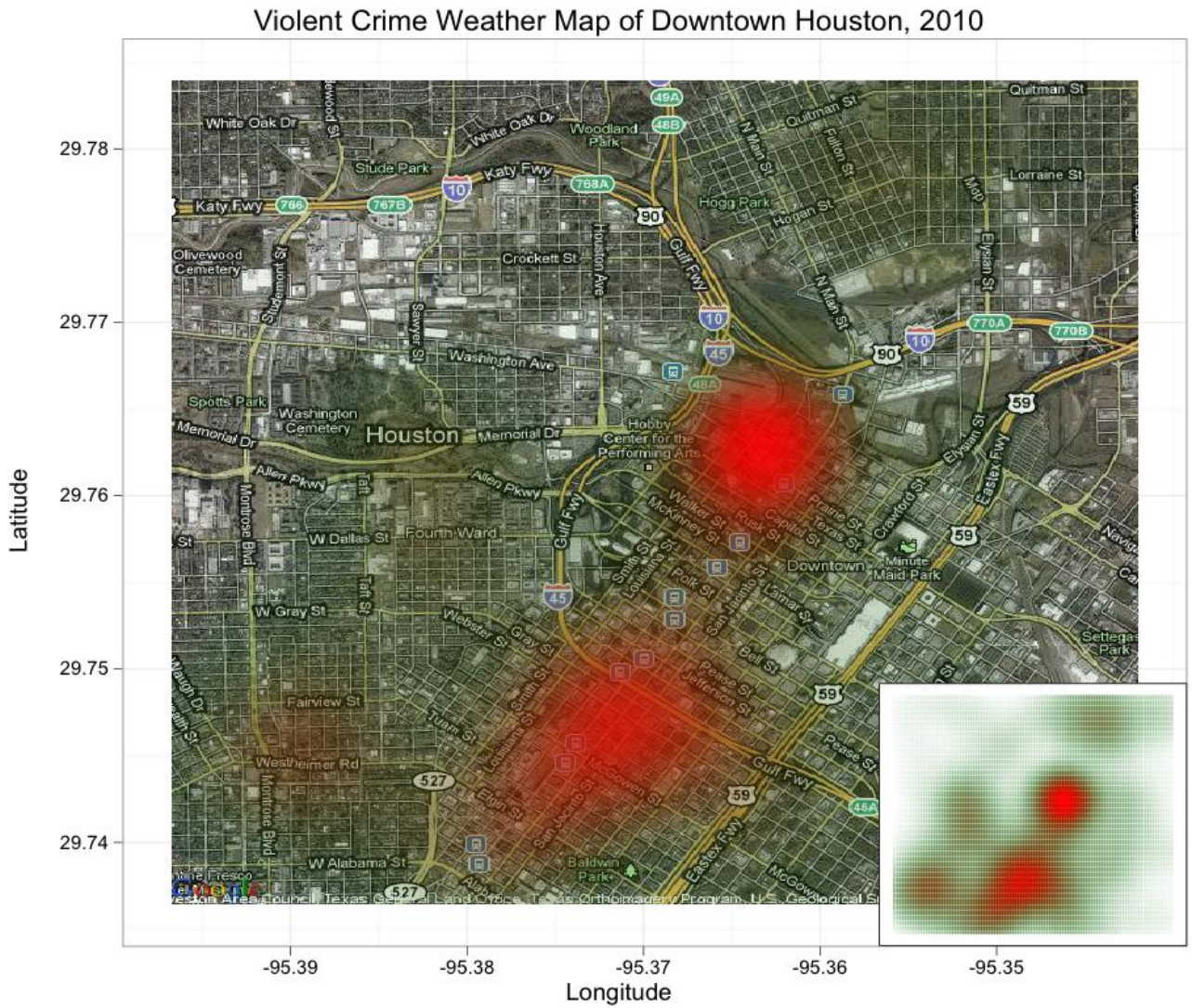
ggplot2 – Exemples



par Claudia Beleites (TU Dresden & Uni. Trieste)

https ://github.com/hadley/ggplot2/wiki

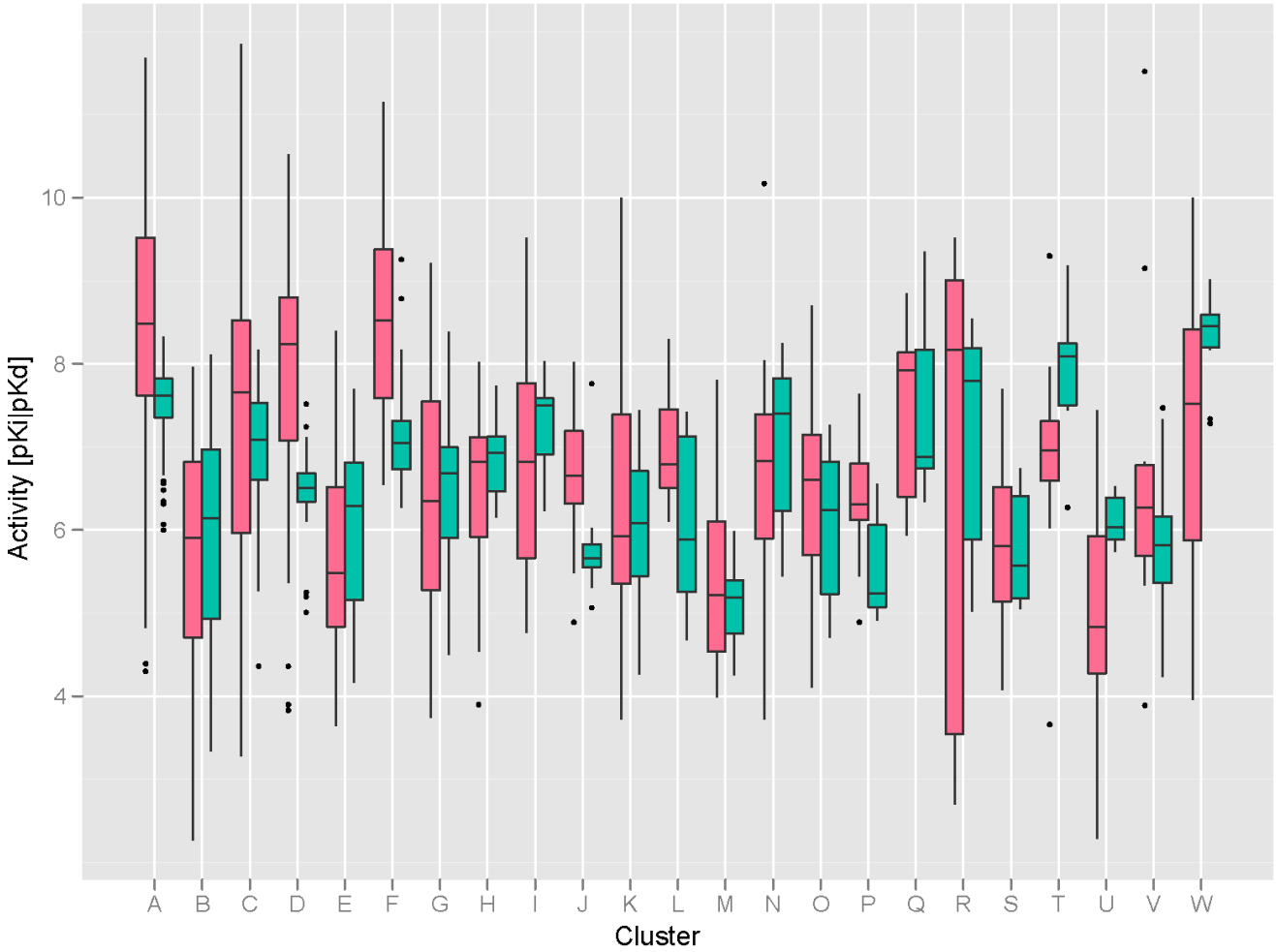
ggplot2 – Exemples



par David Kahle (Rice University)

https ://github.com/hadley/ggplot2/wiki

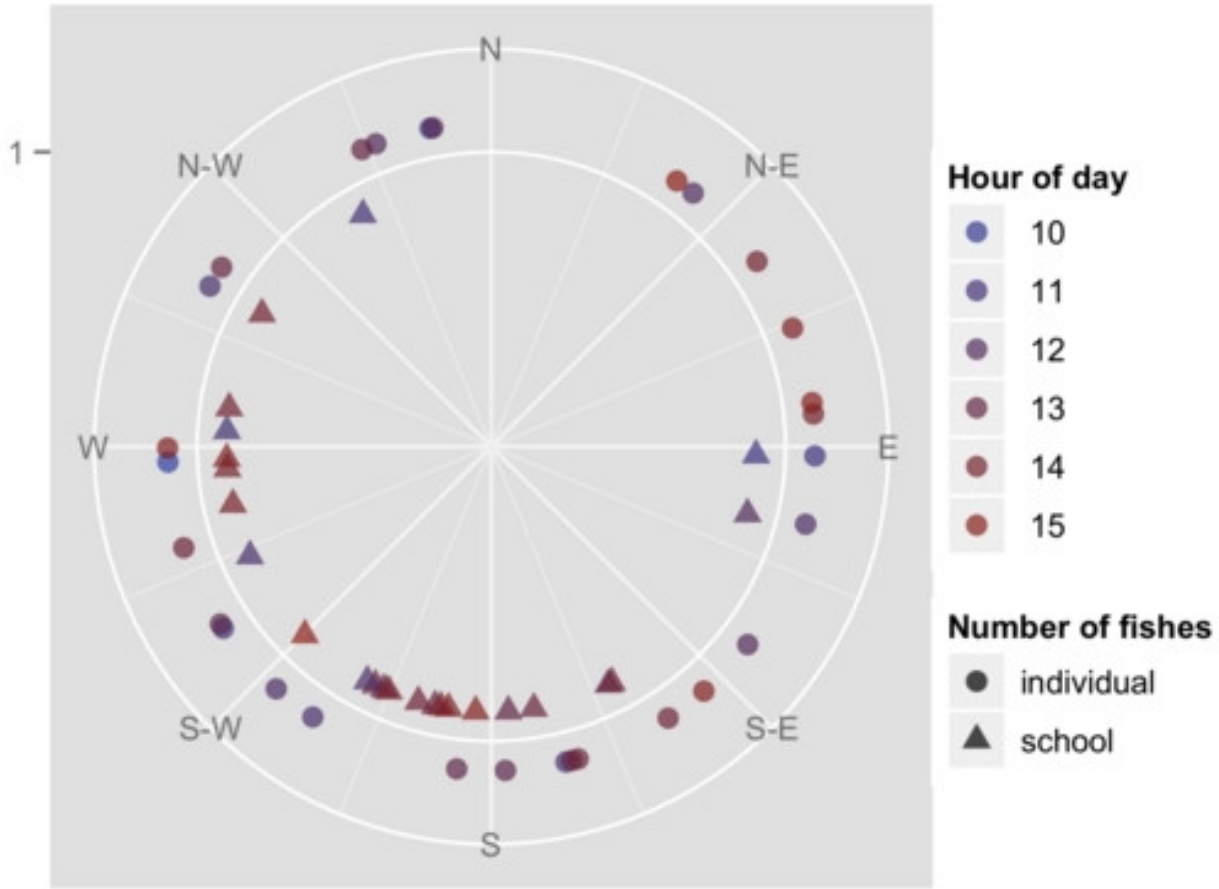
ggplot2 – Exemples



par Christian Kramer (NIBR Basel)

https ://github.com/hadley/ggplot2/wiki

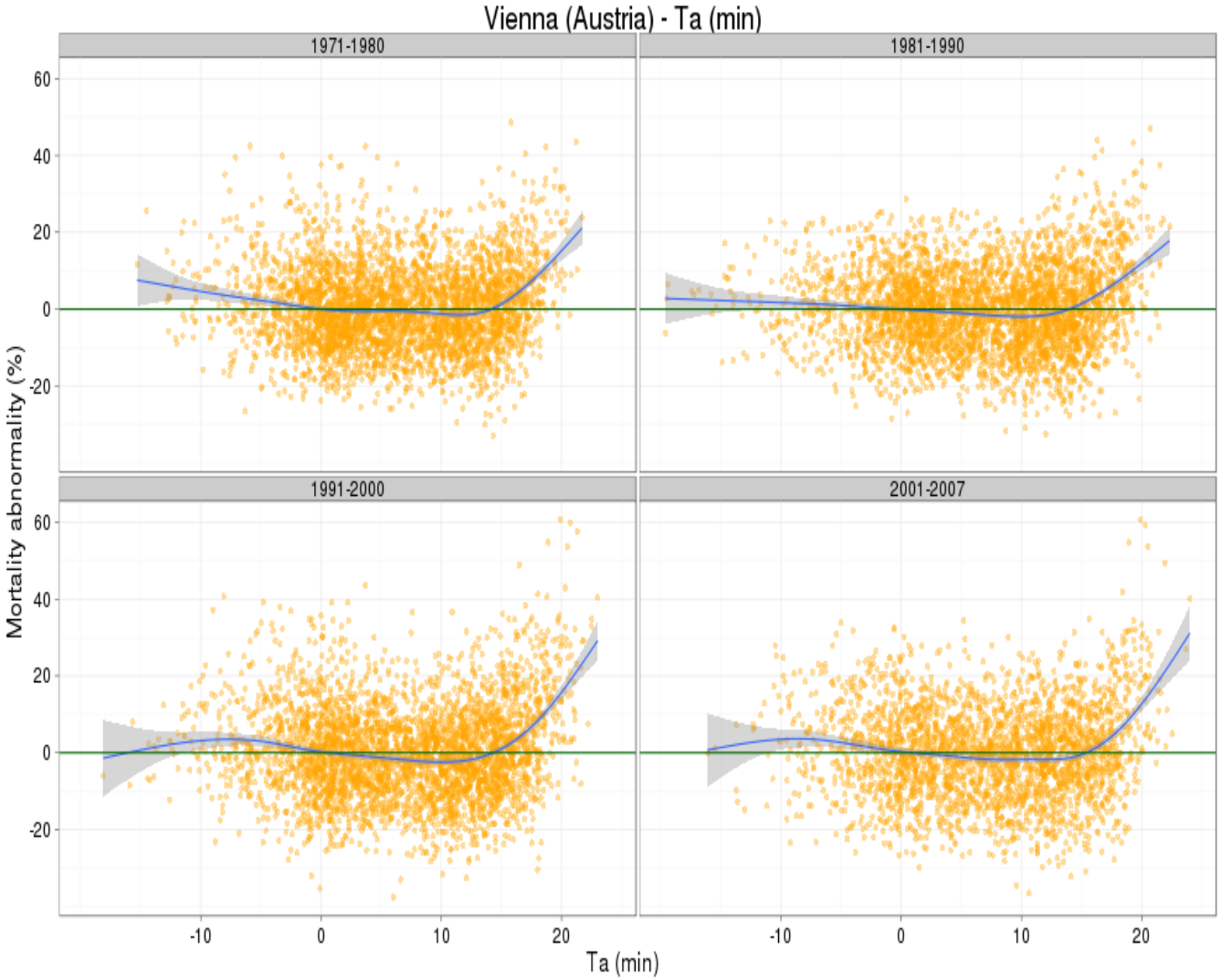
ggplot2 – Exemples



par Jean-Olivier Irisson (JiHO)

https ://github.com/hadley/ggplot2/wiki

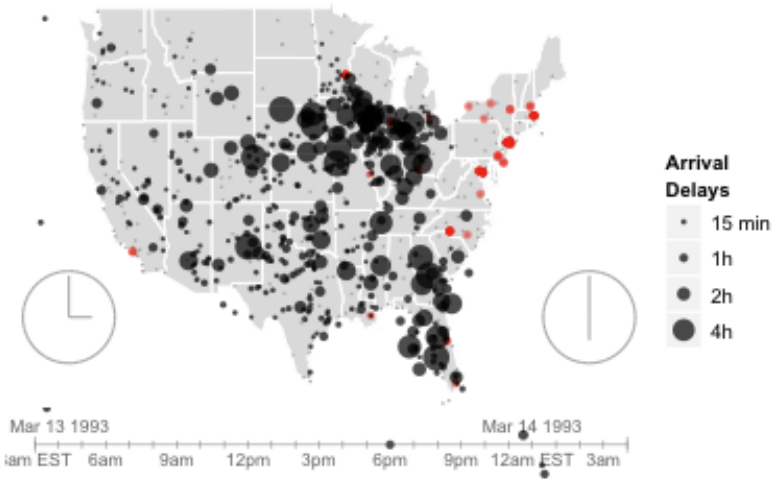
ggplot2 – Exemples



par Stefan Muthers

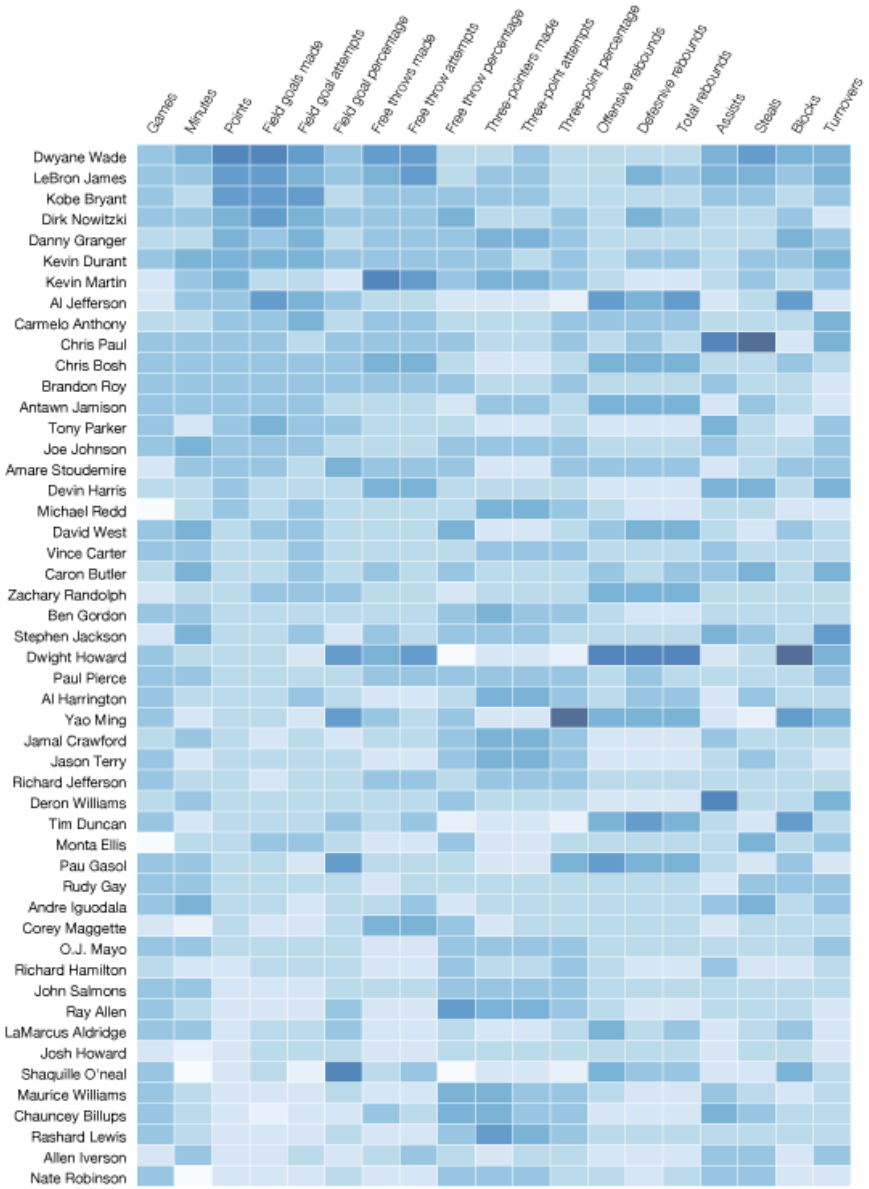
https ://github.com/hadley/ggplot2/wiki

ggplot2 – Exemples



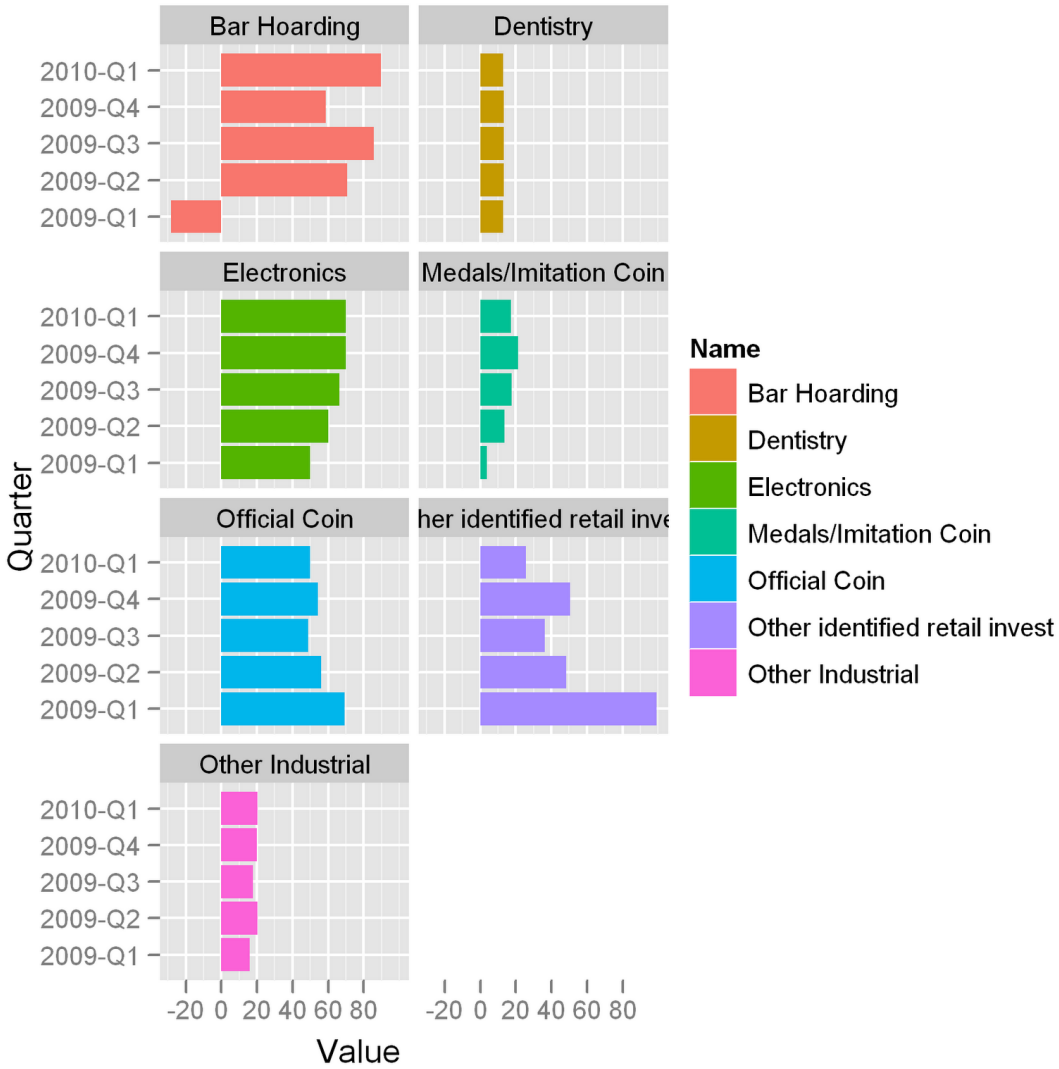
par Heike Hofmann (Iowa State University)
https ://github.com/hadley/ggplot2/wiki

ggplot2 – Exemples



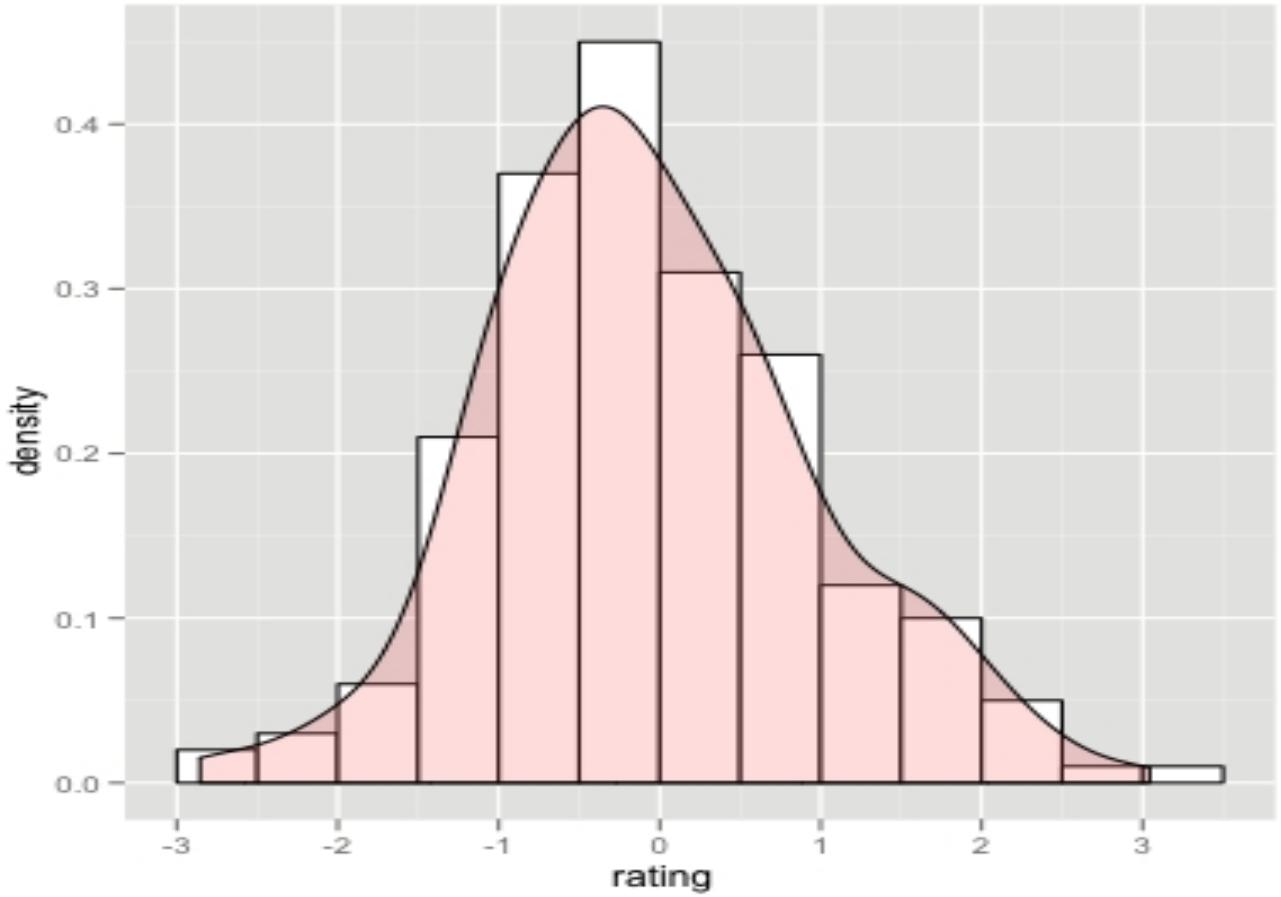
par learnr
http ://learnr.wordpress.com/

ggplot2 – Exemples



par C
http ://www.r-bloggers.com/analyze-gold-demand-and-investments-using-r/

ggplot2 – Exemples



par Winston Chang

http ://wiki.stdout.org/rcookbook

Outline

Références

- Graphiques de la base :
  - Murrell, P. 2005. – *R graphics*. Chapman & hall / CRC.
  - Page web de P. Murrell : http ://www.stat.auckland.ac.nz/ paul/RGraphics/rgraphics.html
  - http ://addictedtor.free.fr/graphiques/index.php
  - Graphic task view http ://cran.at.r-project.org/web/views/Graphics.html
  - Graphical model task view http ://cran.at.r-project.org/web/views/gR.html
- Graphiques de `ggplot2` :
  - Wilkinson, L. (2005) – *The grammar of graphics*. Statistics and Computing. Springer
  - Wickman, H. 2009 – *ggplot2. Elegant graphics for data analysis*. Springer.
  - Pageweb : http ://had.co.nz/ggplot2/
  - Page Git : https ://github.com/hadley/ggplot2
  - Groupe de discussion : http ://groups.google.com/group/ggplot2
  - Application web : http ://yeroon.net/ggplot2/