

# Package ggplot2 – Initiation à l'utilisation du package

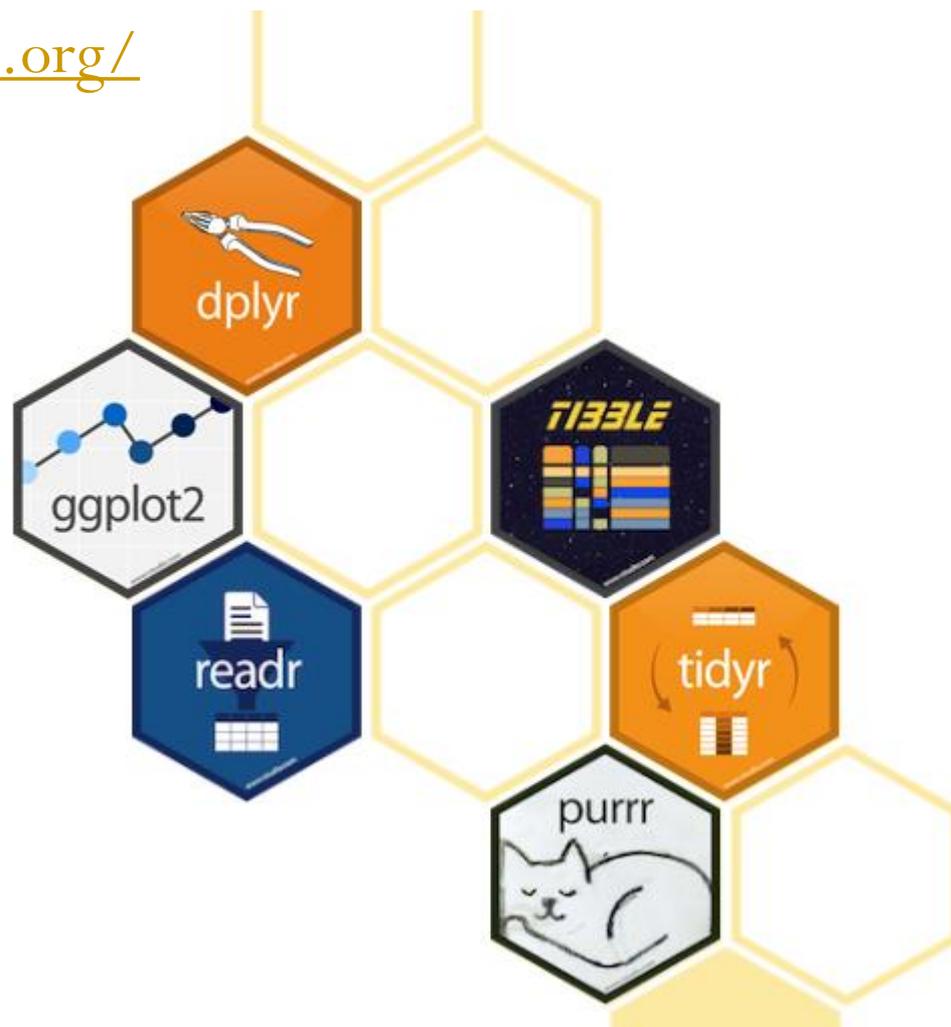
# Plusieurs conseils pour comprendre le fonctionnement du package ggplot2

- Comprendre la logique de construction d'un graphique
- Savoir manipuler les différentes couches
- Trucs et astuces pour personnaliser vos graphiques

# Prise en main du package ggplot2 - introduction

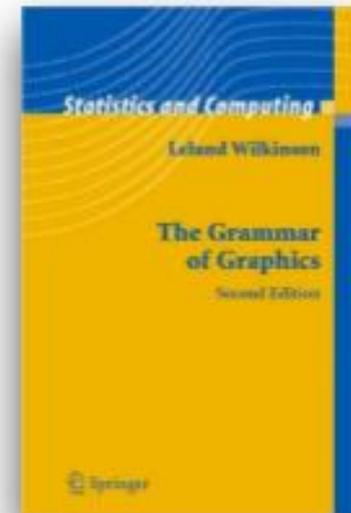
# Ggplot2 – un package du tidyverse

<https://www.tidyverse.org/>



# Grammar of graphics

- « *The Grammar of Graphics* » de Leland Wilkinson a eu une influence importante sur la façon de penser les graphes
- 2 principes
  - Graphique = différentes couches d'éléments grammaticaux
  - Des graphiques signifiants respectant des règles mathématiques et esthétiques



# *The quick brown fox jumps over the lazy dog*

<i>Article</i>	<i>The</i>	<i>A</i>	<i>The</i>
<i>Adjective</i>	<i>quick brown</i>	<i>rabid red</i>	
<i>Noun</i>	<i>fox</i>	<i>fox</i>	<i>Hunter</i>
<i>Verb</i>	<i>jumps</i>	<i>bit</i>	<i>shot</i>
<i>Preposition</i>	<i>over</i>		
<i>Article</i>	<i>the</i>	<i>the</i>	<i>the</i>
<i>Adjective</i>	<i>lazy</i>	<i>friendly</i>	<i>rabid red</i>
<i>Noun</i>	<i>dog.</i>	<i>dog.</i>	<i>fox.</i>

# ggplot2

- Créé par Hadley Wickham, ce package offre un langage graphique puissant pour créer des graphiques élégants et complexe
- Basé sur « The Grammar of Graphics » (Leland Wilkinson)



# ggplot2

- Le package `ggplot2` permet la réalisation de graphiques directement à partir d'un fichier détail, d'un fichier détail pondéré, ou de données agrégées
  - On initialise un graphique
  - On y ajoute des calques



# 1 – Comment initialiser correctement son graphique

# Éléments essentiels de grammaire

Élément	Description
Data	Les données à afficher
Aesthetics	Les dimensions selon lesquelles les données seront représentées
Geometries	Formes utilisées pour représenter les données

# Principe de composition d'un graphique

- Deux fonctions permettent d'initialiser un graphique :
  - `qplot()`
  - `ggplot()`
- Chaque élément s'ajoute à un autre par le symbole : `+`

# Principe de construction d'un graphique

- La construction d'un graphique fonctionne sur le principe d'un ajout de couches successives
- 1 – avec la fonction `ggplot()`, on déclare :
  - Le dataset sur lequel on va travailler
  - Les variables à représenter (argument `mapping` et fonction `aes()` qui peuvent être déclarés ailleurs)
- 2 – on déclare la géométrie, c'est-à-dire le type de graphique souhaité, avec les fonctions préfixés par `geom_XXX()`

# Construire un graphique avec ggplot2 : un exemple

```
# Chargement du package ggplot2  
library(ggplot2)
```

```
# Affichage des 5 premières lignes de la table  
head(mtcars, n = 5)
```

```
> str(mtcars)  
'data.frame': 32 obs. of 11 variables:  
 $ mpg : num 21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...  
 $ cyl : num 6 6 4 6 8 6 8 4 4 6 ...  
 $ disp: num 160 160 108 258 360 ...  
 $ hp : num 110 110 93 110 175 105 245 62 95 123 ...  
 $ drat: num 3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...  
 $ wt : num 2.62 2.88 2.32 3.21 3.44 ...  
 $ qsec: num 16.5 17 18.6 19.4 17 ...  
 $ vs : num 0 0 1 1 0 1 0 1 1 1 ...  
 $ am : num 1 1 1 0 0 0 0 0 0 0 ...  
 $ gear: num 4 4 4 3 3 3 3 4 4 4 ...  
 $ carb: num 4 4 1 1 2 1 4 2 2 4 ...  
> # Affichage des 5 premières lignes de la table  
> head(mtcars, n = 5)  
      mpg  cyl  disp  hp  drat   wt  qsec vs  am  gear  carb  
Mazda RX4      21.0    6  160 110  3.90 2.620 16.46 0  1    4    4  
Mazda RX4 Wag  21.0    6  160 110  3.90 2.875 17.02 0  1    4    4  
Datsun 710     22.8    4  108  93  3.85 2.320 18.61 1  1    4    1  
Hornet 4 Drive 21.4    6  258 110  3.08 3.215 19.44 1  0    3    1  
Hornet Sportabout 18.7    8  360 175  3.15 3.440 17.02 0  0    3    2
```

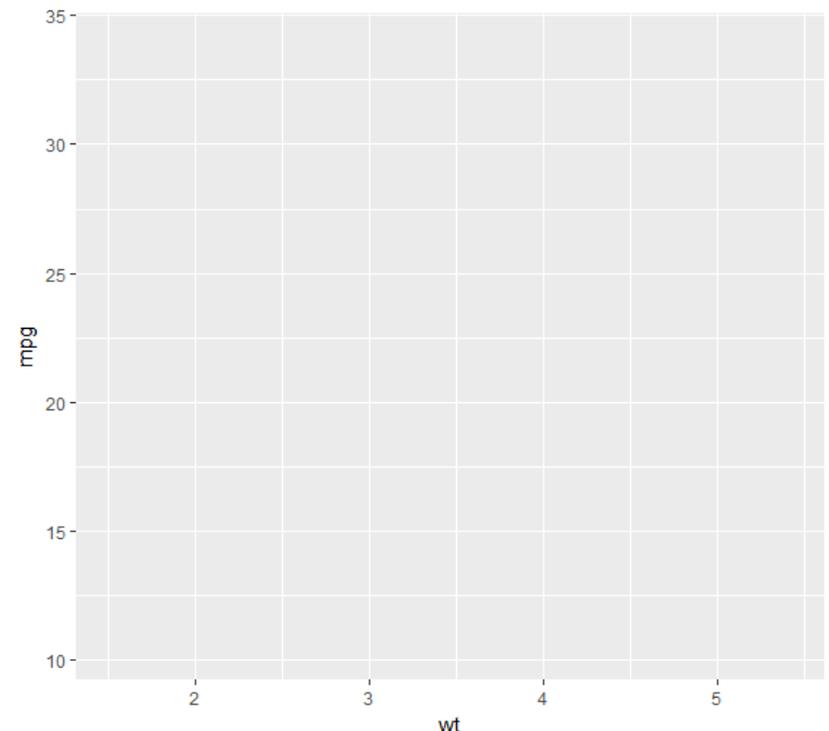
# Construire un graphique avec ggplot2 : un exemple

```
# Initialisation d'un premier graphique
p <- ggplot(data = mtcars, mapping = aes(x = mpg, y=wt))

> # Initialisation d'un premier graphique
> p <- ggplot(data = mtcars, mapping = aes(x = wt, y=mpg))

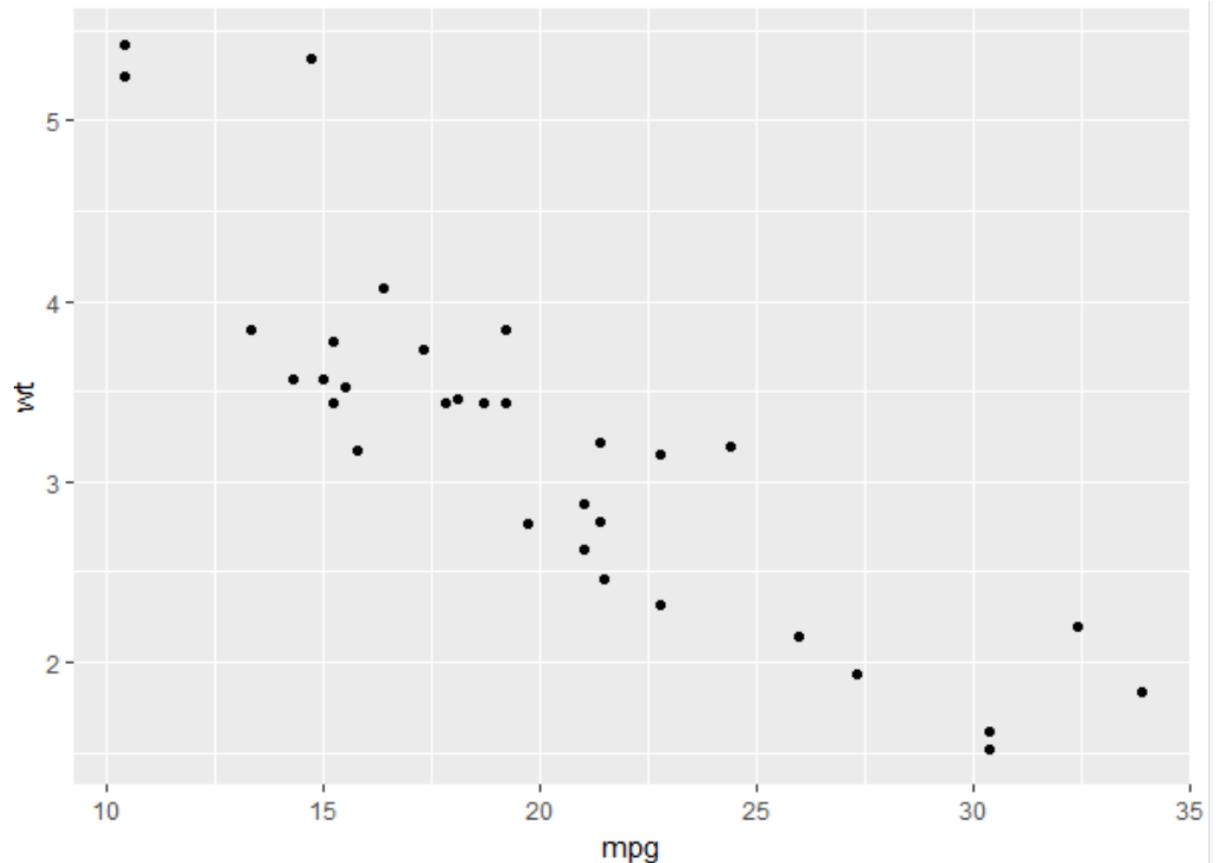
# affichage de l'objet p
p
```

Pour l'instant, rien ne s'affiche, nous n'avons pas ajouter de couches permettant de définir une géométrie



# Construire un graphique avec ggplot2 : un exemple

```
# ajouter une géométrie  
p + geom_point()
```



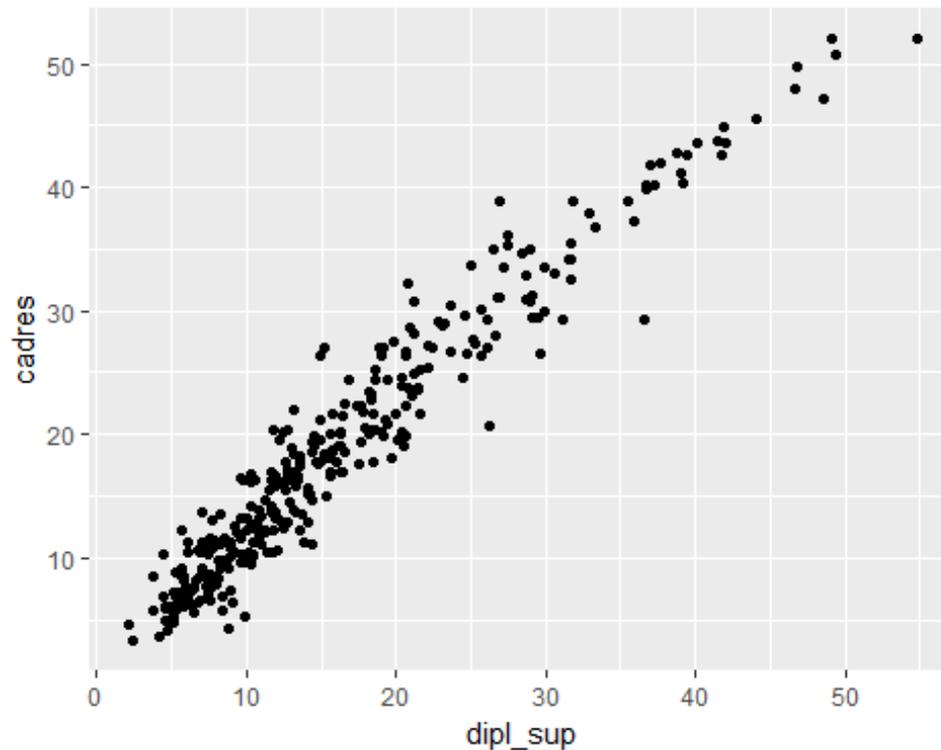
Concevoir ses graphiques sous R – Package  
ggplot2/initiation – prise en main

## 2 – Maîtriser les notions d' « esthétique » et d'attributs

# Modifications d'attributs ou modification d'esthétique ?

- Attributs

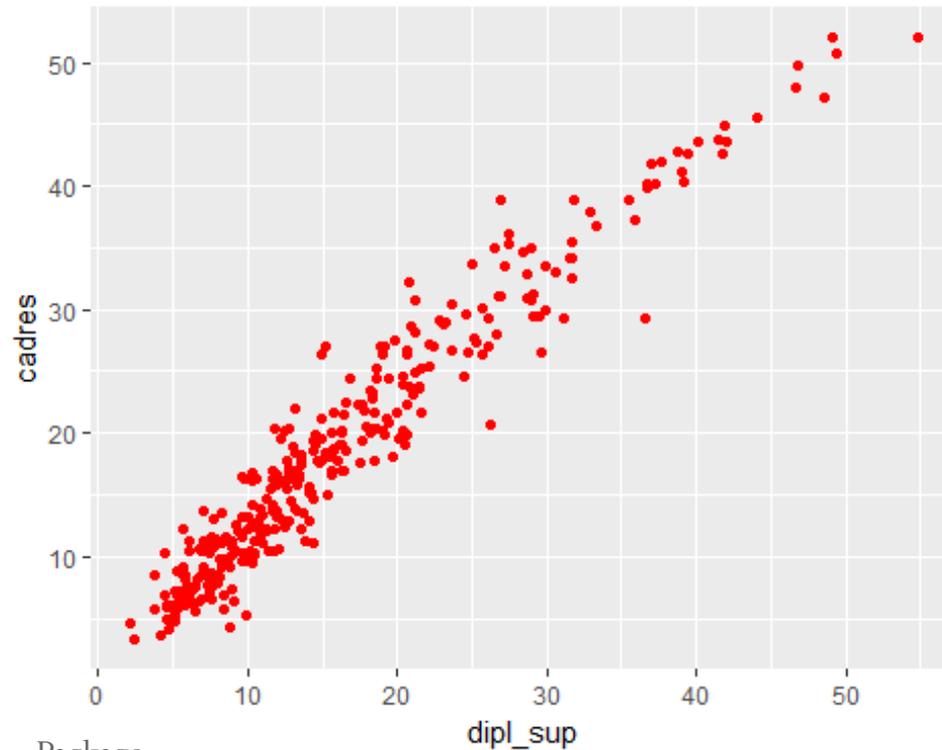
```
ggplot(rp) +  
  geom_point(aes(x = dipl_sup, y = cadres))
```



# Modifier un attribut

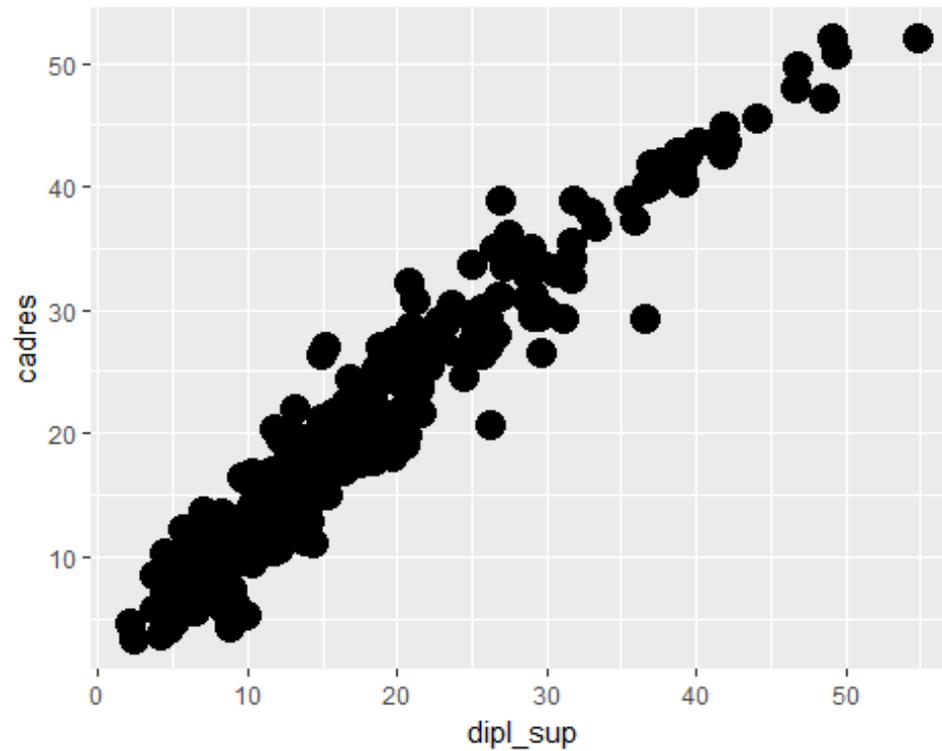
- Attributs

```
ggplot(rp) +  
  geom_point(aes(x = dipl_sup, y = cadres), color = "red")
```



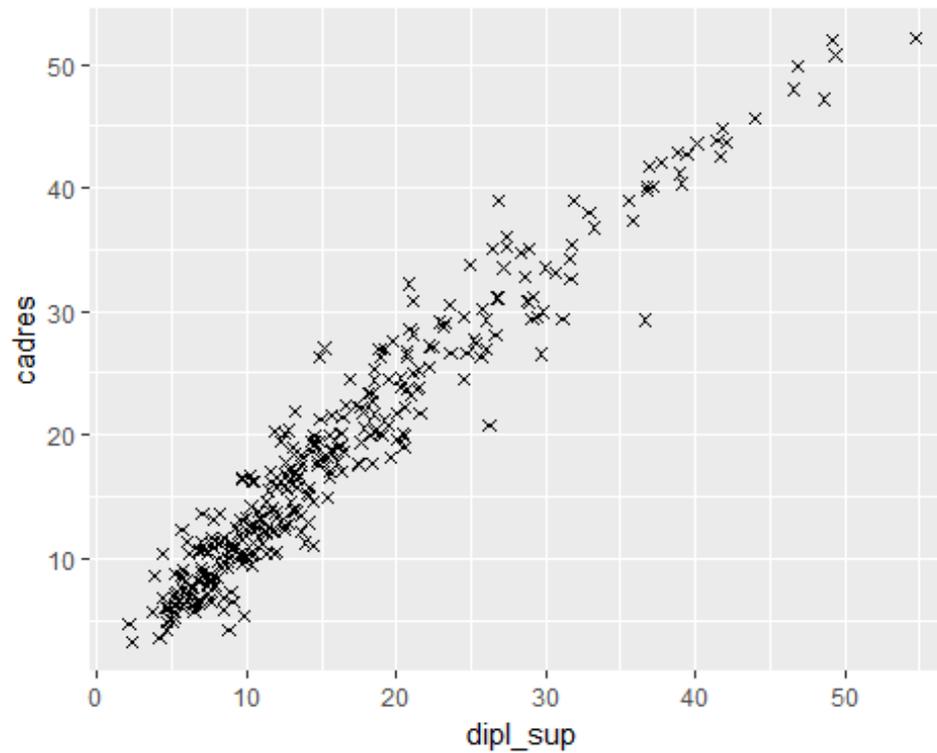
- Attributs

```
ggplot(rp) +  
  geom_point(aes(x = dipl_sup, y = cadres), size = 5)
```



- Attributs

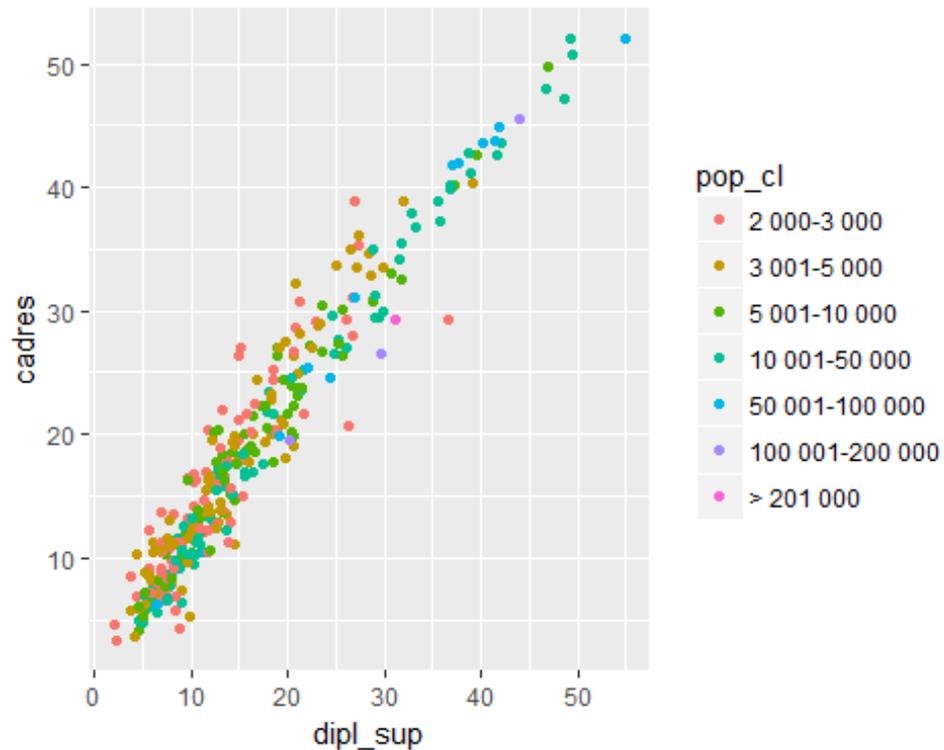
```
ggplot(rp) +  
  geom_point(aes(x = dipl_sup, y = cadres), shape = 4)
```



# Modification d'esthétique

- Esthétique aes()

```
ggplot(rp) +  
  geom_point(aes(x = dipl_sup, y = cadres, colour = pop_cl))
```



# Modifier l'esthétique

Aesthetic	Description
x	X axis position
y	Y axis position
colour	Colour of dots, outlines of other shapes
fill	Fill colour
size	Diameter of points, thickness of lines
alpha	Transparency
linetype	Line dash pattern
labels	Text on a plot or axes
shape	Shape

# En résumé

- La construction d'un graphique fonctionne sur le principe d'un ajout de couches successives
- 1 – avec la fonction `ggplot()`, on déclare :
  - Le dataset sur lequel on va travailler
  - Les variables à représenter (argument mapping et fonction `aes()` qui peuvent être déclarés ailleurs)
- 2 – on déclare la géométrie, c'est-à-dire le type de graphique souhaité, avec les fonctions préfixés par `geom_XXX()`

# Les types de géométries disponibles dans le package ggplot2

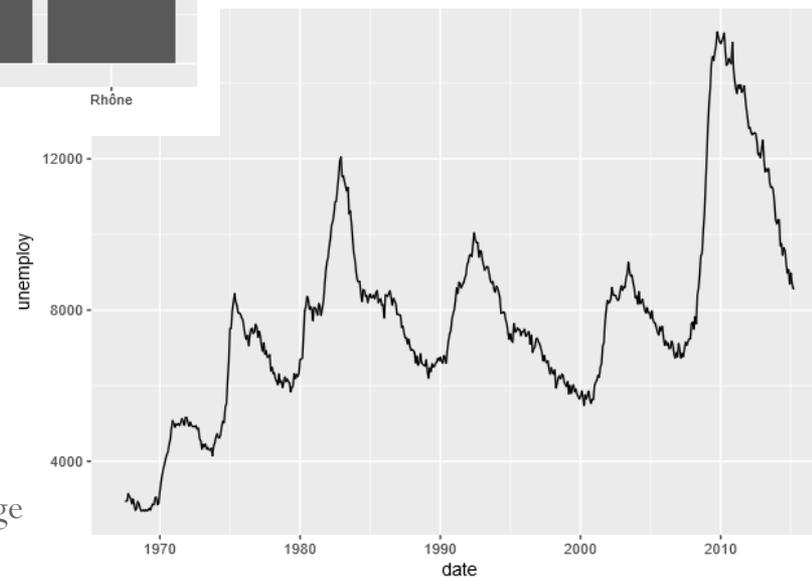
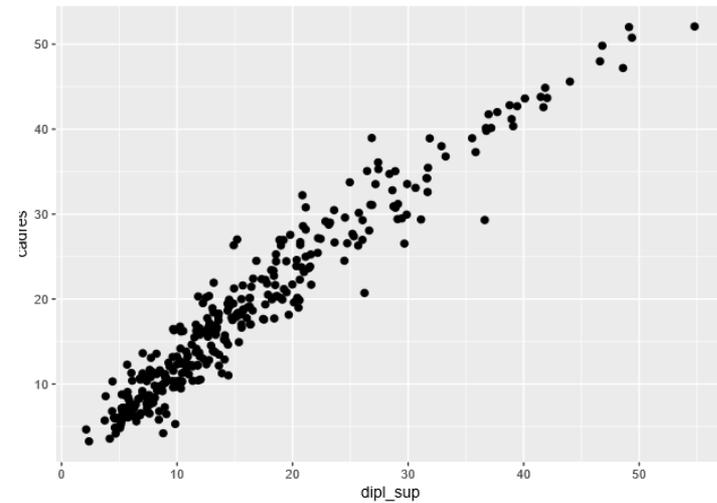
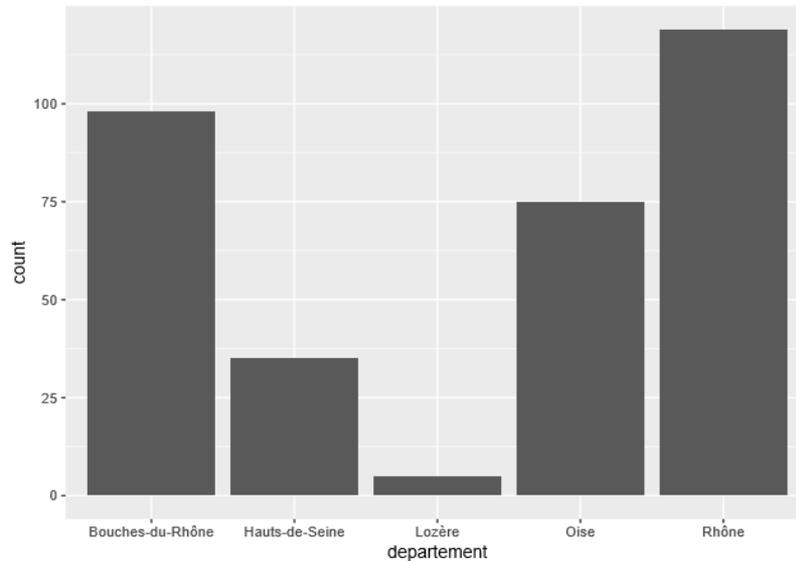
## 37 Geometries

abline	density2d	line	rect	vline
area	dotplot	linerrange	ribbon	
bar	errorbar	map	rug	
bin2d	errorbarh	path	segment	
blank	freqpoly	point	smooth	
boxplot	hex	pointrange	step	
contour	histogram	polygon	text	
crossbar	hline	quantile	tile	
density	jitter	raster	violin	

# Les géométries courantes

- `geom_bar()` : graphique en barres (équivalent de la fonction `barplot()`)
- `geom_histogram` : histogramme (équivalent de la fonction `hist()`)
- `geom_point()` : nuage de points (équivalent de `plot()`)
- `geom_line()` : points reliés par un segment

# Les géométries courantes



# 3 – Maîtriser les autres couches pour améliorer ses graphiques

# Tous les éléments de grammaire...

Élément	Description
Data	Les données à afficher
Aesthetics	Les dimensions selon lesquelles les données seront 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)
Themes	Description de l'arrière plan

## ... on peut y ajouter d'autres couches

- `ggtitle(« »)` : ajoute un titre principal au graphique
- `xlab(« »)` : change le libellé des abscisses
- `ylab(« »)` : change le libellé des ordonnées
  
- `labs( title = « », x = « », y = « »)` : modifie titre principal, libellé des abscisses, libellé des ordonnées en une seule fonction
  
- ...

# 4 – Mise en forme : comprendre le fonctionnement de la couche « theme »

# Theme

- 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 que l’on peut adapter. On peut également créer son propre theme.

# Theme

- La fonction `theme()` est utilisée pour contrôler les éléments graphiques :
  - 3 types :
    - Éléments textuels `element_text()`
    - Éléments linéaires `element_line()`
    - Éléments rectangulaires `element_rect()`

# Themes existants

- **theme\_gray** : couleur de fond grise et grille blanche
- **theme\_bw** : couleur de fond blanche et grille grise
- **theme\_linedraw**: traits noirs autour du graphique
- **theme\_light**: traits et axes légèrement gris (plus d'attention accordée aux données)
- **theme\_minimal**: pas de couleur de fond
- **theme\_classic** : thème avec les traits des axes mais pas de grilles
- **theme\_void**: Thème vide, util pour des graphs avec des coordonnées non-standard ou pour dessiner
- **theme\_dark**: Thème avec un arrière plan noir, designé pour faire ressortir les couleurs.

# 5 – Les outils pour mettre en forme ses graphiques

- Documentation

<http://ggplot2.tidyverse.org/reference/>

<http://dataviz.tools/>

- Choix palette de couleurs

[https:// colors.co/](https://colors.co/) générateur de palette de couleur

<http://chir.ag/projects/name-that-color/#ED616A> nommer la couleur

<http://colorbrewer2.org/#type=sequential&scheme=BuGn&n=3> site ColorBrewer

<https://flatuicolors.com/> Choisir une couleur

# Pour aller plus loin

- Package ggvis
- Package ggtheme
  
- Site <http://ggplot2.tidyverse.org/>
- Site <https://juba.github.io/tidyverse/>
- Site <http://www.sthda.com/french/wiki/ggplot2>
- Github : <https://github.com/tidyverse/ggplot2>