Visualisation de données avec R — TP1

Arthur Katossky

Janvier 2019

Contents

Introduction

Les premières sections de cette introduction sont fortement inspirée du chapitre introductif de Munzner (2014).

La visualisation de données, une représentation *visuelle*

La visualisation repose sur la vision car c'est le sens le plus adapté comme support de mémoire externe:

- .....
- .....
- .....
- .....

Une représentation *complexe*

La visualisation présente des données les plus désagrégées possibles, pour ne pas tomber dans le piège de la synthèse (**Figure 1**).

Ce n'est évidemment pas toujours possible quand les données deviennent beaucoup trop nombreuses. Mais l'idée est d'utiliser les avantages de la vision pour transmettre beaucoup plus d'information que ce qu'il est possible de proprement *comprendre*.

Une représentation *efficace*

La visualisation se distingue du design, de la publicité ou de l'art par .....
.....
.....

Malheureusement .....
.....
.....

Dans ce cours, nous nous intéresserons surtout à .....
.....

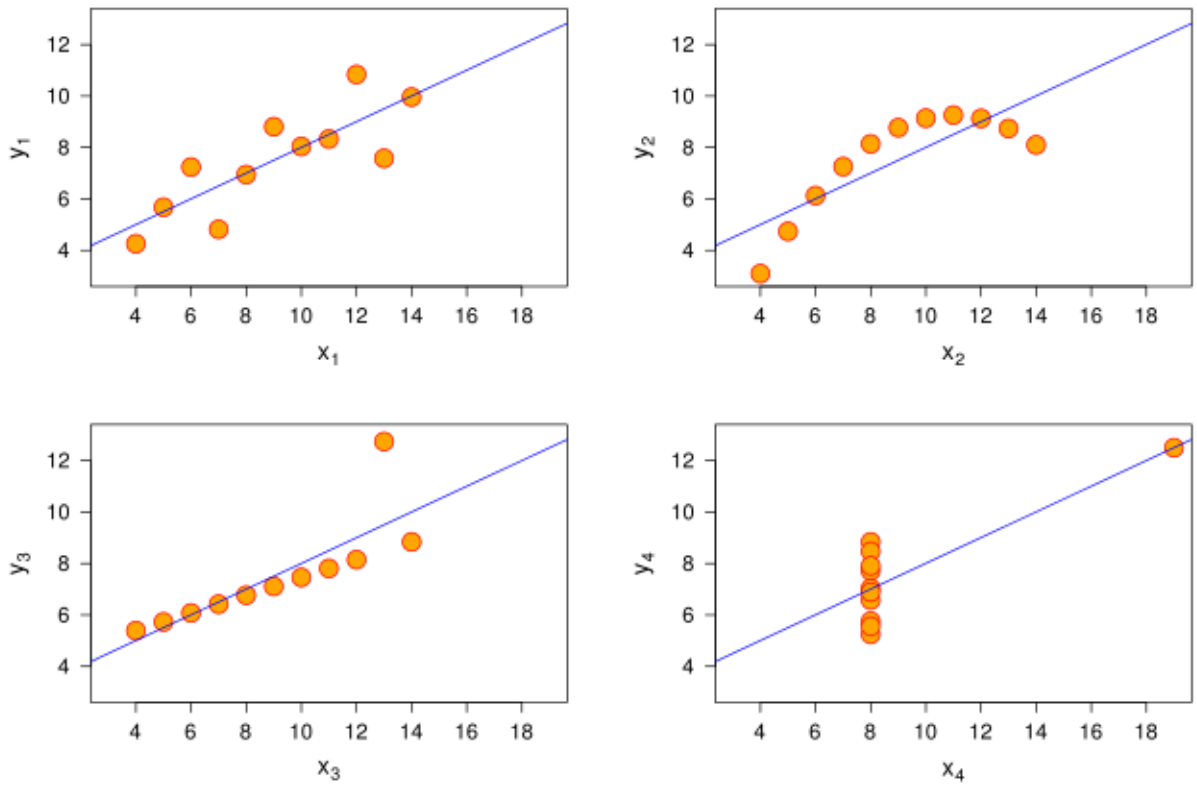


Figure 1: **Le piège de la synthèse.** Le quartet d'Anscombe est cet ensemble de quatre droites de régression identiques, bien que construites sur des données très différentes. La visualisation permet de proposer simmulaténement une lecture agrégée et désagrégée des données. **Source de l'image:** Wikipédia

.....

Mais même avec une tâche précise et avec une bonne définition de "l'efficacité" d'une visualisation pour cette tâche, il est extrêmement difficile/fastidieux de "valider" une visualisation. Aussi, les informations expérimentales sur la visualisation sont plutôt rares.

Une représentation subjective

.....
.....
.....
.....

`ggplot` et `tidyverse`

Nous utiliserons dans ce cours la bibliothèque `ggplot2`, inspiré de *The Grammar of Graphics* (Wilkinson, 2005), qui permet de réaliser des graphiques d'une étonnante richesse, tout en gardant un code clair.

Cette bibliothèque fait partie de la série de bibliothèques `tidyverse`, maintenue et développée les programmeurs de RStudio, au premier rang desquels Hadley Wickham. Ces bibliothèques rendent plus cohérentes de nombreuses fonctionsde R-base, qui fonctionnaient jusqu'alors selon chacune leur logique propre. La plupart étant transparentes pour l'utilisateur, je les utilise sans le mentionner explicitement. Mais n'hésitez pas à me poser des questions sur des fonctions qui vous sont inconnues.

La seule nouveauté sur laquelle je vais insister dès maintenant, c'est la notation `%>%` (aka *pipe*), qui permet de remplacer:

- `f(x)` par `x %>% f` et
- `f(x,a)` par `x %>% f(a)`.

Cela paraît un peu ridicule à premier abord (c'est plus long!), mais cela permet de faire de très longues chaînes de fonctions qui, si les fonctions ont été pensées pour, sont très lisibles.

Par exemple `paste0(rev(toupper(letters)), collapse='')` devient:

```
letters %>% ..................................................
```

Les données

Nous travaillerons avec des données Eurostat, principalement des données de population au niveau NUTS 2 (le découpage officiel statistique européeen, qui possède 3 niveaux). Le tableau de données `NUTS2_year` possède une ligne par région NUTS 2 et par année d'observation:

```
## # A tibble: 9,019 x 15
##    id_anc année superficie comments population_femm~ population_homm~
##    <chr>  <int>      <dbl> <chr>               <int>            <int>
##  1 AT11    2015       3669 <NA>               147246           141110
##  2 AT12    2015      18917 <NA>               832975           803803
##  3 AT13    2015        395 <NA>               929704           867633
##  4 AT21    2015       9360 <NA>               286371           271270
##  5 AT22    2015      16251 <NA>               621265           600305
##  6 AT31    2015      11717 <NA>               727840           709411
##  7 AT32    2015       7050 <NA>               276378           262197
##  8 AT33    2015      12514 <NA>               370936           357890
```

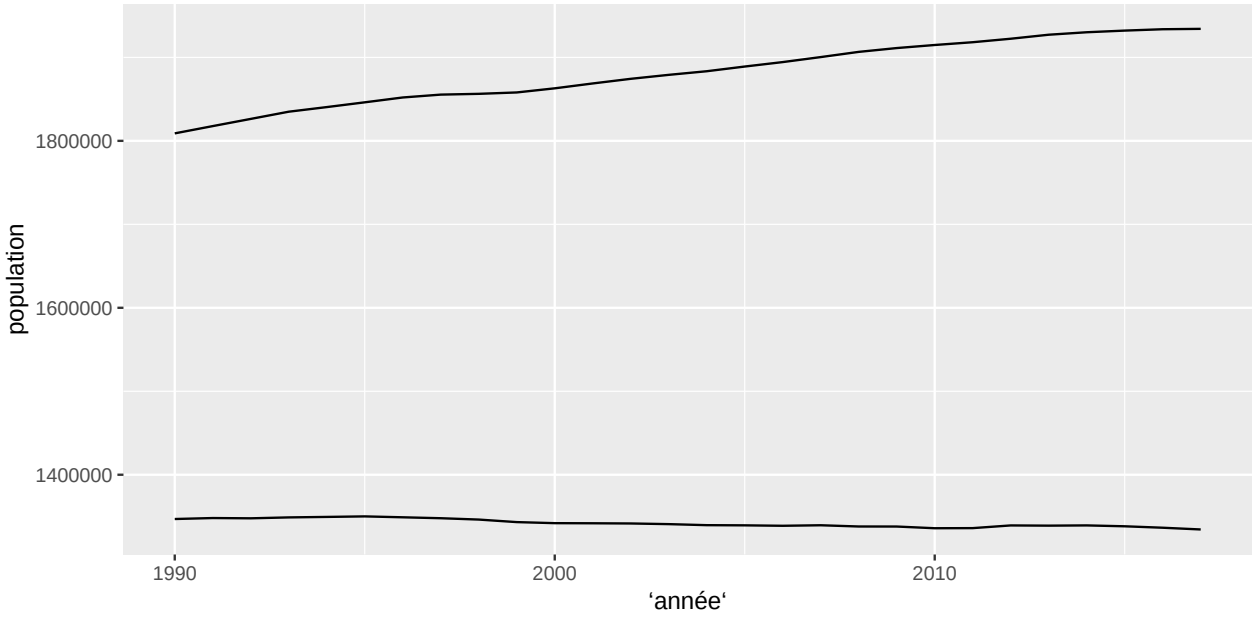
```
##  9 AT34    2015       2534 <NA>               191814           186778
## 10 BE10    2015        161 <NA>               605416           578685
## # ... with 9,009 more rows, and 9 more variables: population <int>,
## #   population_0_19 <int>, population_20_59 <int>,
## #   population_60_plus <int>, id <chr>, nom_anc <chr>, nom <chr>,
## #   chgt <fct>, anc <list>
```

1. Faire un graphique avec ggplot2

1.1 Graphique basique

Commençons par un exemple: l'évolution de la population en Champagne-Ardenne et en Picardie entre 1990 et 2015.

```
NUTS2_year %>%
  filter(nom %in% c('Champagne-Ardenne', 'Picardie')) %>%
  ggplot(aes(x=année, y=population)) +
  geom_line(aes(group=nom))
```



Le code du plus simple graphique `ggplot2` possède trois parties:

1. `data %>% ggplot(...)`: .....

   Notez que `data %>% ggplot(...)` est la même chose que `ggplot(data, ...)`, mais cela est plus lisible quand comme ici je transforme mes données à la vollée avec `filter`. Ainsi:

   ```
   NUTS2_year %>% filter(nom %in% c('Champagne-Ardenne', 'Picardie')) %>% ggplot
   ```

   ... est l'équivalent en R-base de:

   ```
   ggplot(.....................................................................)
   ```

2. `aes(...)`: .....

   Les paramètres graphiques spécifiés dans `ggplot()` sont valables pour **tout** le graphique.

3. `geom_...(...)`: .....

Chaque fonction `geom_...` produit une nouvelle couche (*layer*) contenant les figures (*geometries*) indiquées par le suffixe de la fonction. Ici, `geom_line` produit des lignes (*lines*). Les fonctions graphiques réutilisent les paramètres fournis dans l'argument `aes()` de la fonction `ggplot()`, ici les coordonnées `x=année` et `y=population`. Mais chaque couche peut avoir en sus ses propres paramètres graphiques additionnels. Ici, tous les points correspondant à une même région NUTS 2 (`group=nom`) seront reliés entre eux par des segments, constituant ainsi une ligne.

Exercice 1.1.1 Représentez sous forme d'un nuage de points la superficie (`superficie`, en abscisse) et la population (`population`en ordonnée) des régions NUTS 2 en 2015. Notez le nom de la fonction `geom_...` que vous avez utilisé.

.....

```
NUTS2_year %>%
  filter(.....) %>%
  ggplot(aes(x=....., y=.....)) +
  geom_.....()
```

Avec R Studio*, utilisez l'autocomplétion (touche* `tab`*). Souvenez-vous en effet que toutes les fonctions graphiques commencent par* `geom`*.*

Exercice 1.1.2 Il est possible de superposer plusieurs couches graphiques. Pouvez-vous rajouter des étiquettes à chaque observation? Notez le nom de la fonction que vous avez utilisé.

.....

- *Utilisez la documentation (*`?fonction`*) pour connaître les paramètre graphiques obligatoires.*
- ***Rappel:** Il est possible d'appeler la fonction* `aes()` *dans chaque couche graphique.*

En attendant... Les étiquettes et les points du graphique précédent se chevauchent mutuellement. Comment faire pour décaler — en anglais *to nudge* — une étiquette par rapport à sa position réelle, afin de ne pas recouvrir les points avec les étiquettes? Est-ce que cela résout le problème totalement? Quelles autres améliorations sont possibles?

.....
.....
.....
.....
.....

1.2 Options de présentation

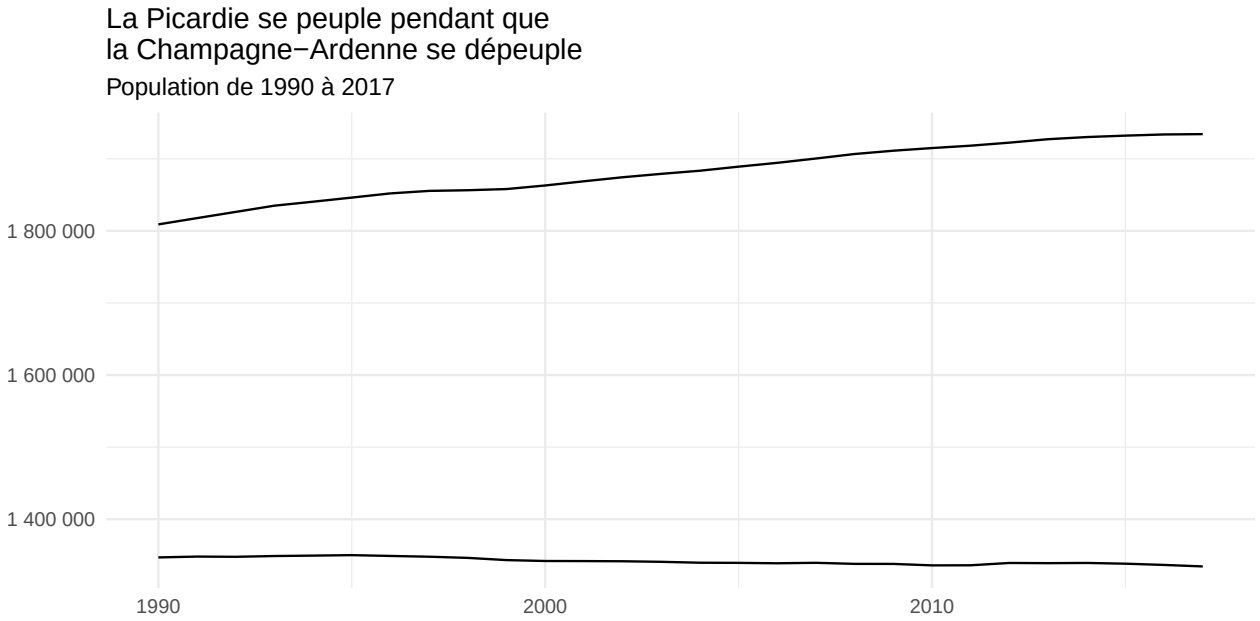
Le graphique Picardie—Champagne-Ardenne obtenu précédemment n'est pas satisfaisant:

1. .....
2. .....
3. .....
4. .....

Des fonctions supplémentaires permettent de régler ces problèmes. Revoici le même graphique qu'initialement, mais avec une mise en page améliorée:

```
NUTS2_year %>%
  filter(nom %in% c('Champagne-Ardenne', 'Picardie')) %>%
  ggplot(aes(x=année, y=population, group=nom)) +
  geom_line() +
  # changer le style de graphique
  theme_minimal() +
  # supprimer les titres des deux axes
```

```r
# utiliser un format plus lisible sur l'axe des ordonnées
scale_x_continuous(name = NULL) +
scale_y_continuous(name = NULL, labels = scales:::number)  +
# ajouter titre et sous-titre
labs(
  title    = "La Picardie se peuple pendant que\nla Champagne-Ardenne se dépeuple",
  subtitle = "Population de 1990 à 2017"
)
```



Exercice 1.2.1 Repérez les éléments qui permettent…

- de supprimer le nom des axes: .....
- d'adopter un thème général plus sobre: .....
- de donner un titre, un sous-titre: .....
- de contrôler le format d'affichage sur les axes: .....

Exercice 1.2.2 À quoi correspond le `\n` dans le code du titre?

.....

Exercice 1.2.3 Le graphique Picardie—Champagne-Ardenne reste cependant incomplet. Que lui manque-t-il avant de pouvoir être publié ?

.....
.....
.....
.....
.....

Exercice 1.6 Reprenez votre graphique population—superficie et habillez-le pour le rendre publiable.

N'oubliez pas de donner un titre et sous-titre à votre graphique. Je recommande de donner pour titre une phrase-choc (ex: "La Normandie, destination rêvée des français.") et pour sous-titre un descriptif plus neutre (ex: "Nombre de nuités par région en 2015").

En attendant... Sur votre graphique population-superficie, explorez quelques thèmes de graphique. Notez le noms de ceux qui vous plaisent. Parmi eux, notez le nom de ceux qui vous semblent les plus *efficaces*.

1. *Tous les thèmes de graphiques commencent par* `theme_...`. *Vous pouvez donc utiliser l'autocomplétion de* RStudio *!*
2. *Vous pouvez créer un objet* `ggplot` *avec* `object <- ggplot(data, aes(...)) + geom_...()` *puis jouer avec les thèmes avec* `objet + theme_...()`.

. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .

2. Représenter des données continues

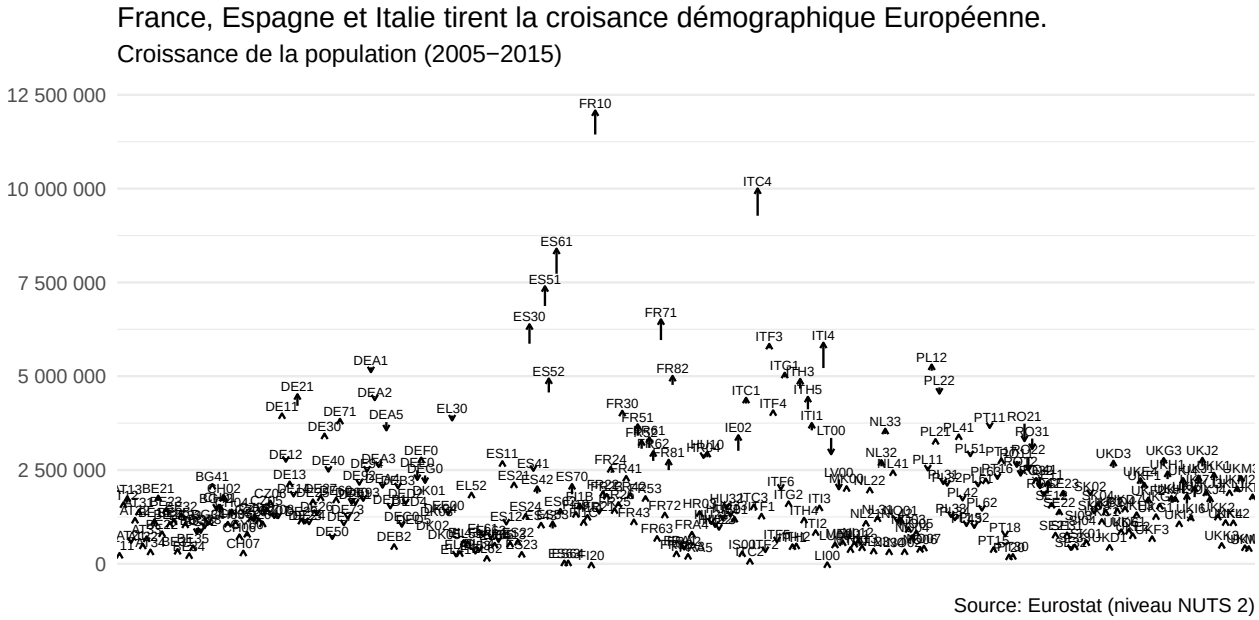
Des données peuvent être représentées visuellement de très, très nombreuses façons. Nous allons explorer quelques-uns de ces **canaux** de transmission (en anglais *channels*) et tenter de les hiérarchiser. Pour commencer, occupons nous des canaux les plus usuels: ceux basés sur les **longueurs** (longueurs au sens strict, mais aussi la **position** dans le plan).

Nous nous tournerons ensuite vers les *angles* et la *pente*, avant de passer à la **couleur** et à la **superficie**.

2.1 Longueurs

Voici la trame d'un graphique qui repésente le changement démographique entre 2005 et 2015, chaque région NUTS 2 étant représentée par une flèche. L'origine de la flèche part de la population en 2005, et la pointe arrive à la population en 2015.

```
NUTS2_year %>%
  # pré-traitement
  filter(!str_detect(id_anc, '^TR')) %>% # <-------------------------------- (1)
  filter(.....................) %>%
  arrange(année) %>% # <---------------------------------------------------- (A)
  # graphique (cœur)
  ggplot(aes(x=................., y=.....................)) +
  geom_line(arrow = arrow(length = unit(0.1, "cm"))) +
  # ajouter des étiquettes
  geom_text(
    data = . %>%
      group_by(id_anc) %>%
      summarise(population=max(population)), # <--------------------------- (2)
    aes(label=id_anc),
    size=2, # <---------------------------------------------------------------- (3)
    nudge_y=200000 # <-------------------------------------------------------- (B)
  ) +
  # graphique (mise en page)
  scale_x_discrete(name=NULL, breaks=NULL) + # <---------------------------- (C)
  scale_y_continuous(name=NULL, labels = scales:::number) +
  theme_minimal() +
  labs(
    title    = "France, Espagne et Italie tirent la croisance démographique Européenne.",
    subtitle = "Croissance de la population (2005-2015)",
    caption = "Source: Eurostat (niveau NUTS 2)"
  )
```



Exercice 2.1.1 À quoi correspondent l'élément (1)? En particulier, à quoi correspond le `'^'`?

.....
.....
.....

Exercice 2.1.2 Au niveau de l'élément (2), la combinaison `group_by() %>% summarize()` permet de faire des opérations par groupe. De quelle opération s'agit-il? En quoi est-ce utile?

.....
.....
.....

Exercice 2.1.3 À quoi correspondent l'élément (3)? Pourquoi ne pas mettre `size` à l'intérieur de la fonction `aes()`?

.....
.....
.....

Exercice 2.1.4 Complétez la trame.

Exercice 2.1.5 Que se passe-t-il si j'efface la ligne `arrange(...) %>%` (A)? Si je supprime `nudge_y=...` dans la fonction `geom_text()` (B)? Que se passe-t-il si j'enlève `breaks=NULL` dans la fonction `scale_x_discrete()` (C)?

.....
.....
.....
.....
.....
.....
.....

Exercice 2.1.6 La croissance démographique est-elle plus grande en île de France (`FR10`) ou à Rome (`ITI4`)?

.....
.....
.....

. . . . . .
. . . . . .

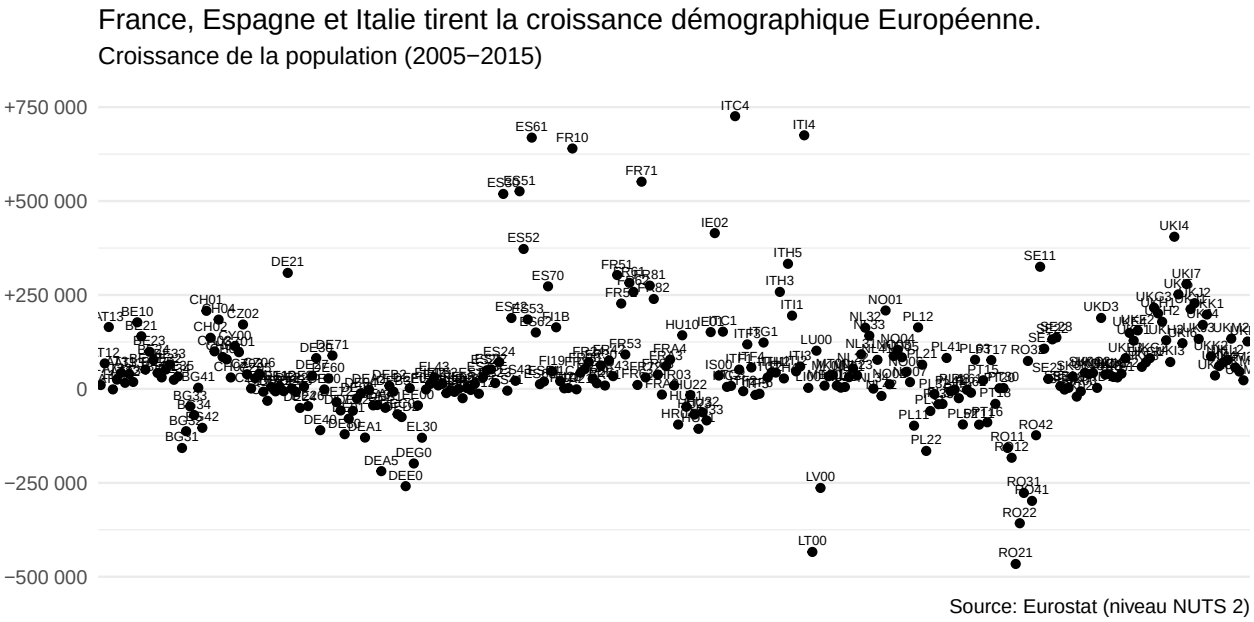
En attendant... Colorez la flêche en fonction de si la croissance et positive ou négative.

En attendant... Comment est-il possible de réordonner les flèches pour nous aider à répondre à la question précédente?

- *La fonction* `reorder` *aide à réordonner les modalités d'une variable de type* `factor`.

. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .

2.2 Position



Exercice 2.2.1 Complétez la trame suivante pour obtenir le graphique ci-dessus. La croissance démographique entre 2005 et 2015 est en ordonnée, et les régions en abscisse.

- *Inspirez vous librement des exercices précédents!*

```
NUTS2_year %>%
  ..................
  ..................
  ..................
  ..................
  ..................
  ..................
  ..................
  ..................
  .................. +
  scale_x_discrete(  name=NULL, ............) +
  scale_y_continuous(name=NULL, ............) +
  theme_minimal() +
```

```
  labs(
    title    = "France, Espagne et Italie tirent la croissance démographique Européenne.",
    subtitle = "Croissance de la population (2005-2015)",
    caption  = "Source: Eurostat (niveau NUTS 2)"
  )
```

Exercice 2.2.2 La perte démographique est-elle plus grande au Saxe-Anhalt (`DEE0`) ou en Lettonie (`LV00`)?

.....
.....
.....

En attendant... Pour nous aider à répondre à la question précédente, colorez les points correspondants au Saxe-Anhalt (`DEE0`) ou et à la Lettonie (`LV00`), ainsi que, pour comparaison, les point correspondant à Paris (`FR10`) et Rome (`ITI4`) d'une autre couleur.

En attendant... Proposer une réorganisation des points qui facilite la comparaison.

2.3 Longueurs + position

Exercice 2.3.1 Changez une ligne du code de l'exercice 2.2.1 pour maintenant représenter la croissance démographiques à l'aide d'un diagramme en barres.

- *Par soucis de lisibilité, vous pouvez également retirer les régions de croissance démographique proche de zéro.*

```
NUTS2_year %>%
  filter(année %in% c(2005, 2015)) %>%
  group_by(id_anc) %>% summarize(
    croissance = population[order(année)] %>% {last(.)-first(.)}
  ) %>%
  filter(!is.na(croissance)) %>%
  ggplot(aes(x=id_anc, y=croissance)) +
  geom_point() +
  geom_text(
    aes(label = id_anc),
    size=2, nudge_y=30000
  ) +
  scale_x_discrete(  name=NULL, breaks=NULL) +
  scale_y_continuous(name=NULL, labels = number_plus) +
  theme_minimal() +
  labs(
    title    = "France, Espagne et Italie tirent la croissance démographique Européenne.",
    subtitle = "Croissance de la population (2005-2015)",
    caption  = "Source: Eurostat (niveau NUTS 2)"
  )
```

Exercice 2.3.2 La croissance démographique est-elle plus grande en Émilie-Romagne (`ITH5`) ou à Stockholm `SE11`?

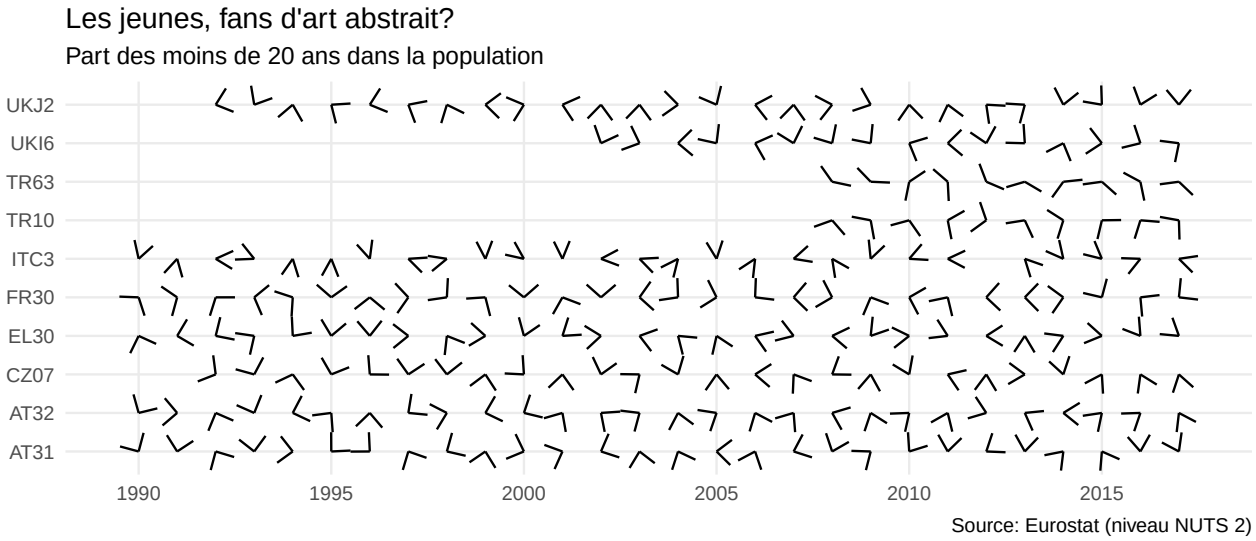
.....
.....
.....

Exercice 2.3.3 Rajouter une ligne au code suivant pour réorganiser les barres par ordre croissant (ou décroissant). Est-il plus facile de répondre?

En attendant... Faire en sorte que les étiquettes restent lisibles dans la partie haute et dans la partie basse du graphique.

2.4 Angles

Le graphique ci-dessous repésente la part des moins de 20 ans dans la population de quelques régions européennes sélectionnées au hasard.



Exercice 2.4.1: Voici la trame du code utilisé pour produire le graphique ci-dessus. À quoi correspondent les parties numérotées? Pourquoi est-ce important de spécifier `coord_equal()` (A)?

```
NUTS2_year %>%
  # pré-traitement
  filter(!is.na(population), !is.na(population_0_19)) %>%
  filter(id_anc %in% sample(unique(id_anc), size = 10)) %>% # <------------- (1)
  mutate(angle_seed = runif(n(), 0, 2*pi)) %>% # <---------------------------(2)
  # graphique (cœur)
  ggplot(aes(y=................, x=.....................)) +
  # ------------------------------------------------------------------ DÉBUT (3)
  geom_spoke(aes(angle=angle_seed), radius=0.5) +
  geom_spoke(aes(angle=angle_seed+.............................. ), radius=0.5) +
  # -------------------------------------------------------------------- FIN (3)
```

```r
# graphique (mise en page)
scale_y_discrete(name=NULL)+
scale_x_continuous(name=NULL, breaks=seq(1990,2020,5), minor_breaks = NULL) +
theme_minimal() +
coord_equal() + # <------------------------------------------------------- (A)
labs(
  title    = "Les jeunes, présents de façon uniforme en Europe",
  subtitle = 'Part des moins de 20 ans dans la population',
  caption  = "Source: Eurostat (niveau NUTS 2)"
)
```

Exercice 2.4.2: Complétez la trame.

Exercice 2.4.3: Est-ce facile de comparer différentes zones à différents moments dans le temps? Arrives-tu à repérer une zone NUTS avec plus de 30% de moins de 20 ans?

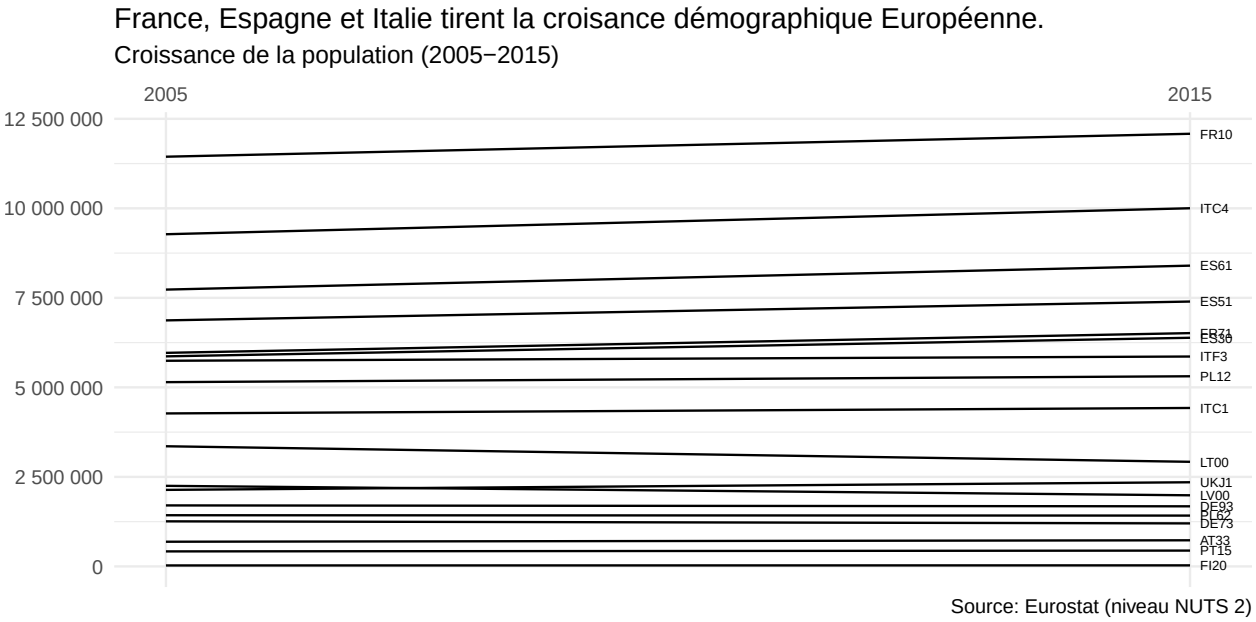
.....
.....
.....

Exercice 2.4.4: Modifier le code précédent de façon à ce que tous les angles aient un côté commun. Est-ce plus facile de comparer différentes régions?

.....
.....
.....

2.5 Pente

Le graphique ci-dessous repésente la croissance démographique entre 2005 et 2015 dans un petit nombre de régions européennes sélectionnées arbitrairement.



Exercice 2.5.1: Voici la trame du code utilisé pour produire le graphique. À quoi correspondent les parties numérotées? Pourquoi est-ce important de **ne pas** utiliser `coord_equal()`?

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

```
NUTS2_year %>%
  # pré-traitement
  filter(année %in% c(2005, 2015), id_anc %in% c(
    'ITC4', 'FR10', 'ES30', 'FR71', 'ES51', 'ES61', 'PL12', 'ITF3',
    'FI20', 'PT15', 'AT33', 'DE73', 'PL62', 'DE93', 'LV00',
    'UKJ1', 'LT00', 'ITC1'
  )) %>%
  # graphique (cœur)
  ggplot(aes(y=..................., x=................)) +
  geom_line(aes(group= ..............................)) +
  geom_text(aes(label= ..............................),
    data    = .%>% filter(année==2015), # <--------------------------------- (1)
    size    = 2, hjust=0,
    nudge_x = 0.1 # <------------------------------------------------------ (2)
  ) +
  # graphique (mise en page)
  scale_y_continuous(name=NULL, labels=scales:::number)+
  scale_x_continuous(
    name=NULL,
    breaks=c(2005L, 2015L), # <-------------------------------------------------
    minor_breaks = NULL, # <---------------------------------------------------- (3)
    labels=scales:::number_format(big.mark = ''), # <----------------------- (4)
    position = "top" # <-------------------------------------------------------- (5)
  ) +
  theme_minimal() +
  labs(
    title    = "France, Espagne et Italie tirent la croisance démographique Européenne.",
    subtitle = "Croissance de la population (2005-2015)",
    caption  = "Source: Eurostat (niveau NUTS 2)"
  )
```

Exercice 2.5.2: Complétez la trame.

Exercice 2.5.3: Arrivez-vous à repérer des NUTS avec des croissances de population comparable? Modifiez le code de la question 1.3 pour que la pente des droites soient plus prononcées. Est-ce plus facile de comparer plusieurs les zones NUTS entre elles?

- *Vous pouvez utiliser* `coord_fixed()` *avec des paramètres de votre choix.*

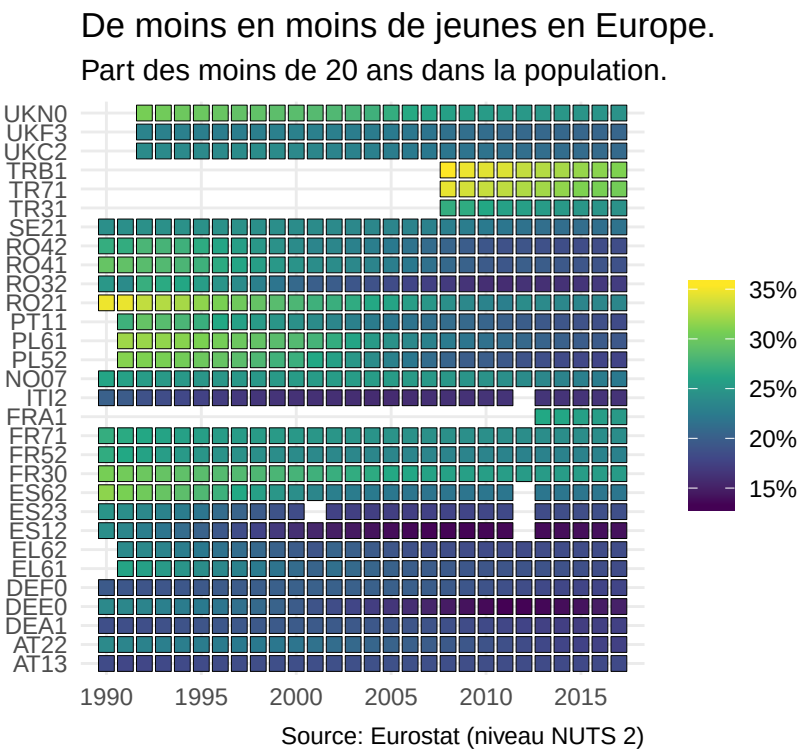
En attendant... Modifiez le code précédent en optant pour une base 0 en 2005. Est-ce plus facile de comparer plusieurs zones? Pour quelle raison à votre avis?

.....

En attendant... Ajoutez des couleurs pour aider à distinguer les lignes voisines! Avec trois ou quatre couleurs seulement, cela aide déja beaucoup.

- *affecter les couleurs aléatoirement est un premier pas, mais il est plus efficace d'alterner les couleurs en fonction de la population ;*
- *créer des classes est une bonne idée ;*
- *pensez à* `%/%` *;*
- *vous pouvez désactiver la légende (car les couleurs, ici, ne codent rien) avec* `guides(color=FALSE)`

2.6 Couleurs



Exercice 2.6.1: Voici la trame du code ayant servi à produire le graphique. À quoi correspondent les parties numérotées?

.....
.....

```
NUTS2_year %>%
  # pré-traitement
  filter(!is.na(population) & !is.na(population_0_19)) %>%
  filter(id_anc %in% sample(unique(id_anc), size = 30)) %>% # <------------- (1)
  # graphique (cœur)
  ggplot(aes(y=....................., x=.....................)) +
  geom_point(aes(.............=.............), size=3, shape=22) + # <--- (A)
  # gaphique (mise en page)
  scale_y_discrete(name=NULL)+
  scale_x_continuous(
    name=NULL,
    breaks=seq(1990,2020,5),
    minor_breaks = NULL
  ) +
  scale_fill_continuous( # <------------------------------------------------ (2)
    name = NULL,
    type = 'viridis',
    labels = scales:::percent_format(accuracy=1) # <----------------------- (3)
  ) +
  coord_equal() +
  theme_minimal() +
  labs(
    title    = "De moins en moins de jeunes en Europe.",
    subtitle = 'Part des moins de 20 ans dans la population.',
    caption  = "Source: Eurostat (niveau NUTS 2)"
  )
```

Exercice 2.6.2: Complétez la trame.

Exercice 2.6.3: Remplacez `shape=22` par `shape=15` puis par `shape=0` (A). Quelle différence notez-vous?

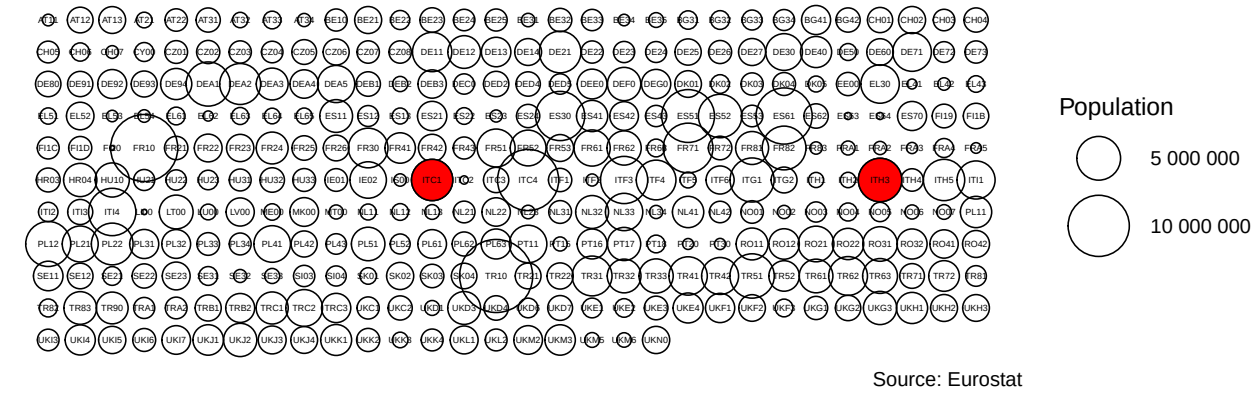
.....
.....
.....
.....
.....
.....
.....

Exercice 2.6.4: Avec quelle précision est-il possible d'identifier une région européenne avec 20% de chômage? À quel point est-il facile de comparer deux régions entre elles?

.....
.....
.....

En attendant... Modifiez le code pour utiliser une autre échelle de couleur, par exemple un dégradé du rose vers le jaune.

2.7 Superficie



Exercice 2.7.1: Voici la trame du code ayant servi à produire le graphique. À quoi correspondent les parties numérotées?

.....
.....
.....
.....
.....
.....
.....

```
NUTS2_year %>%
  # pré-traitement
  arrange(id_anc) %>%
  filter(année==2015, !is.na(population)) %>%
  mutate(
    y = -(1:n()) %/% 30,    # <----------------------
    x = 0:(n()-1) %%  30    # <---------------------------------------------- (1)
  ) %>%
  # graphique (cœur)
  ggplot(aes(....................................., size = ...................)) +
  geom_point(shape=21) +
  geom_point(
    data = . %>% filter(id_anc %in% c('ITC1', 'ITH3')), # <------------------(2)
    shape=21,
    fill='red',
    show.legend = FALSE  # <---------------------------------------------- (3)
  ) +
  geom_text(aes(......... = .............), size=1.5) +
  # graphique (mise en page)
  scale_y_discrete(name=NULL, breaks=NULL)+
  scale_x_continuous(name=NULL, breaks = NULL ) +
  scale_size_area(labels=scales:::number, max_size = 15) +
  theme_minimal() +
  coord_equal() + # <---------------------------------------------------- (4)
  labs(
```

```
    size     = "Population",
    title    = "Disparité des maillages administratifs en Europe",
    subtitle = 'Population des régions NUTS 2 (2015)',
    caption  = "Source: Eurostat"
  )
```

Exercice 2.7.2: Complétez la trame.

Exercice 2.7.2: Quel est la région NUTS 2 la plus peuplée: Turin (`ITC1`) ou Venise (`ITH3`) ? Est-ce facile de répondre?

.....
.....
.....

En attendant... À partir de ton expérience, penses-tu que l'humain est davantage sensible au périmètre / rayon du cercle ou au contraire à son aire? Remplace `scale_size_area` par `scale_radius` pour tester.

3. Représenter des données catégorielles

3.1 Formes

Exercice 3.1.1: reproduisez le graphique ci-dessus: dans un carré 10×10 chaque pictogramme représente l'âge de 1 centile de la population européenne en 2015, selon trois tranche d'âge (0-20 ans, 20-60 ans, 60 ans et plus).

- *Rappelez vous que la fonction* `ggplot()` *demande un tableau avec autant d'observations que de marqueurs à placer sur le graphique donc ici 100 lignes.*
- *Vous pouvez commencer avec des formes géométriques de base (avec* `geom_point` *et l'argument* `shape` *) puis essayer d'utiliser des icônes (bibliothèque* `ggimage` *) ou des emojis (bibliothèque* `emojifont` *).*

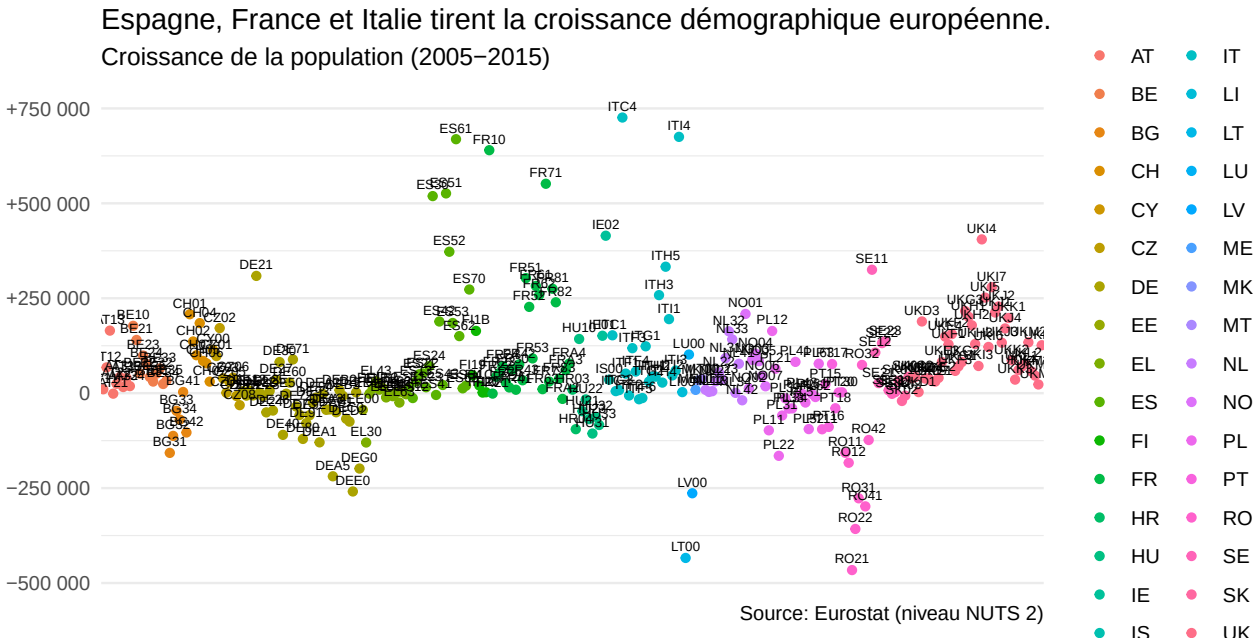
- *Comme à la section précédente, il est possible de supprimer tous les éléments d'arrière plan avec* `theme_void()`.

Exercice 3.1.2: Représentez la même information par un graphique en colonnes.

Exercice 3.1.3: Lequel des deux graphiques est le plus efficace? Dans quel cas l'utilisation d'un diagramme en pictrogrammes se justifie-t-il?

.....
.....
.....
.....
.....

3.2 Couleurs



Exercice 3.2.1: modifiez le code de l'exercice question 2.2.2 pour avoir une couleur par pays

- *Les identifiants commencent par deux lettres qui signalent le pays. N'hésitez pas à utiliser les fonction* `str_...` *de la bibliothèque* `stringr`.

Exercice 3.2.2: Est-il facile de repérer le point correspondant à la Chypre (`CY`) ou à l'Islande (`IS`)? La légende vous paraît-elle être celle d'une variable discrète?

.....
.....
.....
.....

Exercice 3.2.3: Explorer l'outil ColorBrewer (http://colorbrewer2.org), en mode données qualitatives. À partir de combien de couleurs devient-il difficile de conserver l'impression que les couleurs représentent réellement des catégories discontinues?

.....
.....

. . . . . .
. . . . . .

En attendant... Reproduire le graphique avec 8 couleurs "Color Brewer" pour les 8 pays avec le plus de régions NUTS 2 et une couleur par défaut (gris) pour les autres.

- *Il existe une fonction* `scale_colour_brewer()` *mais par défaut, l'échelle de couleur est continue.*

3.3 Regrouper, relier, encadrer

Pendant cette section, nous allons travailler sur des données simulées accessible avec l'objet `data`. `data` a pour variables `x` (l'abscisse), `y` (l'ordonnée) et `label` (l'identifiant de chaque observation).

Exercice 3.3.1: Complétez la trame suivante pour représenter ces données sous forme d'un nuage de points. Les labels doivent être lisibles, superposés *à l'intérieur* des points.

```
data %>% ggplot(aes(x=x, y=y)) +
    ..................................................
    ..................................................
    ..................................................
    .................................................. +
    theme_void()
```

Exercice 3.3.2: S'il se trouvait uniquement deux catégories dans les données, quels points de même catégorie vous attendriez-vous à trouver?

. . . . . .
. . . . . .
. . . . . .

Exercice 3.3.3: Ajouter une ligne entre les points D et G, et entre les points F et H. Quelle impression domine maintenant?

- *La ligne ne doit pas couvrir les points eux-mêmes!*

.....
.....
.....

Exercice 3.3.4: Dessiner un cadre (`geom_.....`) ou une ellipse (`geom_.....`) derrière ou autour ou des points G à K. Quelles impression domine finalement?

.....
.....
.....

En attendant... En reprenant le premier graphique de la section, attribuez aux points des couleurs ou des formes aléatoirement parmi 2. Votre perception des catégories implicites change-t-elle?

.....
.....
.....

4. Synthèse

Il n'y a pas unanimité entre les chercheurs en sciences cognitives sur la hiérarchie des vecteurs de représentation, pour les variables quantitatives et qualitatives. (Et encore moins sur des variables moins communes comme les variables discrètes ordonnées.) Néanmoins, il se dégage des lignes générales.

4.1. Variables quantitatives continues

Pour les variables quantitatives continues, retenons la hiérarchie suivante:

1. Position
2. Longueur
3. Superficie

Les vecteurs *pente* et *angle* ont une efficacité variable (du même niveau que les longueurs à pire que la superficie) et le vecteur *couleur* est au mieux du même niveau de précision que la superficie. Le *volume* arrive en dernier.

La **précision** d'un cannal de représentation est sa capacité à être perçu sans interférence par le système nerveux. Par exemple, seule la longueur est perçue de façon proportionnelle à son support physique (un segment deux fois plus long sur le papier est perçu comme deux fois plus long). En revanche la *profondeur* et la *luminosité* (couleur) sont perçues plus faiblement, et la *saturation* (couleur) plus fortement que leur contre-partie mesurable respective.

4.2. Variables qualitatives discrètes

Pour les variables qualitatives discrètes, la hiérarchie est différente:

1. Encadrement
2. Connexion
3. Proximité
4. Couleur
5. Forme

Évidemment, il n'est possible d'encadrer qu'un nombre très restreint d'objet sur un graphique. Ce canal est donc limité à des cas avec un nombre limité de données et de catégories: c'est le diagramme de Venn. En revanche, la connexion est largement utilisé, sous la forme classique du diagramme en ligne. En l'absence de toute connection, la proximité des marqueurs est naturellement perçus comme une variable catégorielle.

La *saillance* (en anglais *pop-out*) est la capacité de certains cannaux à singulariser un petit nombre de marqueurs. Tous les canaux de sont pas égaux: il est quai-immédiat d'identifier un marqueur de couleur parmi des marqueurs noirs, ou, dans un contexte interactif, un marqueur agité d'un moubement parmi des marqueurs statiques.

4.3. Autres considérations

La **redondance** de plusieurs vecteurs améliore grandement l'efficacité. Par exemple (position+longueur+superficie) est la combinaison la plus efficace connue, avec le diagramme en barres. Le diagramme de pentes, qui utilise la combinaison (pente+position), est lui-aussi très efficace.

Il y a **interférence** (*channel interference*) lorsque certains canaux se parasitent mutuellement. Si je décide d'utiliser à la fois la taille d'un marqueur (sa *superficie*) et sa *couleur*, les objets plus gros auront l'air d'avoir une couleur plus vive, par contraste avec le fond. Des canaux qui n'interfèrent pas sont dit **séparables** (les coordonnées x et y, par exemple) et des canaux dont l'interférence est complète sont dit **intégrés** (par exemples les canaux rouge, bleu et vert dans le modèle de couleur RGB).

La **distinguabilité** (*discriminability*) d'un canal est d'autant plus grande que l'œil peut discerner de niveaux séparés sur ce canal. Par exemple la largeur d'une ligne a une faible distinguabilité, car à partir d'un certain moment, c'est la superficie du rectangle qui prend le pas sur la largeur du trait. En revanche, la longueur, la position, la couleur.. permettent chacun à l'œil de distinguer plus d'une centaine de valeurs.

5. Critiquer un graphique

5.1. Introduction

Nous allons maintenant réutiliser nos connaissances pour critiquer des graphiques trouvés au fil de mes lectures. Certains sont particulièrement mauvais, d'autres particulièrement réussis. Le but de ces 3 séances de 30 min est d'apprendre à critiquer un graphique de façon structurée, en réutilisant les notions découvertes.

1) N'hésitez pas à me faire suivre des graphiques que vous trouvez particulièrement intéressants (peu importe si c'est parce qu'ils posent problème ou non)
2) À la fin des deux premières séances, je vous propose un graphique à critiquer et, surtout, à reproduire avec `ggplot2` en deux versions: une version identique et, surtout, une version améliorée pour tenir compte des critiques. Nous corrigerons ces graphiques à la séance suivante. (Le partiel consiste en une telle critique—reproduction d'un graphique par groupe de 2 élèves.)

5.2 Principes

Pour structurer nos critiques, nous utiliserons le « *Trifeca Checkup* » développé par Kaiser Fung sur sont blog *Junk Charts*, après plusieurs années de critique quotidienne. En effet, il ne suffit pas de dire qu'un graphique « n'est pas joli » ou « ne fonctionne pas » pour que cela constitue une critique pertinente, à même de permettre aux auteurs de proposer de meilleures alternatives. Notez bien que, de toute façon, **nous ne jugeons pas ici de la qualité esthétique du graphique**, ce qui est en dehors de nos compétences, en plus d'être extrêmement subjectif.

Le *Trifeca Checkup* s'articule autour de trois questions:

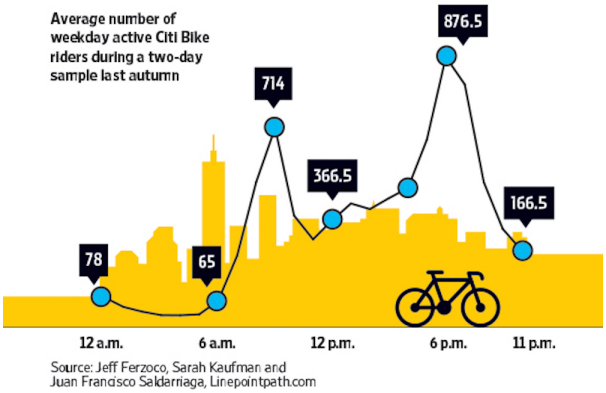
- la question à laquelle répond le graphe est-elle claire? (Q)
- les données mobilisées permettent-elles de répondre à la question posée? (D)

- la représentation visuelle choisie utilise-t-elle correctement les données pour répondre à la question? (V)

Le graphique idéal combine une question claire (Q) et des données (D) et un visuel (V) adaptés. C'est hélas loin d'être le cas.

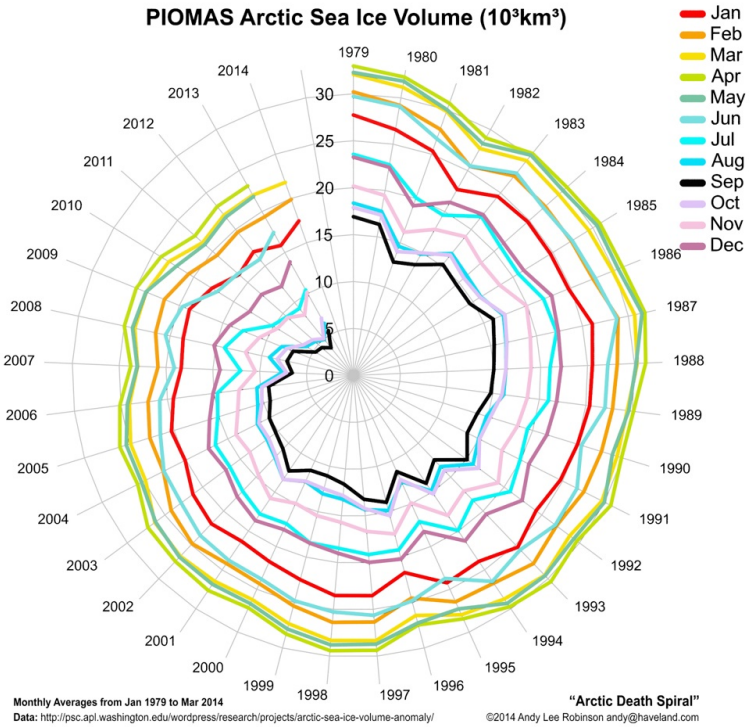
5.3 Exemples

Exemple 1



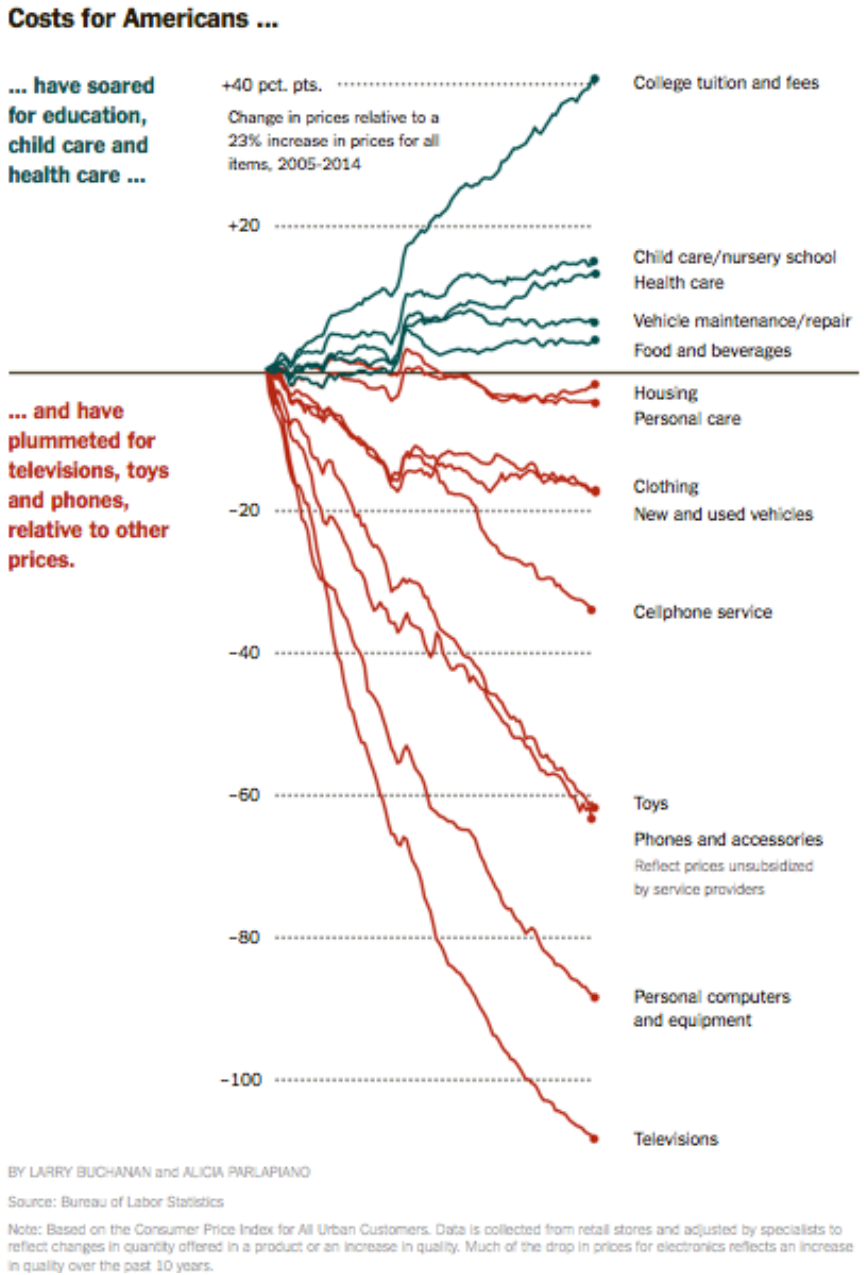
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .

Exemple 2

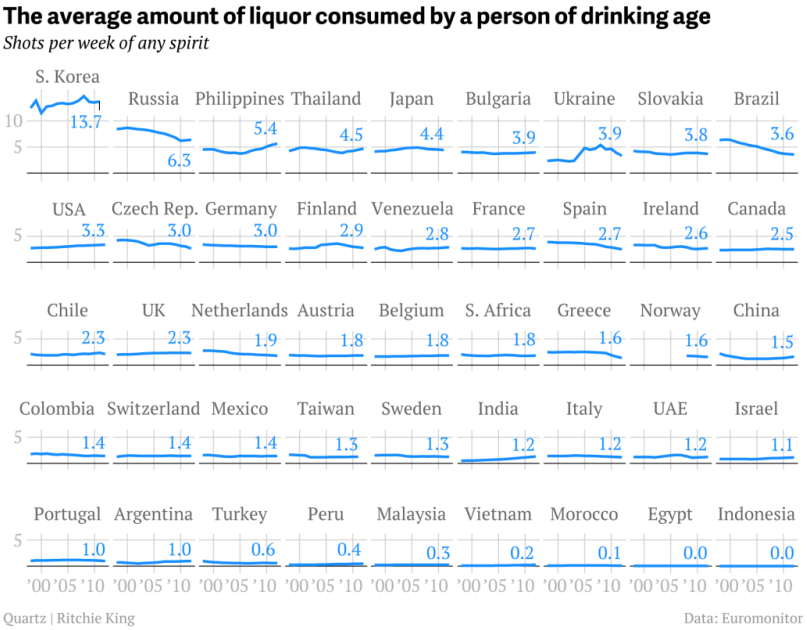


. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .

Exemple 3 (source)

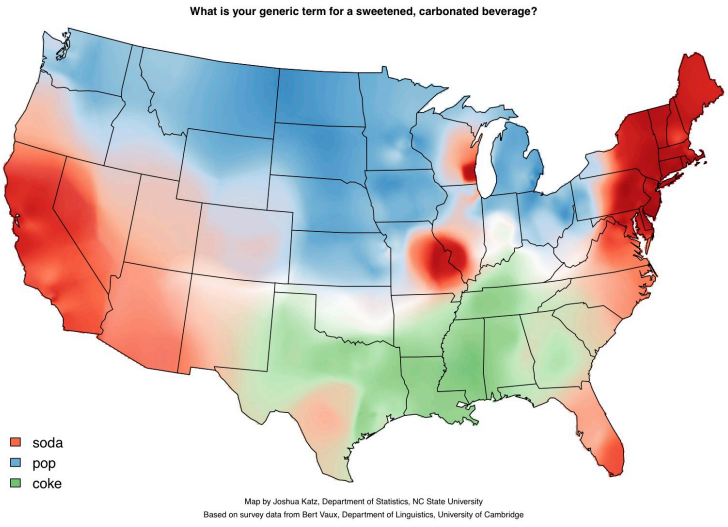


Exemple 4 (source)



. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .

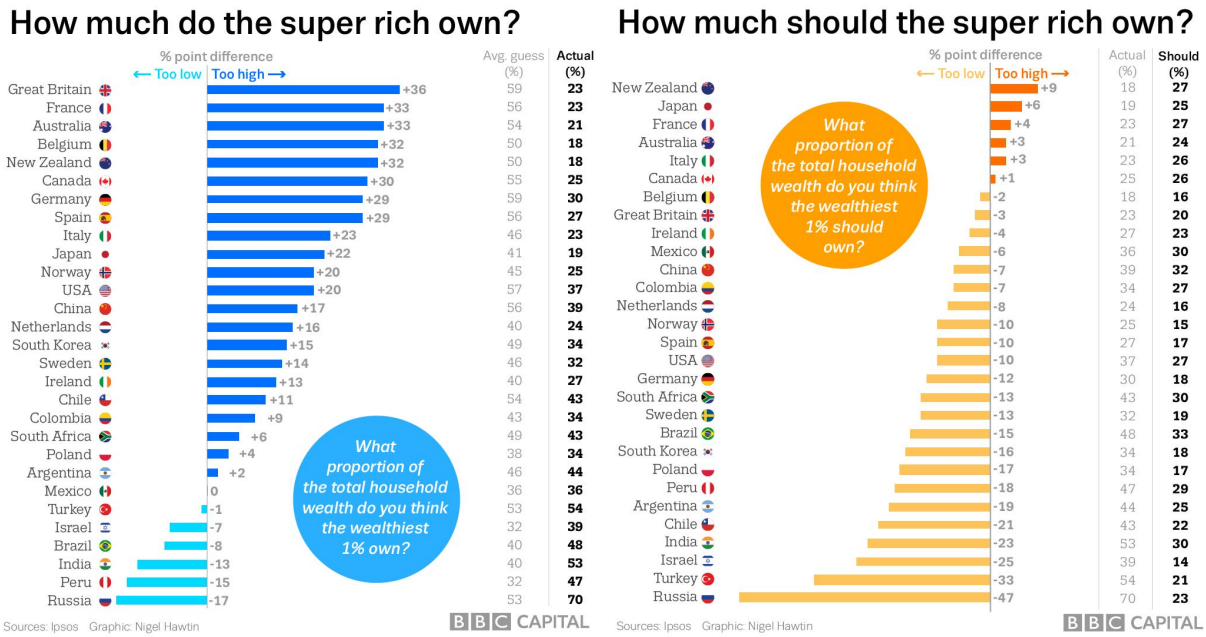
Exemple 5 (source)



. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .

5.4 Graphique pour la semaine prochaine

1. Produisez une critique constructive du graphique suivant
2. Proposez une visualisation alternative des données (`data/how-much-the-rich.csv`)



Source.

Ressources

Livres

- *Visualisation Analysis and Desgin*, de Tamara Munzner (2014) Une nomenclature universelle des graphiques, très détaillée. Un effort particulier est fait sur l'utilisation du vocabulaire, pour permettre aux statisticiens, graphistes et cognitivistes, etc. de se comprendre.
- *Data Visualization, A practical introduction*, de Kieran Healy (2018) Une introduction plus souple, mais moins exhaustive, dans le monde des graphiques, disponible également gratuitement en ligne.
- *The Grammar of Graphics*, de Leland Wilkinson (2005) L'ouvrage qui a inspiré la bibliothèque `ggplot2`.

Blogs et Twitter

- The Pudding
- Junk Charts, de Kaiser Fung (@junkcharts)
- Edward Tufte (@EdwardTufte)

Technique

- Stackoverflow, où il est possible filtrer uniquement les questions relatives à une technologie avec la syntaxe `[mot-clé]` dans la barre de recherche
- Le site de `tidyverse` (lien), où vous trouverez des introductions à `ggplot2`, `dplyr` (fonctions `%>%`, `mutate`, `select`, `filter`, etc.), `tidyr` (fonctions `spread` et `gather`) et bien d'autres packages utiles (`lubridate` pour les dates, `forcats` pour les facteurs). Je recommande également les nombreuses fiches-outils (*cheat-sheets*) mises à diposition.
- Le site STHDA (lien), avec ses nombreux articles dédiés à `ggplot2`.