

# Graphics in R

Christophe Lalanne

Fall 2017

Les différents systèmes graphiques

Le package ggplot2

Paramètres avancés

Graphiques interactifs

*The greatest value of a picture is when it forces us to notice  
what we never expected to see.*

*–John Tukey*

## Les différents systèmes graphiques

# Systèmes graphiques

R dispose de deux principaux système graphiques, base et grid (P. Murrell 2005), et de trois interfaces/packages : graphics (Becker, Chambers, and Wilks 1988), lattice (Sarkar 2008) et ggplot2 (Wickham 2009) ; voir aussi Chang (2013).

# Graphiques de base

```
1 > plot(density(ToothGrowth$len[ToothGrowth$supp == "O"]),
2       main = "", xlab = "len", las = 1, lwd = 2, col = "coral")
3 > lines(density(ToothGrowth$len[ToothGrowth$supp == "VC"]),
4       lwd = 2, col = "cornflowerblue")
5 > ## rug() does not allow to use a grouping factor
6 > points(x = ToothGrowth$len[ToothGrowth$supp == "O"],
7        y = runif(n = length(ToothGrowth$len[ToothGrowth$supp == "O"]),
8              min = -0.001, max = 0.001),
9        col = "coral")
10 > points(x = ToothGrowth$len[ToothGrowth$supp == "VC"],
11        y = runif(n = length(ToothGrowth$len[ToothGrowth$supp == "VC"]),
12              min = -0.001, max = 0.001),
13        col = "cornflowerblue")
14 > legend("top", levels(ToothGrowth$supp),
15        col = c("coral", "cornflowerblue"),
16        lty = 1, bty = "n")
```

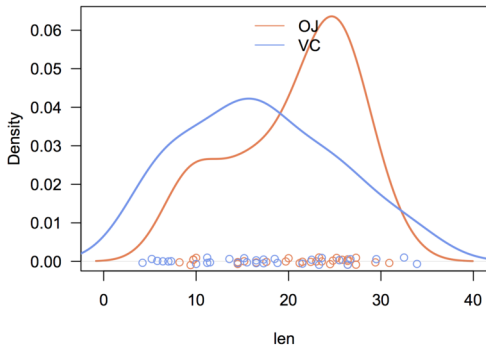


Figure 1: Graphique de type base

# Graphiques lattice

```
library(lattice)  
densityplot(~ len, data = ToothGrowth, group = supp,  
            auto.key = TRUE)
```

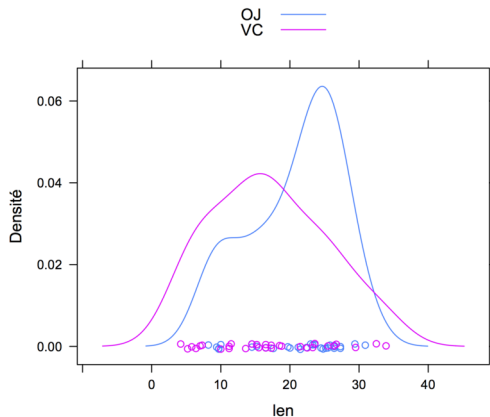


Figure 2: Graphique de type lattice



# Graphiques ggplot

```
library(ggplot2)
ggplot(data = ToothGrowth, aes(x = len, color = supp)) +
  geom_line(stat = "density") + geom_rug()
```

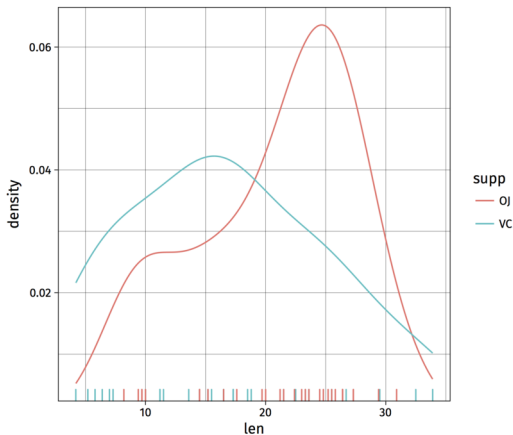


Figure 3: Graphique de type ggplot

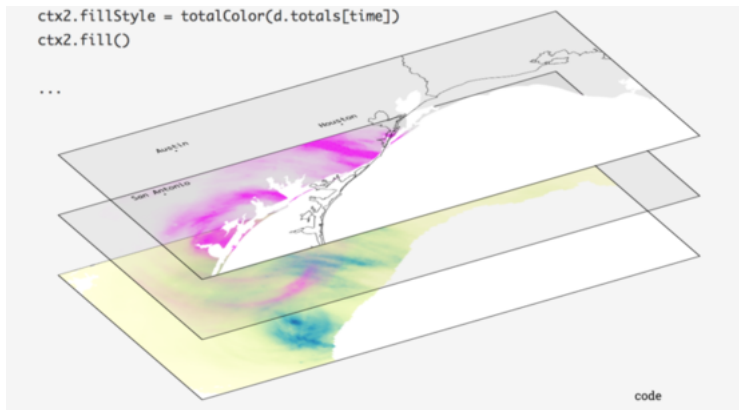
## Le package ggplot2

# “The Grammar of Graphics”



- ▶ Wilkinson (2005) fournit un cadre de réflexion et des idées d'application d'une grammaire des graphiques
- ▶ Wickham (2009) offre une implémentation en langage R : <https://github.com/hadley/ggplot2-book>

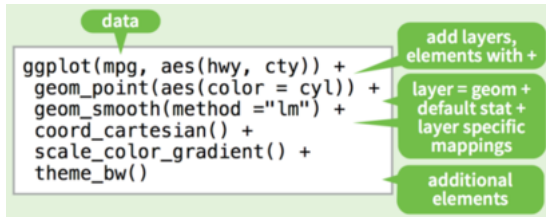
# Un concept de couches



<https://roadtolarissa.com/hurricane/>

# Les bases d'un graphique ggplot

- ▶ `ggplot()` : un data frame (`data =`) et un mapping (`aes()`)
- ▶ `geom_*()` : un ou plusieurs objets géométriques
- ▶ `facet_wrap()` : un système de facettes (conditionnement)
- ▶ `scale_*_*` : une échelle pour les axes ou les palettes de couleurs
- ▶ `coord_*()` : un système de coordonnées
- ▶ `labs()` : des annotations pour les axes et le graphique
- ▶ `theme_*()` : un thème personnalisé



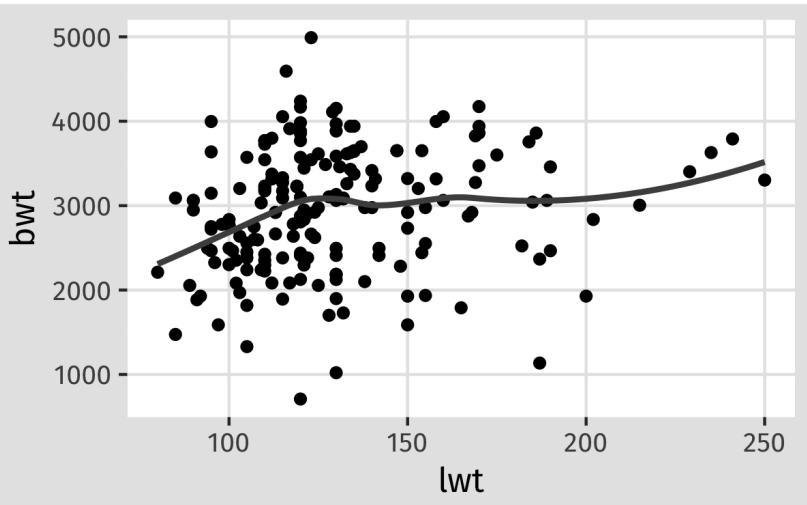
[ggplot2-cheatsheet.pdf](#)<sup>1</sup>

---

<sup>1</sup>version plus récente disponible sur le site de RStudio.

# Mise en œuvre

```
p = ggplot() +  
  layer(data = MASS::birthwt,  
        stat = "identity",  
        geom = "point",  
        mapping = aes(x = lwt, y = bwt),  
        position = "identity") +  
  layer(data = MASS::birthwt,  
        stat = "smooth",  
        geom = "line",  
        mapping = aes(x = lwt, y = bwt),  
        position = "identity",  
        params = list(method = "auto"))
```





# Syntaxe ggplot

Formulation équivalente et simplifiée :

```
library(MASS)
p = ggplot(data = birthwt, aes(x = lwt, y = bwt))
p + geom_point() + geom_smooth(method = "auto")
```

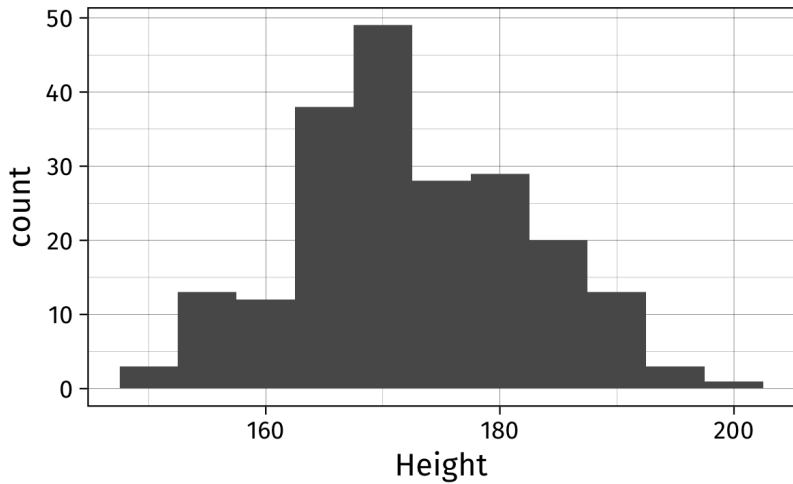
→ structure graphique (ggplot) et objets géométriques (geom\_\*).

# Histogramme d'effectifs

MASS::survey = “responses of 237 Statistics I students at the University of Adelaide to a number of questions.” (Venables and Ripley 2002)

12 variables : Sex Wr.Hnd NW.Hnd W.Hnd Fold Pulse Clap Exer  
Smoke Height M.I Age

```
p = ggplot(data = survey, aes(x = Height))  
p + geom_histogram(binwidth = 5) ## bins = 11
```

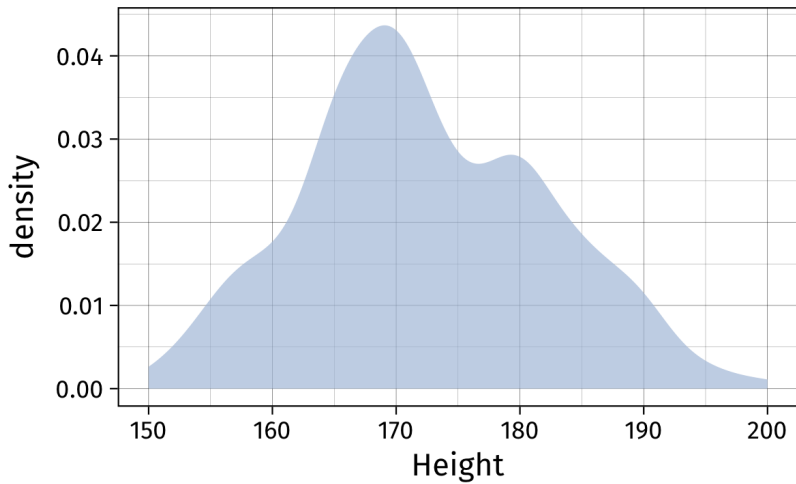


# Courbe de densité

```
p = ggplot(data = survey, aes(x = Height))  
p + geom_density(adjust = 0.8)      ## (1)
```

On peut également construire une courbe de densité explicitement à l'aide de `geom_line()`

```
p + geom_line(stat = "density", ...) ## (1)
```



# Estimateur

Venables and Ripley (2002), §5.6 – *Density Estimation* (pp. 126–130)

$$\hat{f}(x) = \frac{1}{nb} \sum_{j=1}^n K\left(\frac{x - x_j}{b}\right)$$

$x_1, \dots, x_n$  un échantillon de taille  $n$

$K()$  une fonction noyau fixée, par défaut gaussienne

$\hat{b} = 1.06 \min(\hat{\sigma}, \text{IQR}/1.34) n^{-1/5}$  la largeur de la fenêtre de lissage.

# Histogramme d'effectifs revisité

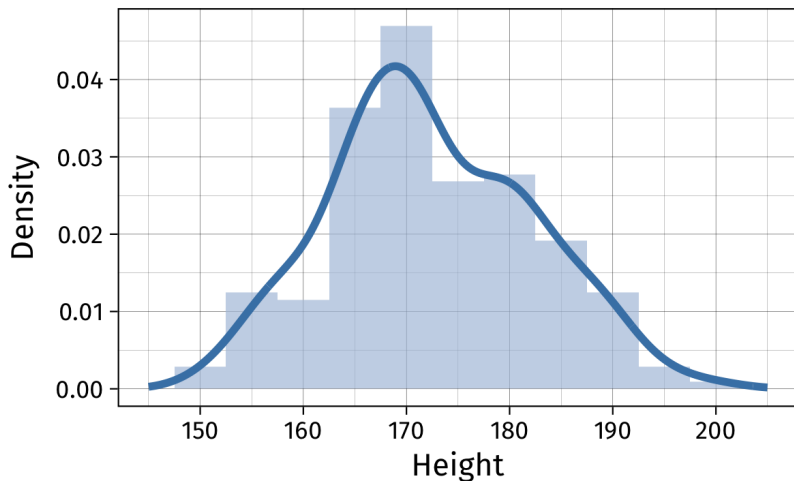


Figure 4: Distribution de la taille des répondants

# Boîte à moustaches (“boxplot”)

Un diagramme de type boîte à moustaches (Tukey 1977) fournit une représentation graphique (ou schématique) du résumé numérique renvoyé par `summary()`.

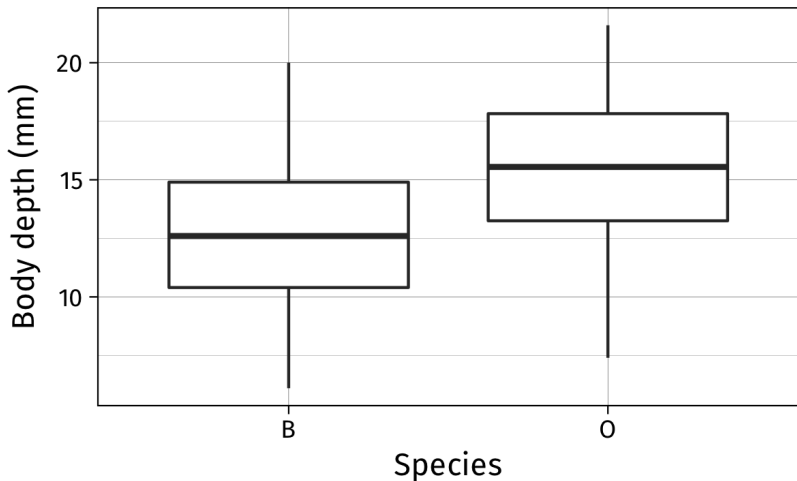
MASS: : crabs = “morphological measurements on 50 crabs each of two colour forms and both sexes, of the species *Leptograpsus variegatus* collected at Fremantle, W. Australia.”

8 variables : sp sex index FL RW CL CW BD

```
> summary(crabs[["BD"]])
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
6.10	11.40	13.90	14.03	16.60	21.60





```
p = ggplot(data = crabs, aes(x = sp, y = BD))  
p + geom_boxplot()
```

# Changement de format (large → long)

Le package `reshape2` fournit deux commandes qui permettent d'alterner entre le format large (`dcast`) et long (`melt`). Cela permet, entre autres, de travailler avec des séries de mesure multivariées.

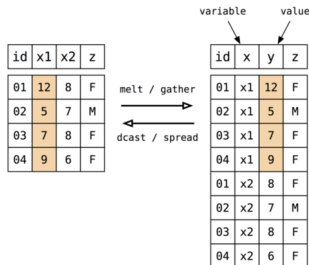
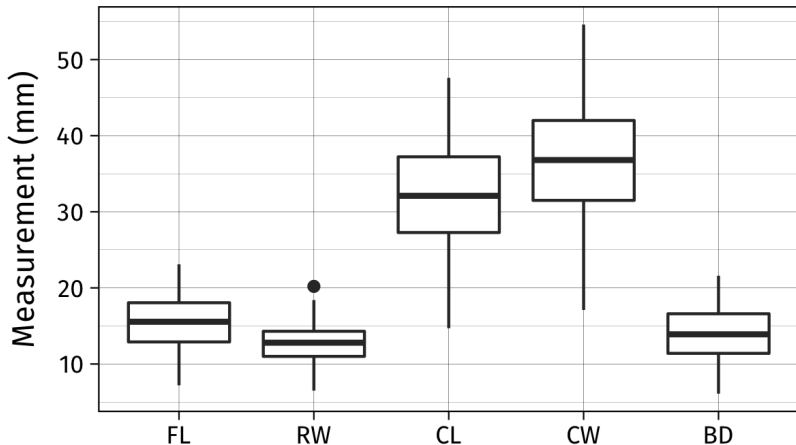


Figure 5: Représentation au format wide/long

# Illustration

```
crabs.df = reshape2::melt(crabs, measure.vars = 4:8)
p = ggplot(data = crabs.df, aes(x=variable, y=value))
p = p + geom_boxplot()
```

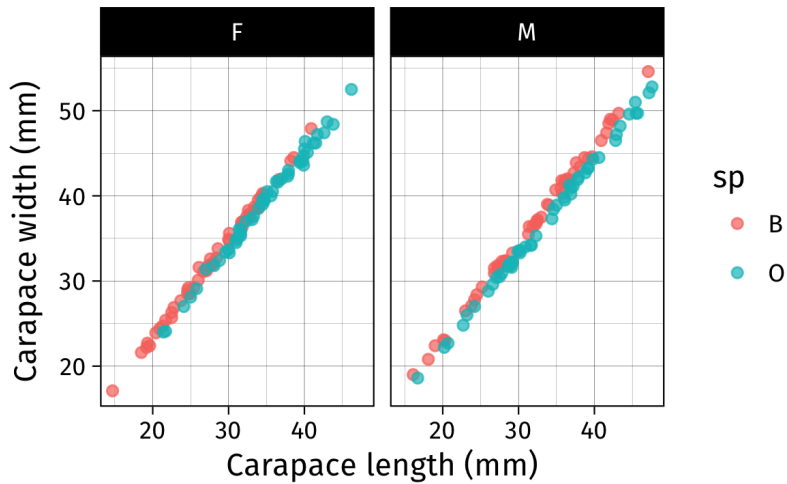


# Utilisation de facettes

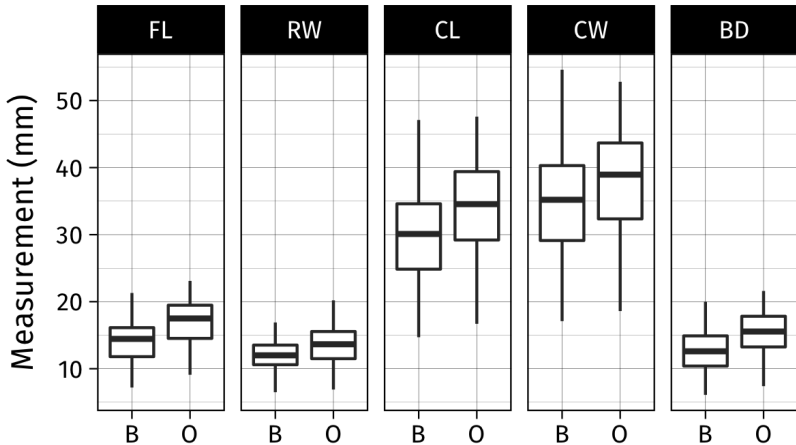
Les “facettes” permettent de prendre en considération une ou plusieurs variables de conditionnement, e.g. une relation de type  $Y \sim X \mid Z1$  ou  $Y \sim X \mid Z1 + Z2$  (Becker, Cleveland, and Shyu 1996).

Ici, Z1 et Z2 peuvent être de type numérique ou discret (avec niveaux ordonnés ou non). Le conditionnement peut être rendu apparent *via* des attributs propres aux objets géométriques (couleur, forme, etc.) et/ou *via* des facettes.

```
p = ggplot(data = crabs,  
           aes(x = CL, y = CW, color = sp))  
p = p + geom_point(alpha = .7)  
p + facet_wrap(~ sex)
```



```
p = ggplot(data = crabs.df, aes(x = sp, y = value))  
p = p + geom_boxplot()  
p + facet_grid(~ variable)
```



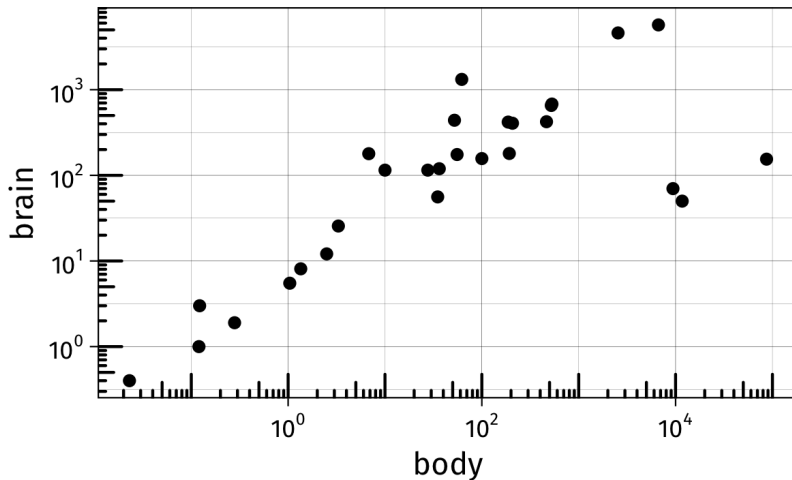
## Paramètres avancés

# Echelles et repères

- ▶ **type**: `scale_x_continuous(name, breaks, labels, limits =, trans =)`
- ▶ **limites**: `xlim(), ylim(), expand_limits(), limits =`
- ▶ **transformation**: `trans =, scale_y_log10(), coord_trans(y = "log10")`
- ▶ **format {scales}**: `scale_x_date(labels = date_format("%m/%d")), annotation_logticks()`
- ▶ **propriétés**: `coord_flip(), coord_equal(), coord_polar()`



# Illustration



# Changement de repère

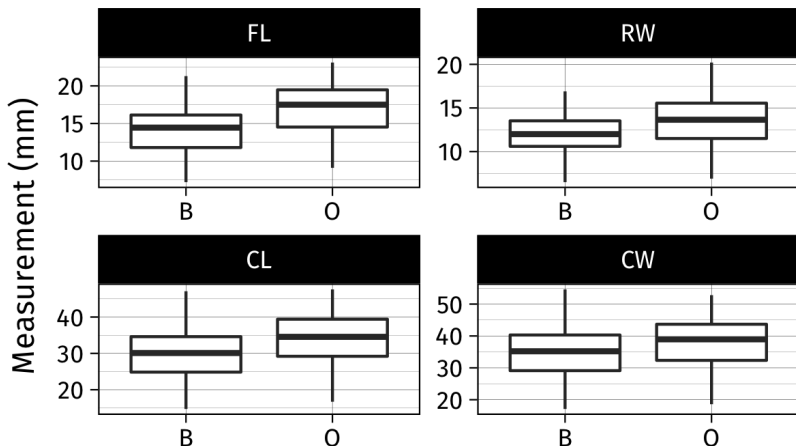
Sans toucher au mapping (`aes()`) définissant le rôle joué par les variables, il est possible d'échanger les axes du repère cartésien :

```
p = ggplot(data = crabs.df,  
           aes(x = variable, y = value))  
p + geom_boxplot()
```

```
p = ggplot(data = crabs.df,  
           aes(x = variable, y = value))  
p = p + geom_boxplot()  
p + coord_flip()
```

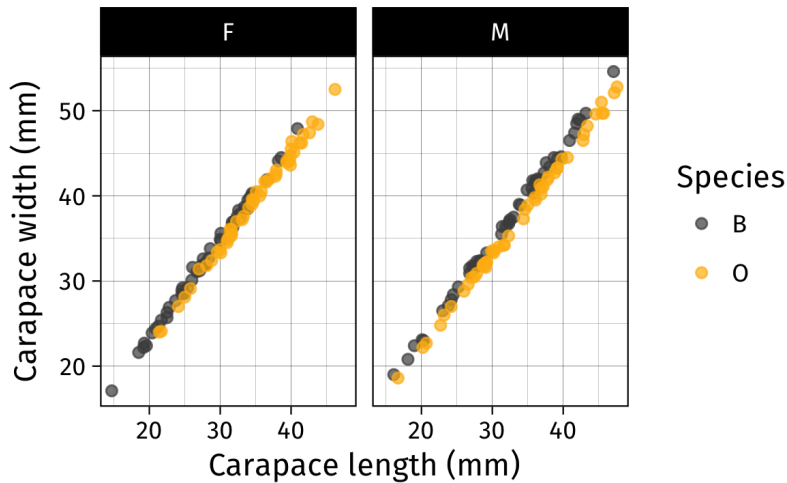
## Cas des facettes “libres”

```
p = ggplot(data = subset(crabs.df, variable != "BD"),  
           aes(x = sp, y = value))  
p = p + geom_boxplot()  
p + facet_wrap(~ variable, nrow = 2, scales = "free")
```



## Palettes “divergente” de couleurs

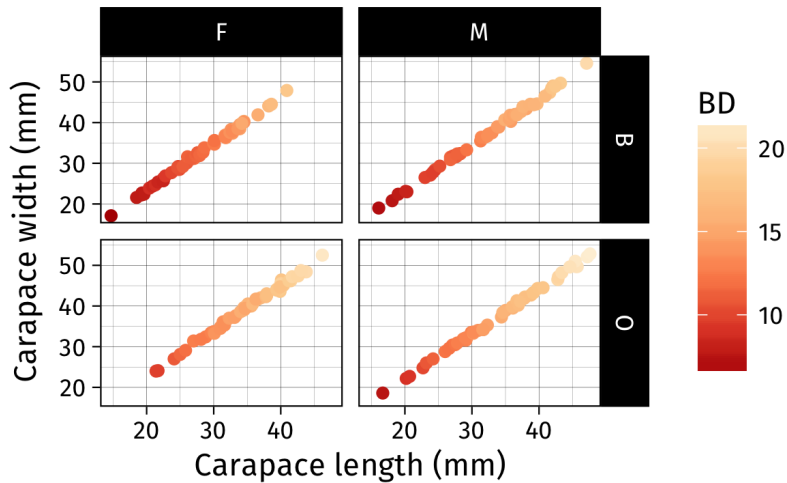
```
p = ggplot(data = crabs,  
           aes(x = CL, y = CW, color = sp))  
p = p + geom_point(alpha = .7)  
p = p + scale_color_manual("Species",  
                           values = c(grey(.3),  
                                       "darkgoldenrod1")) ## (1)  
p = p + facet_wrap(~ sex)  
  
p + scale_color_brewer(palette = "Pastel1") ## (1)
```



# Palette séquentielle (cas continu)

- ▶ <http://colorbrewer2.org>
- ▶ <http://hclwizard.org>
- ▶ pour une discussion, voir Zeileis, Hornik, and Murrell (2009)

```
p = ggplot(data = crabs,  
           aes(x = CL, y = CW, color = BD))  
p = p + geom_point()  
p = p + scale_color_distiller(palette = "OrRd") ## (1)  
p = p + facet_grid(sp ~ sex)  
  
p <- p + scale_color_gradient() ## (1)
```



# Thèmes graphiques

- ▶ 8 thèmes de base, `ggplot2::theme_*`
- ▶ `ggthemes`, `ggthemr`, `hrbrthemes`

À partir d'un thème de base, il est toujours possible de redéfinir soi-même certains éléments du thème.

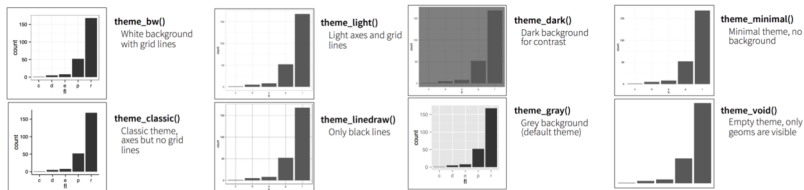


Figure 6: Thèmes de base ggplot. Adapté de Wickham and Grolemund (2017),  
§3 – Data visualisation



## Graphiques interactifs

# Le package ggvis

<http://ggvis.rstudio.com>

- ▶ basé sur Vega (projets connexes @ UW Interactive Data Lab, e.g. Wongsuphasawat et al. (2016))
- ▶ intégration à RStudio (“Viewer”)
- ▶ fonctionnalités plus limitées que ggplot2, mais dans l’esprit de l’approche Shiny

# Similarité avec ggplot

chaînage des couches (layer) avec + versus {dplyr} %>%:

```
ggplot(data = birthwt, aes(x = lwt, y = bwt)) +  
  geom_point() +  
  geom_smooth(method = "auto")
```

```
ggvis(data = birthwt, ~ lwt, ~ bwt) %>%  
  layer_points() %>%  
  layer_smooths()
```

# Interactivité

```
ggvis(data = birthwt, ~ lwt, ~ bwt) %>%  
  layer_points(fill := "cornflowerblue") %>%  
  layer_smooths(span = input_slider(0.5, 1.5,  
                                     value = 1))
```

```
ggvis(data = survey, x = ~ Height) %>%  
  layer_densities(adjust = input_slider(0.1, 2,  
                                         value = 1,  
                                         step = 0.1))
```

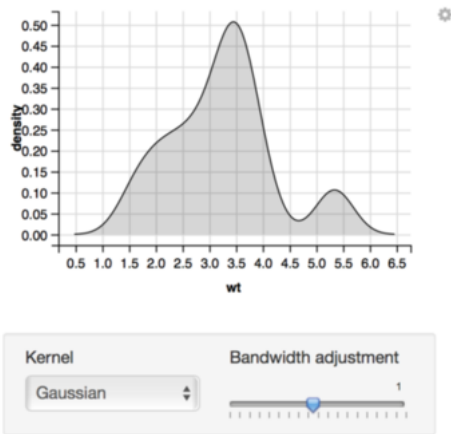


Figure 7: Exemple de graphique interactif avec ggvis

# plotly

Basé sur [plotly.js](https://plot.ly), le package `plotly` permet de construire des graphiques interactifs ou d'embarquer directement des graphiques `ggplot`, publiables sur <https://plot.ly>.

```
library(plotly)
p = ggplot(data = birthwt, aes(x = lwt, y = bwt)) +
  geom_point() +
  geom_smooth(method = "auto")
ggplotly(p)
```

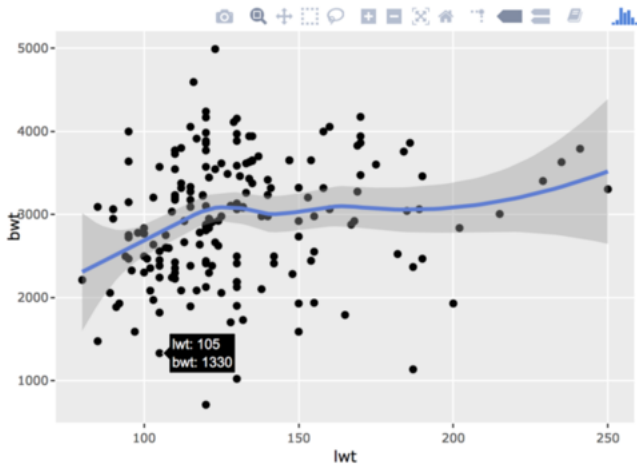


Figure 8: Exemple de graphique interactif avec ggplotly

# References I

Becker, R. A., J. M. Chambers, and A. R. Wilks. 1988. *The New S Language*. Chapman & Hall.

Becker, R. A., W. S. Cleveland, and M. J. Shyu. 1996. "The Visual Design and Control of Trellis Display." *Journal of Computational and Statistical Graphics* 5 (2):123–55.

Chang, W. 2013. *R Graphics Cookbook*. O'Reilly Media.

Murrell, Paul. 2005. *R Graphics*. Chapman & Hall/CRC.

Sarkar, Deepayan. 2008. *Lattice, Multivariate Data Visualization with R*. Springer.

Tukey, J. W. 1977. *Exploratory Data Analysis*. Addison-Wesley.

Venables, W. N., and B. D. Ripley. 2002. *Modern Applied Statistics with S*. 4th ed. Springer.



## References II

Wickham, Hadley. 2009. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. <http://ggplot2.org>.

Wickham, Hadley, and Garrett Grolemund. 2017. *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. O'Reilly Media, Inc. <http://r4ds.had.co.nz>.

Wilkinson, Leland. 2005. *The Grammar of Graphics*. Springer.

Wongsuphasawat, K., D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. 2016. "Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations." *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*. <http://idl.cs.washington.edu/papers/voyager>.

Zeileis, A., K. Hornik, and P. Murrell. 2009. "Escaping RGBland: Selecting Colors for Statistical Graphics." *Computational Statistics & Data Analysis* 53 (9):3259–70. <https://doi.org/10.1016/j.csda.2008.11.033>.