Présentation de ggplot2

Eric Matzner-Lober

Présentation de ggplot2

ggplot2 a été développé par Hadley Wickham comme une implémentation de [Grammar of Graphics] C'est un package relativement complet et puissant. Il peut faire énormément de chose, mais pas de 3D.

Les données

Nous allons travailler sur les données "Diamonds Data" livrées avec le package ggplot2

- Environ 54000 diamants de forme ronde
- Source : http://www.diamondse.info/
- 10 variables : carat, cut, color, clarity, depth, table, price, x, y, z
- Pour l'ensemble de l'aide et de la documentation : http://ggplot2.org/

```
library(ggplot2)
data(diamonds)
str(diamonds)
```

```
## 'data.frame':    53940 obs. of  10 variables:
##  $ carat  : num  0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...
##  $ cut    : Ord.factor w/ 5 levels "Fair"<"Good"<..: 5 4 2 4 2 3 3 3 1 3 ...
##  $ color  : Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<..: 2 2 2 6 7 7 6 5 2 5 ...
##  $ clarity: Ord.factor w/ 8 levels "I1"<"SI2"<"SI1"<..: 2 3 5 4 2 6 7 3 4 5 ...
##  $ depth  : num  61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
##  $ table  : num  55 61 65 58 58 57 57 55 61 61 ...
##  $ price  : int  326 326 327 334 335 336 336 337 337 338 ...
##  $ x      : num  3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
##  $ y      : num  3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
##  $ z      : num  2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...
```

La grammaire des graphes

- **Data** : Les variables à afficher
- **Aesthetics mapping** : Les dimensions selon lesquelles les données sont représentées
- **Geometries** : Formes utilisées pour représenter les données
- **Facets** : Tableau (lignes et colonnes) de graphes
- **Statistics** : Modèles ou transformations statistiques des données
- **Coordinates** : L'espace de représentation (horizontal, vertical, cartésien, polaire)
- **Scales** : L'échelle des axes (linéaire, logarithmique, à l'envers), les couleurs de remplissage
- **Thèmes** : Description de l'arrière plan

Qu'est ce qu'un graphique ?

- Un ensemble de couches

- Un ensemble d'échelles de dimensions (linéaire, logarithmique, ...)
- Un système de coordonnées (XY, polaire, ...)
- Une disposition (si plusieurs graphes)

Implémentation de cette grammaire dans ggplot2

- Data
- Aesthetic mapping : `aes()`
- Geometries : `geom`
- Statistics : `stat`

Une aide précieuse : Lorsque vous écrivez vos graphes, reportez vous à la Cheat Sheet ggplot2. En 2 pages, elle présente l'ensemble des fonctionnalités de bases que vous pouvez combiner !

https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf

Le principe en schéma

Geometries

Les `geoms` définissent les formes des éléments du graphes.

- Formes basiques : `geom_point()`, `geom_line()`, `geom_polygon()`, `geom_bar()`, `geom_text()`
- Formes composites : `geom_boxplot()`, `geom_pointrange()`
- Formes attachées à des objets statistiques : `geom_histogram()`, `geom_smooth()`, `geom_density()`

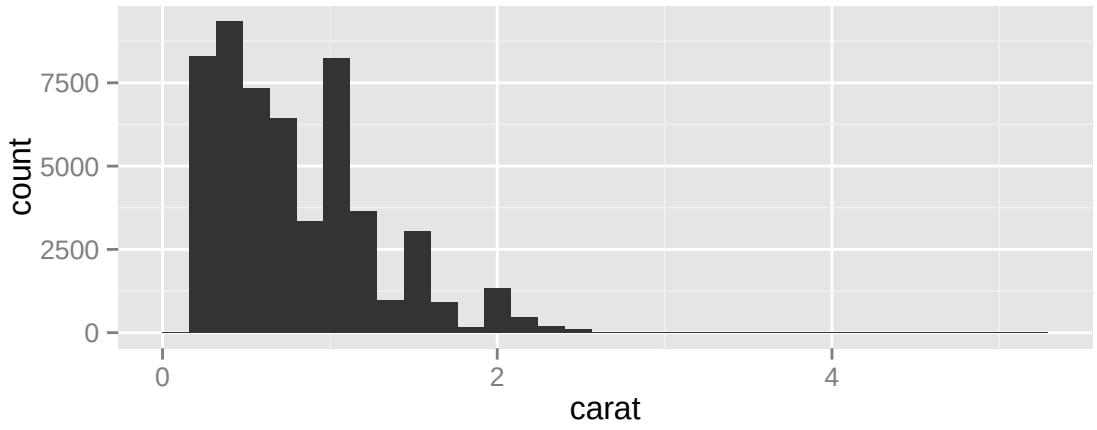
Statistics

Au lieu de représenter les données brutes, nous pouvons représenter une **transformation** des données : utilisation de la famille de fonction `stat_*()`.

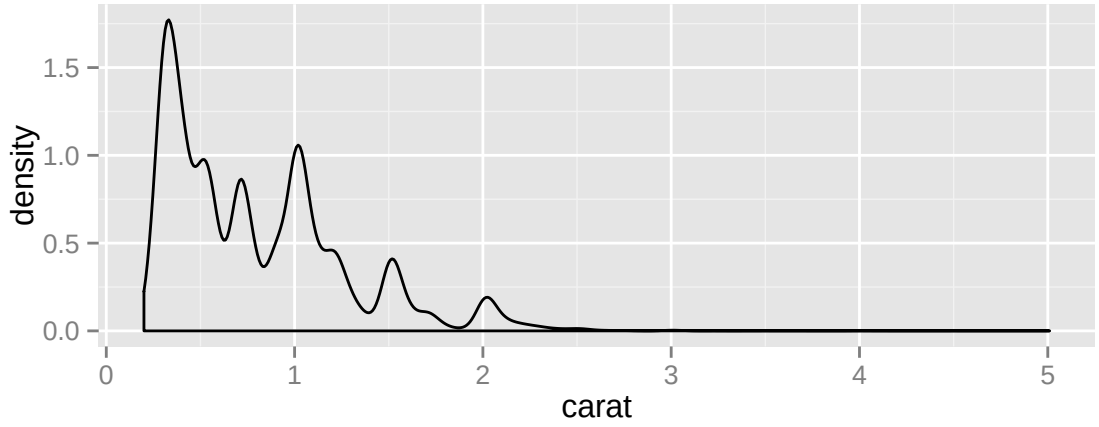
Représentation univariée

Variable quantitative : histogramme, densité, points

```
ggplot(diamonds, aes(carat)) + geom_histogram()
```

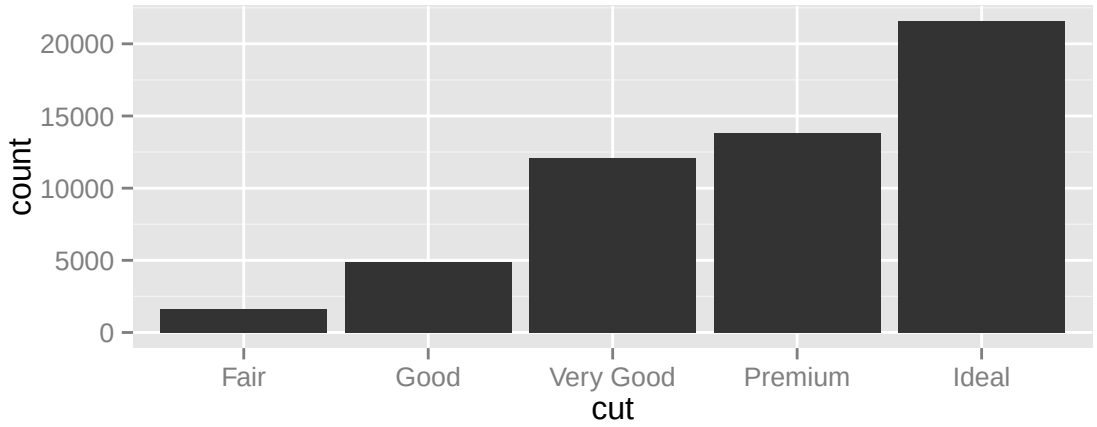



```
ggplot(diamonds, aes(carat)) + geom_density()
```



Variable quantitative

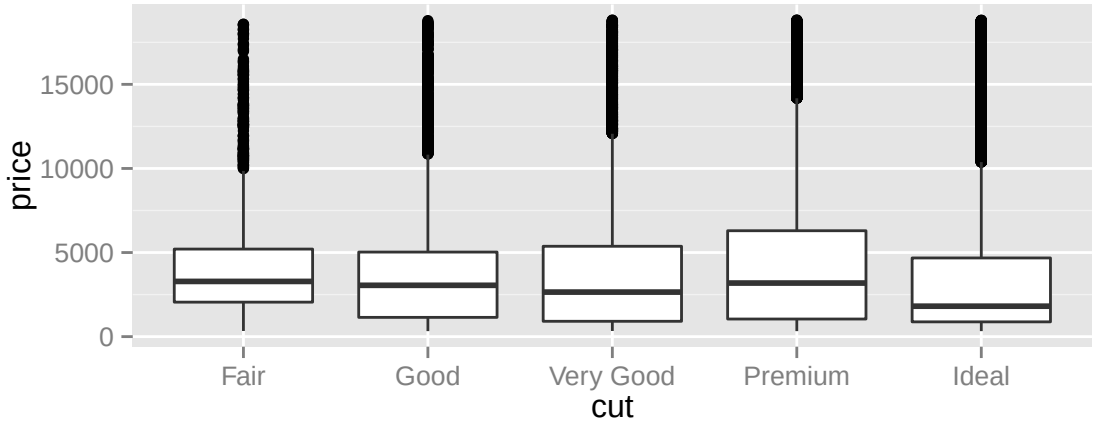
```
ggplot(diamonds, aes(cut)) + geom_bar()
```



Représentation bivariée

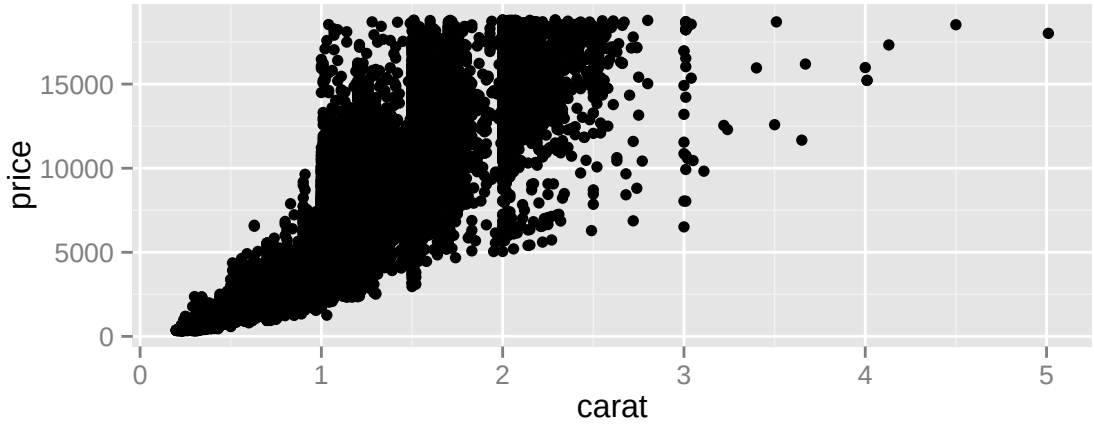
Quanti * quali

```
ggplot(diamonds, aes(x=cut, y=price)) + geom_boxplot()
```



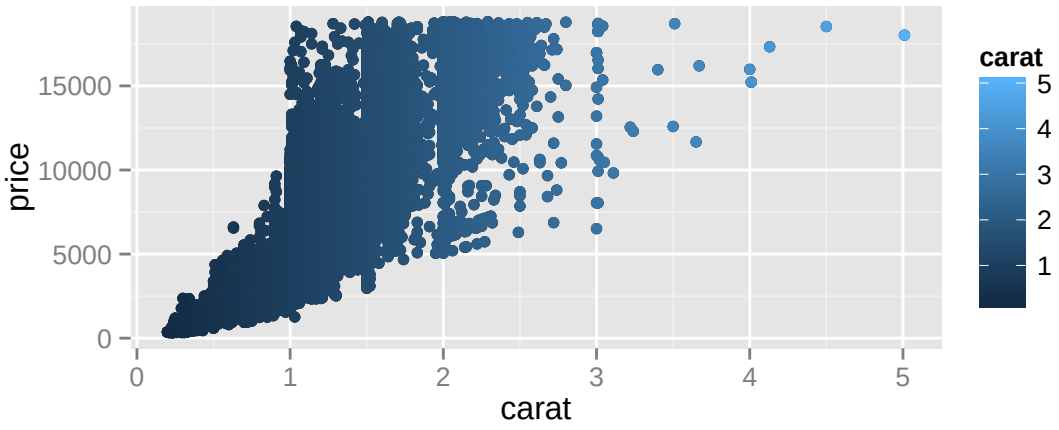
Quanti * quanti

```
ggplot(diamonds, aes(x=carat, y=price)) + geom_point()
```



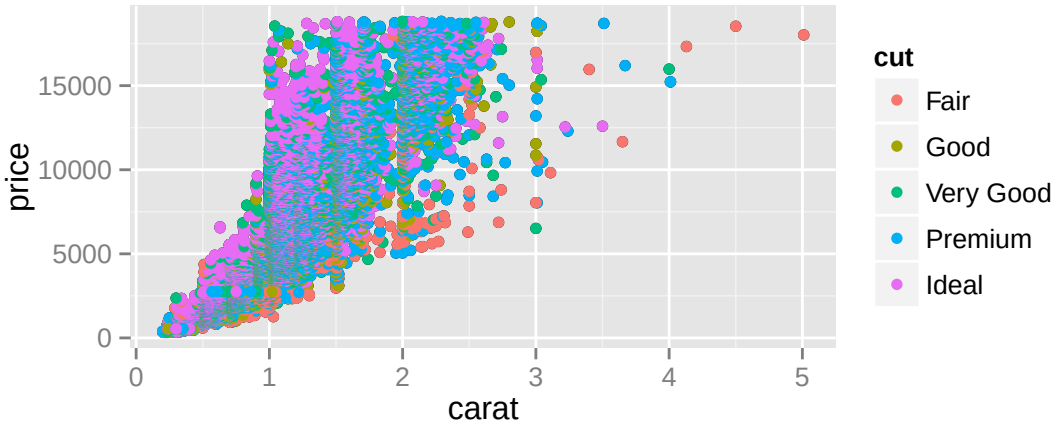
Exemple : Scatter plot avec couleurs venant d'une variable quanti

```
ggplot(diamonds, aes(x=carat, y=price)) + geom_point() + geom_point(aes(colour = carat))
```



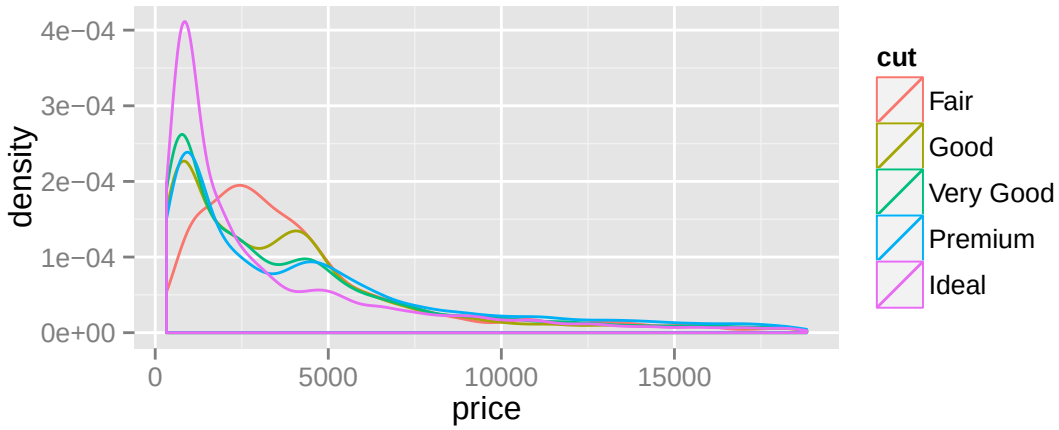
Exemple : Scatter plot avec couleurs venant d'une variable quali

```
ggplot(diamonds, aes(x=carat, y=price)) + geom_point() + geom_point(aes(colour = cut))
```

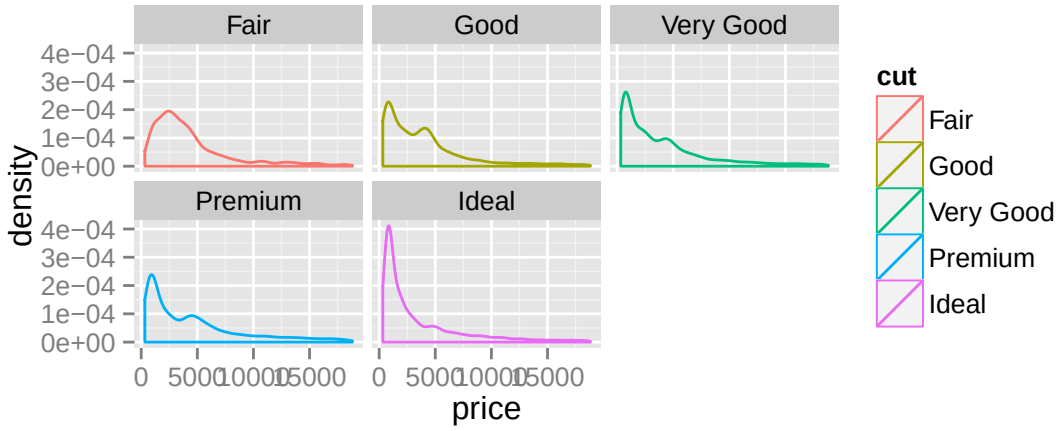


Exemple de grouping

```
ggplot(diamonds, aes(x = price, color = cut)) + geom_density()
```




```
ggplot(diamonds, aes(x = price, color = cut)) + geom_density() + facet_wrap(~ cut)
```



Que se passe t-il ?

```
ggplot(diamonds, aes(x=price)) + geom_histogram()
```

```
ggplot(diamonds, aes(x=price)) + stat_bin(geom="area")
```

```
ggplot(diamonds, aes(x=price)) + stat_bin(geom="point")
```

```
ggplot(diamonds, aes(x=price)) + geom_histogram(aes(fill = clarity))
```

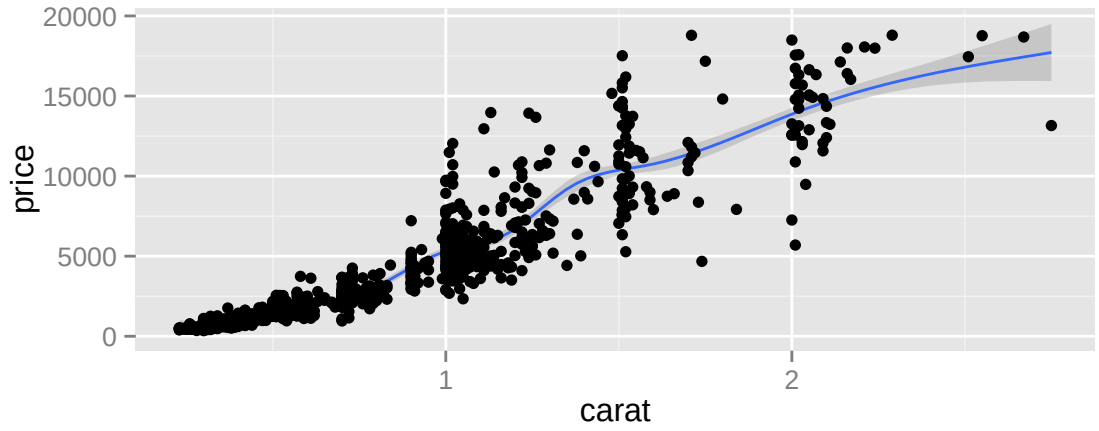
Les paramètres

Les paramètres modifient l'apparence des `geoms` et les opérations des `statistics` :

- `geom_smooth(method=lm)`
- `stat_bin(binwidth = 100)`
- `stat_summary(fun="mean_cl_boot")`
- `geom_boxplot(outlier.colour = "red")`
- `geom_point(colour = "red", size = 5)`
- `geom_line(linetype = 3)`

Pour représenter un sous-ensemble de points ainsi qu'une courbe de lissage :

```
ggplot(diamonds[sample(nrow(diamonds), size = 1000),],aes(x = carat, y = price)) +
  stat_smooth() + geom_point()
```

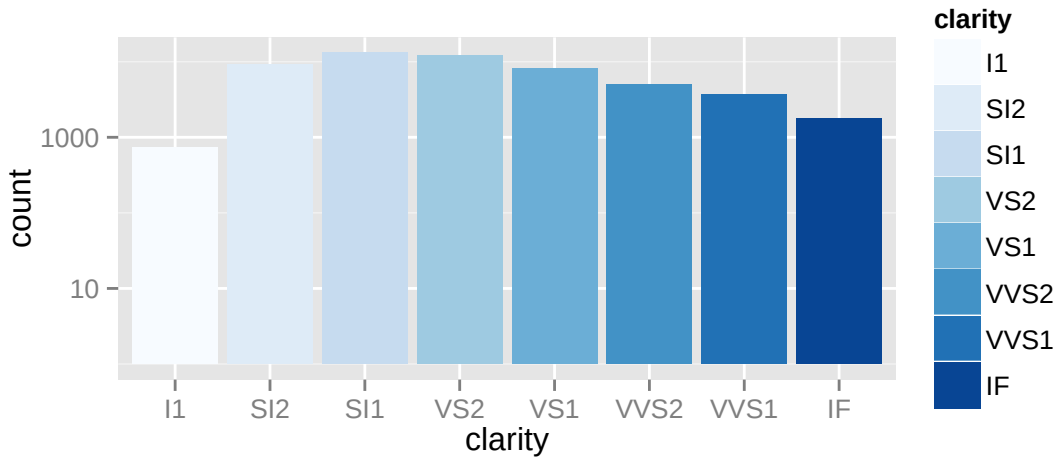


Les échelles (scales)

Les `scales` permettent de changer la manière dont est représentée une dimension. On peut par exemple :

- Représenter un axes en échelle logarithmique
- Changer les limites des axes
- Changer les `breaks` qui sont représentés sur l'axe
- Changer les labels
- Changer les couleurs

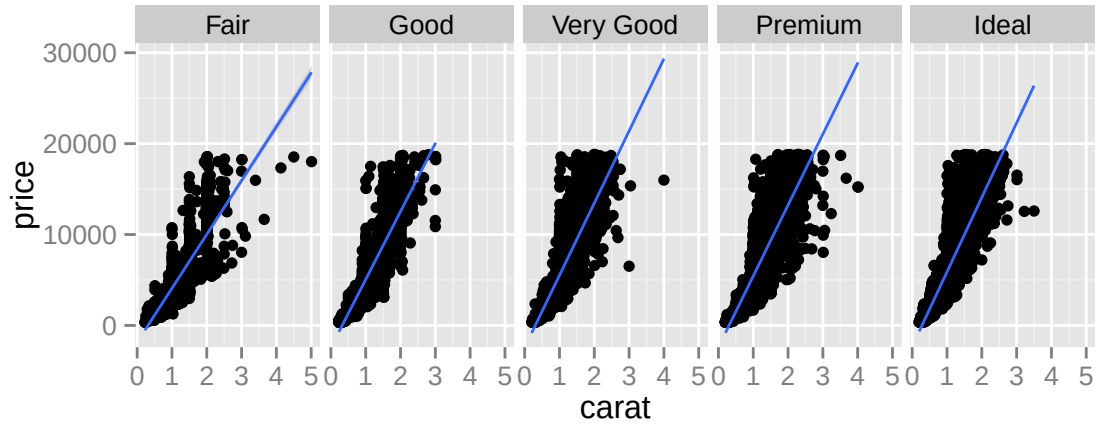
```
b <- ggplot(diamonds, aes(x=clarity))
b + geom_bar(aes(fill = clarity)) + scale_fill_brewer() + scale_y_log10()
```



Facetting

Lorsque l'on dispose de données multivariées, il est utile d'afficher dans une même fenêtre graphique plusieurs graphes correpondants aux différents sous-ensembles.

```
p <- ggplot(diamonds, aes(x = carat, y = price)) + geom_point()
# With one variable
p + facet_grid(. ~ cut) + geom_smooth(method=lm)
```



Coordinates

Le système de coordonnées permet de contrôler comment les deux dimensions comprises dans `aes(x = , y = )` sont reliées. Par défaut, c'est le système de coordonées cartésiennes (X, Y)

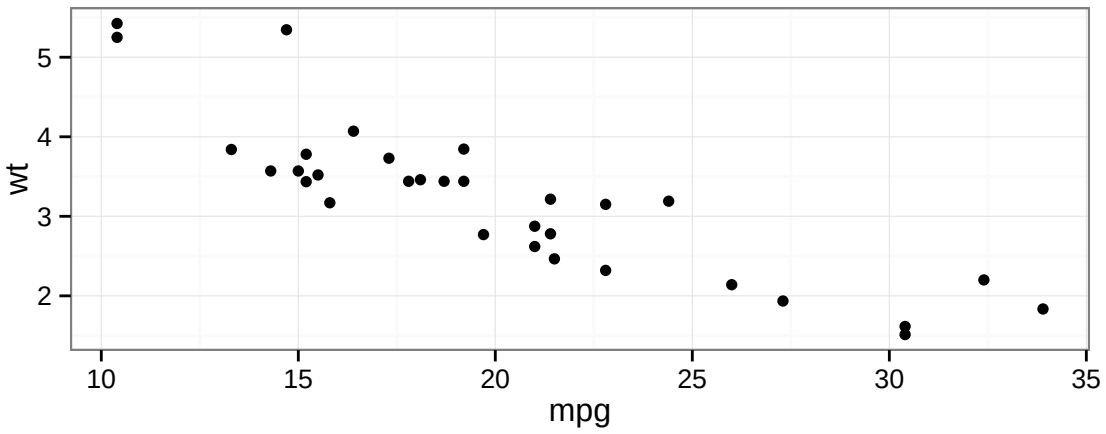
Les autres types de coordonées :

- `coord_flip()`
- `coord_map()`
- `coord_polar()`

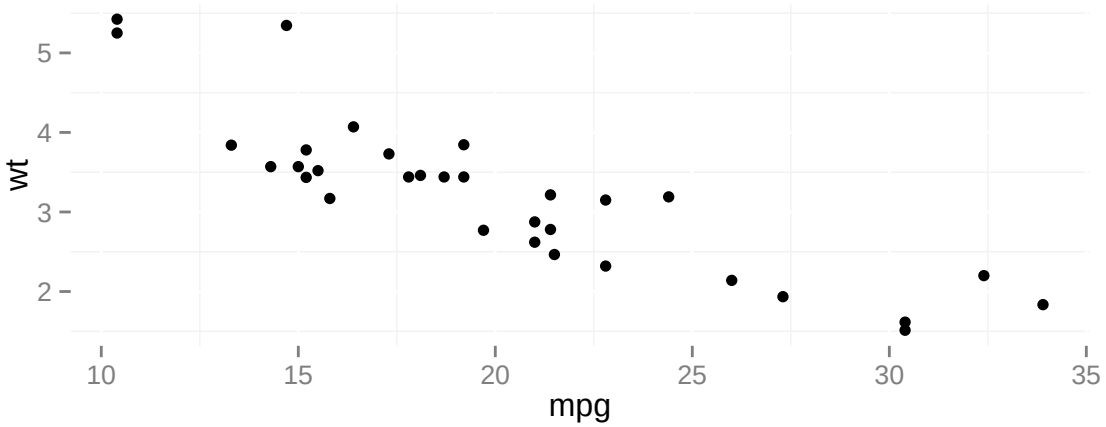
Themes

Les thèmes définissent l'arrière-plan.

```
ggplot(mtcars, aes(x = mpg, y = wt)) + geom_point() + theme_bw()
```




```
ggplot(mtcars, aes(x = mpg, y = wt)) + geom_point() + theme(panel.background = element_blank())
```



Le package ggthemes permet de choisir différents background.

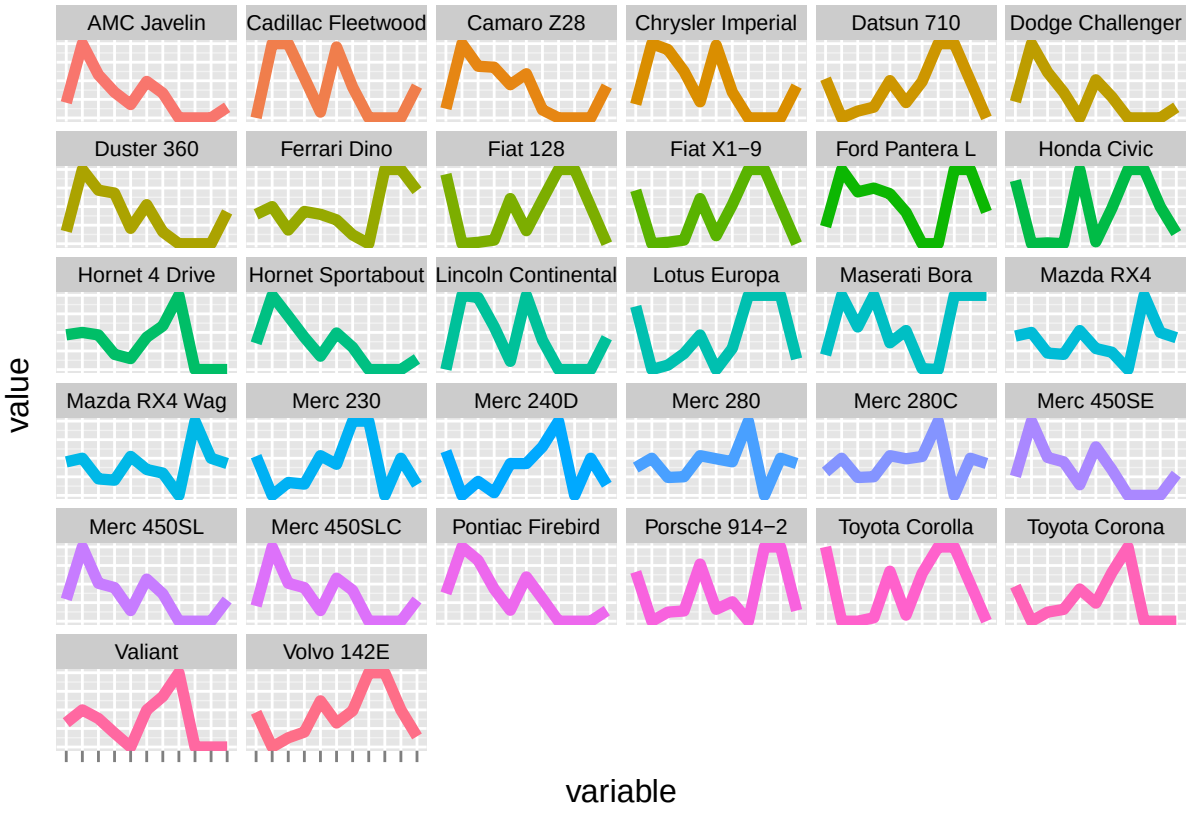
```
require(ggthemes)
```

Parallel Coordinates and Radar Plots

```
# rescale all variables to lie between 0 and 1
scaled <- as.data.frame(lapply(mtcars, ggplot2:::rescale01))
scaled$model <- rownames(mtcars)    # add model names as a variable
mtcarsm <- reshape2::melt(scaled)
```

```
## Using model as id variables
```

```
ggplot(mtcarsm, aes(x = variable, y = value)) +
  geom_line(aes(group = model, color = model), fill = NA, size = 2) +
  theme(strip.text.x = element_text(size = rel(0.8)),
        axis.text.x = element_blank(),
        axis.ticks.y = element_blank(),
        axis.text.y = element_blank()) +
  guides(color = "none") + facet_wrap(~ model)
```




```
coord_radar <- function(...) {
  structure(coord_polar(...), class = c("radar", "polar", "coord"))
}
is.linear.radar <- function(coord) TRUE
```