Visualisation de données avec R – TP3

Arthur Katossky

Janvier 2019

Contents

Introduction ~ 10 min.

Dans ce TP, nous approfondirons deux aspects de la visualisation de données:

1. la visualisation de données denses
2. la scénarisation d'un graphique

Nous finirons par des discussions autour d'une poignée de graphiques.

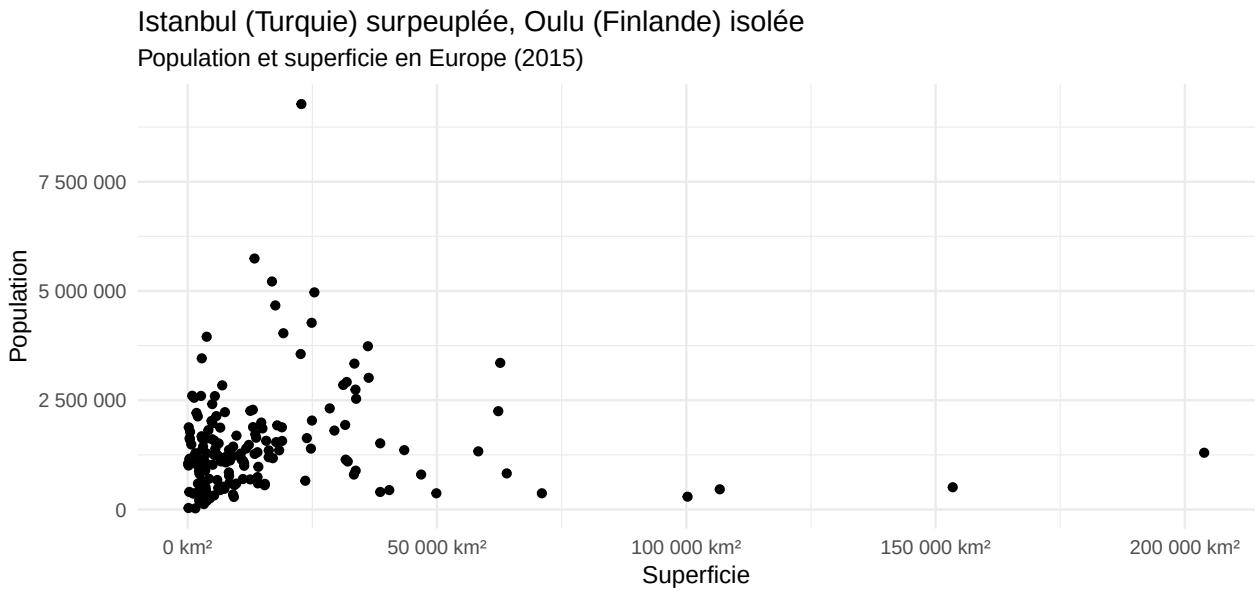
Visualisation de données denses

Idées à faire passer: 1. toutes les formes de graphique ne sont pas adpatées à toutes les données

En cartographie, problème quand trop de figurés au même endroit. Ex des lignes de train. Il faut tricher pour que l'on puisse continuer à voir les données. On a un compromis à faire entre "généralisation" (aggréger l'information), "ajustement" (décaler les marqueurs par rapport aux données) et lisibilité.

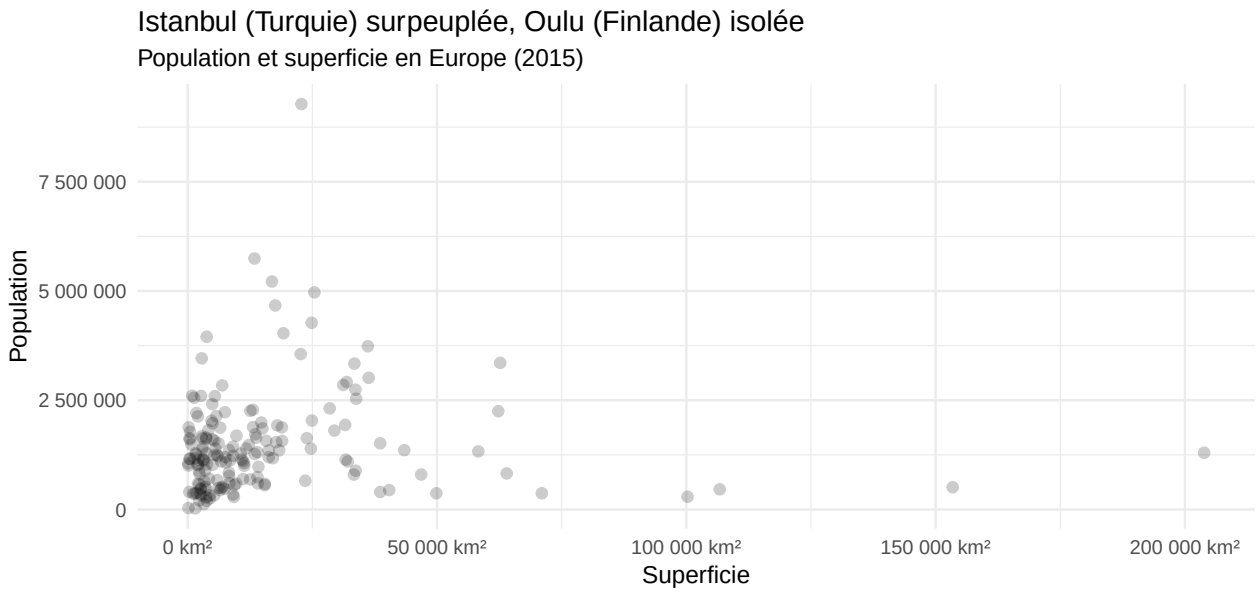
1.2. 2 variables continues

```
NUTS2_year %>%
  filter(!is.na(population), !is.na(superficie), année == 2005) %>%
  ggplot(aes(x=superficie, y=population)) +
  geom_point() +
  # geom_text(aes(label=id_anc), size=2) +
  scale_x_continuous(
    labels = function(x){str_c(scales:::number(x), ' km²')}
  ) +
  scale_y_continuous(
    labels = scales:::number
  ) +
  theme_minimal() +
  labs(
    x        = 'Superficie',
    y        = 'Population',
    title    = "Istanbul (Turquie) surpeuplée, Oulu (Finlande) isolée",
    subtitle = "Population et superficie en Europe (2015)",
    caption  = "Source: Eurostat (niveau NUTS 2)"
  )
```



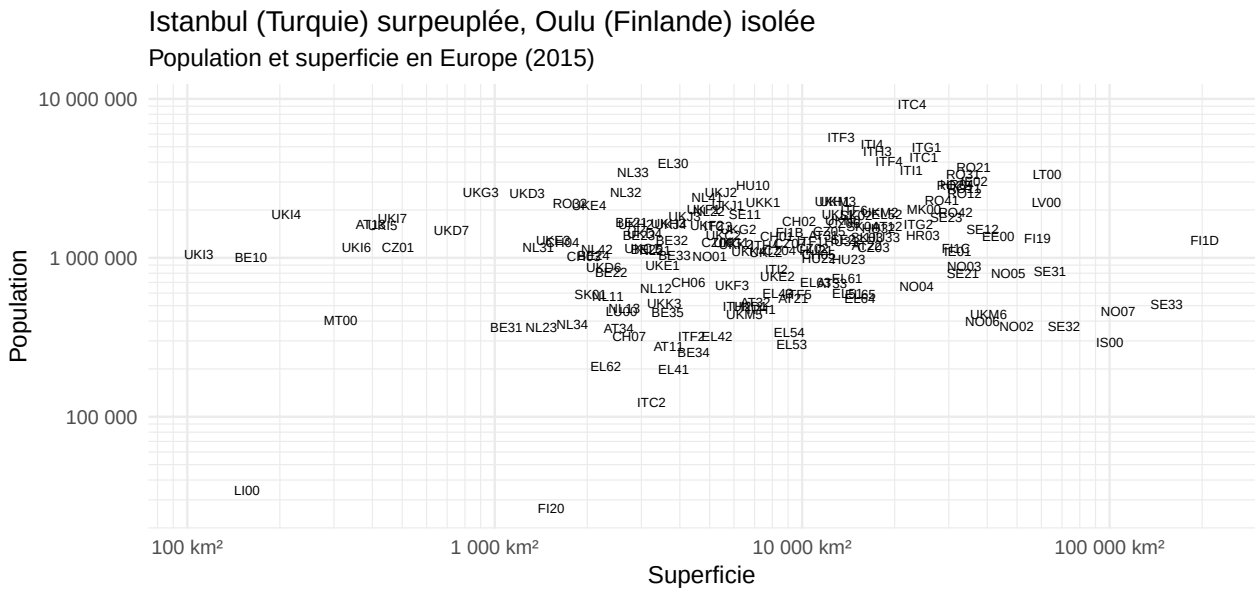
1.2.0 Transparence

```
NUTS2_year %>%
  filter(!is.na(population), !is.na(superficie), année == 2005) %>%
  ggplot(aes(x=superficie, y=population)) +
  geom_point(alpha=0.2, size=2) +
  scale_x_continuous(
    labels = function(x){str_c(scales:::number(x), ' km²')}
  ) +
  scale_y_continuous(
    labels = scales:::number
  ) +
  theme_minimal() +
  labs(
    x        = 'Superficie',
    y        = 'Population',
    title    = "Istanbul (Turquie) surpeuplée, Oulu (Finlande) isolée",
    subtitle = "Population et superficie en Europe (2015)",
    caption  = "Source: Eurostat (niveau NUTS 2)"
  )
```



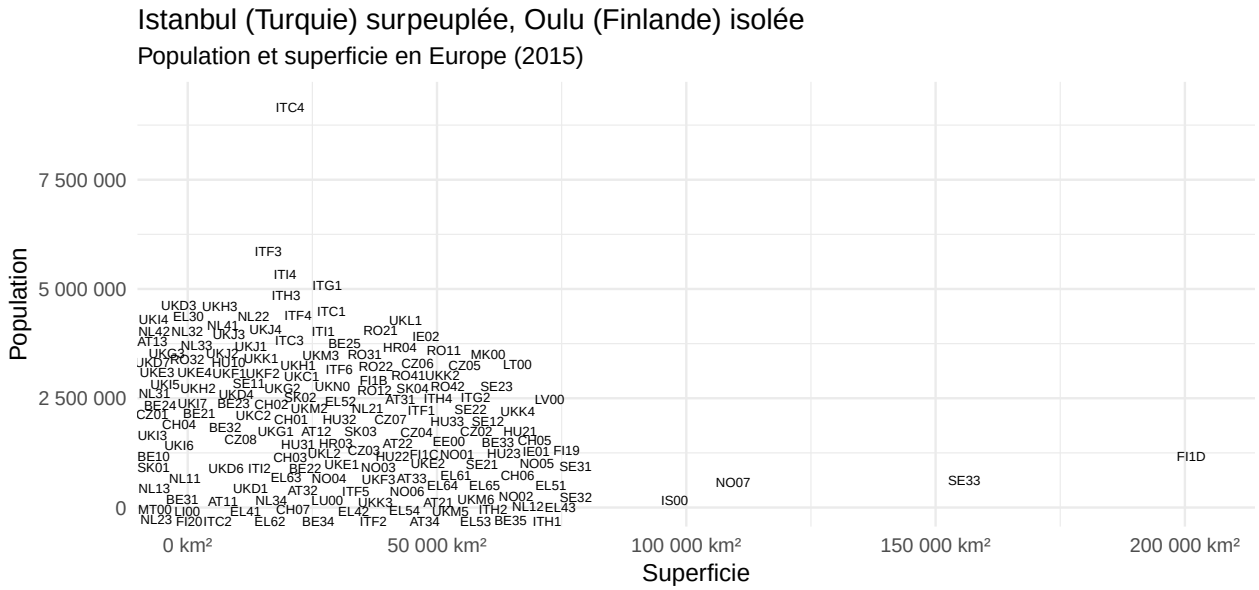
1.2.1 Transformation des axes

```
NUTS2_year %>%
  filter(!is.na(population), !is.na(superficie), année == 2005) %>%
  ggplot(aes(x=superficie, y=population)) +
  geom_text(aes(label=id_anc), size=2) +
  scale_x_log10(
    minor_breaks = rep(1:10, times=10)*10^rep(1:10, each=10),
    labels = function(x){str_c(scales:::number(x), ' km²')}
  ) +
  scale_y_log10(
    minor_breaks = rep(1:10, times=10)*10^rep(1:10, each=10),
    labels = scales:::number
  ) +
  theme_minimal() +
  labs(
    x        = 'Superficie',
    y        = 'Population',
    title    = "Istanbul (Turquie) surpeuplée, Oulu (Finlande) isolée",
    subtitle = "Population et superficie en Europe (2015)",
    caption  = "Source: Eurostat (niveau NUTS 2)"
  )
```



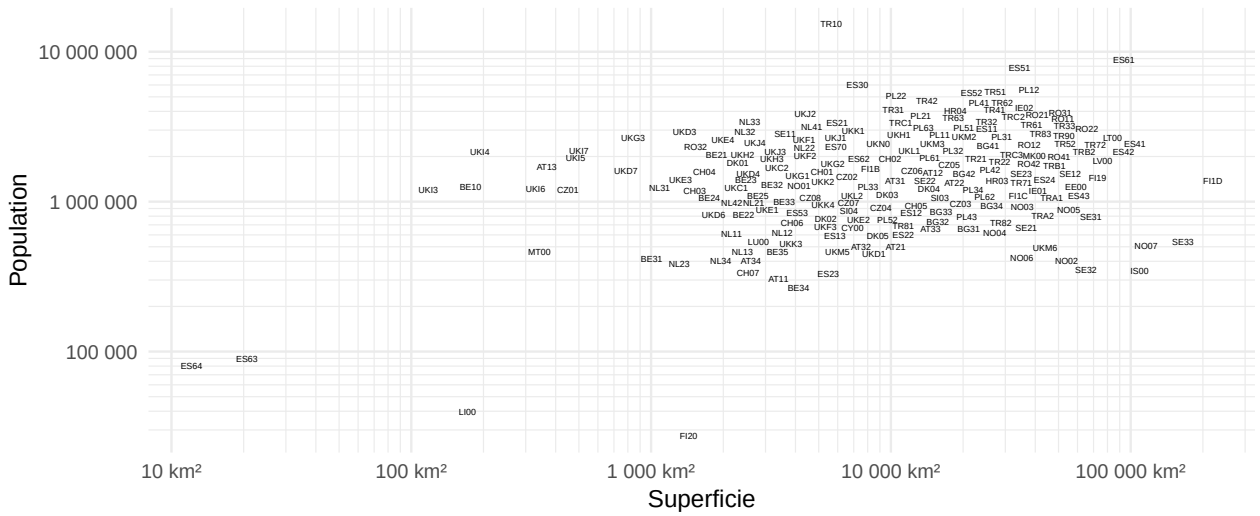
1.2.1 Agglutination

```
library(ggrepel)
NUTS2_year %>%
  filter(!is.na(population), !is.na(superficie), année == 2005) %>%
  ggplot(aes(x=superficie, y=population)) +
  # geom_dl(aes(label=id_anc, group=id_anc), method='smart.grid', size=2) +
  geom_text_repel(aes(label=id_anc), size=2, force=1, segment.colour = NA, box.padding=0) +
  scale_x_continuous(
    labels = function(x){str_c(scales:::number(x), ' km²')}
  ) +
  scale_y_continuous(
    labels = scales:::number
  ) +
  theme_minimal() +
  labs(
    x        = 'Superficie',
    y        = 'Population',
    title    = "Istanbul (Turquie) surpeuplée, Oulu (Finlande) isolée",
    subtitle = "Population et superficie en Europe (2015)",
    caption  = "Source: Eurostat (niveau NUTS 2)"
  )
```



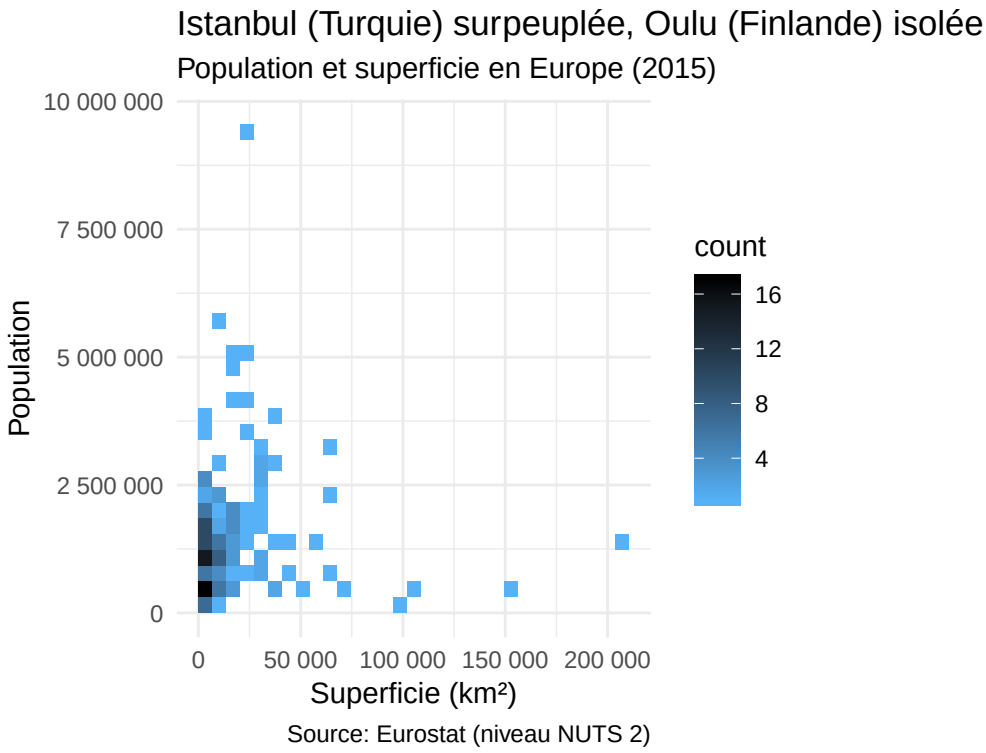
Source: Eurostat (niveau NUTS 2)

```
library(ggrepel)
NUTS2_year %>%
  filter(!is.na(population), !is.na(superficie), année == 2015) %>%
  ggplot(aes(x=superficie, y=population)) +
  # geom_dl(aes(label=id_anc, group=id_anc), method='smart.grid', size=2) +
  geom_text_repel(aes(label=id_anc), size=1.5, force=1, segment.colour = NA, box.padding=0) +
  scale_x_log10(
    minor_breaks = rep(1:10, times=10)*10^rep(1:10, each=10),
    labels = function(x){str_c(scales:::number(x), ' km²')}
  ) +
  scale_y_log10(
    minor_breaks = rep(1:10, times=10)*10^rep(1:10, each=10),
    labels = scales:::number
  ) +
  theme_minimal() +
  labs(
    x        = 'Superficie',
    y        = 'Population',
    title    = "Istanbul (Turquie) surpeuplée, Oulu (Finlande) isolée",
    subtitle = "Population et superficie en Europe (2015)",
    caption  = "Source: Eurostat (niveau NUTS 2)"
  )
```

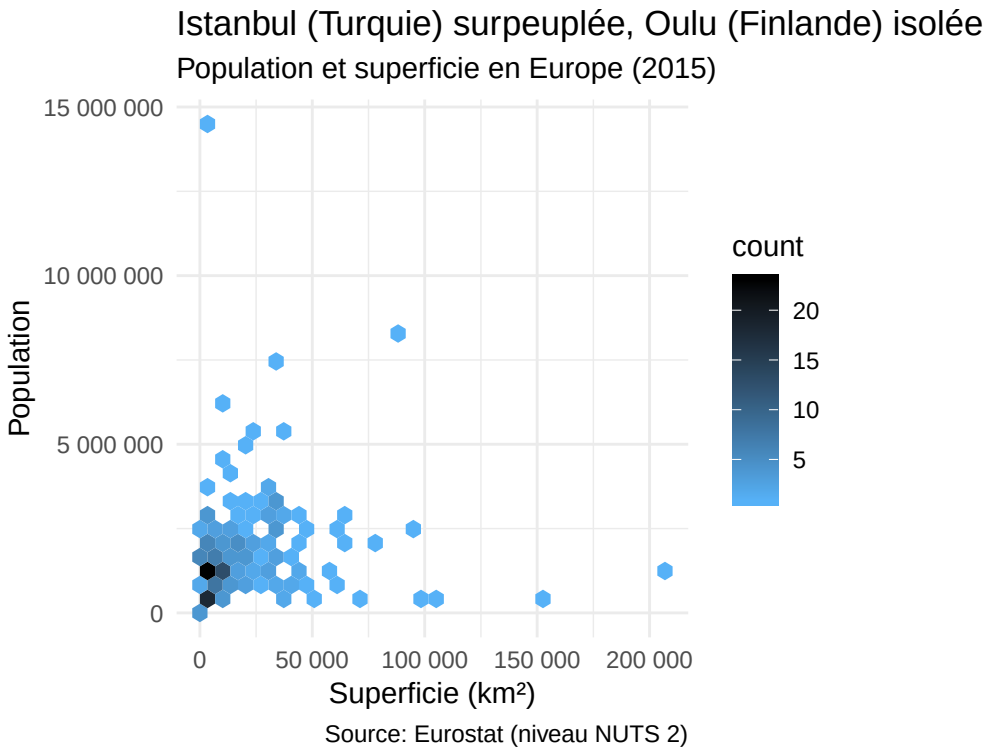


1.2.2 Heatmaps (incl hexbins)

```
NUTS2_year %>%
  filter(!is.na(population), !is.na(superficie), année == 2005) %>%
  ggplot(aes(x=superficie, y=population)) +
  geom_bin2d() +
  scale_x_continuous(
    labels = scales:::number
  ) +
  scale_y_continuous(
    labels = scales:::number
  ) +
  scale_fill_gradient(low='#56b1f7', high='black') +
  theme_minimal() +
  labs(
    x        = 'Superficie (km²)',
    y        = 'Population',
    title    = "Istanbul (Turquie) surpeuplée, Oulu (Finlande) isolée",
    subtitle = "Population et superficie en Europe (2015)",
    caption  = "Source: Eurostat (niveau NUTS 2)"
  ) +
  coord_fixed(ratio=0.025)
```

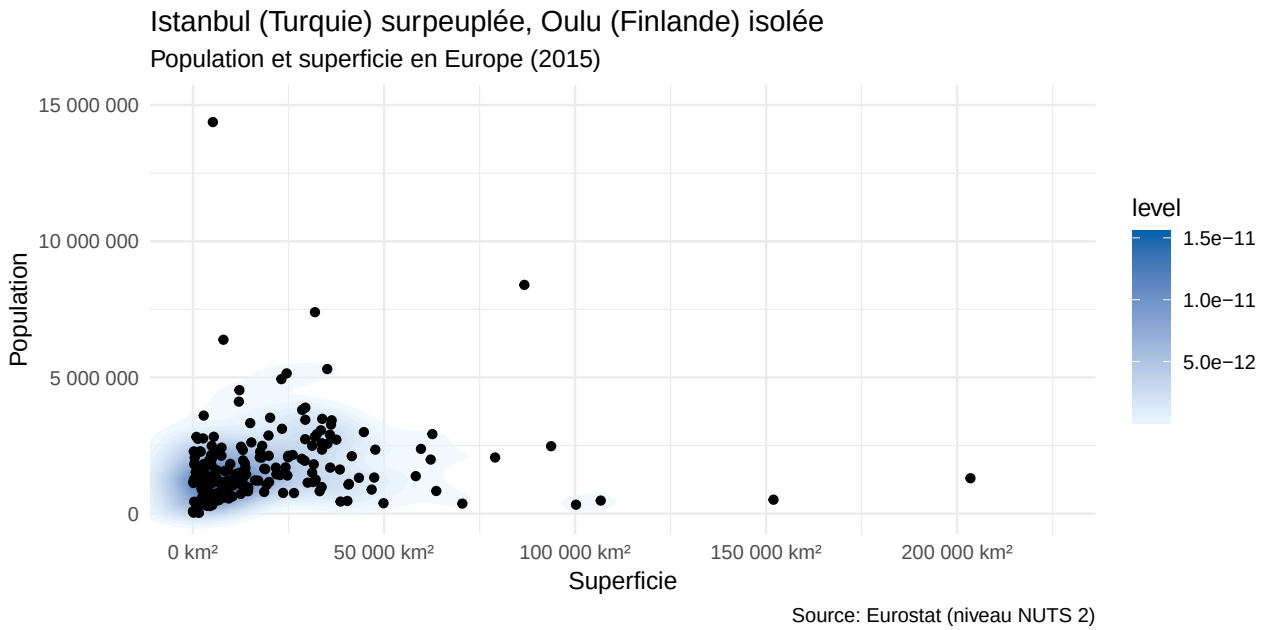



```
NUTS2_year %>%
  filter(!is.na(population), !is.na(superficie), année == 2015) %>%
  ggplot(aes(x=superficie, y=population)) +
  geom_hex() +
  scale_x_continuous(
    labels = scales:::number
  ) +
  scale_y_continuous(
    labels = scales:::number
  ) +
  scale_fill_gradient(low='#56b1f7', high='black') +
  theme_minimal() +
  labs(
    x        = 'Superficie (km²)',
    y        = 'Population',
    title    = "Istanbul (Turquie) surpeuplée, Oulu (Finlande) isolée",
    subtitle = "Population et superficie en Europe (2015)",
    caption  = "Source: Eurostat (niveau NUTS 2)"
  ) +
  coord_fixed(ratio=0.015)
```



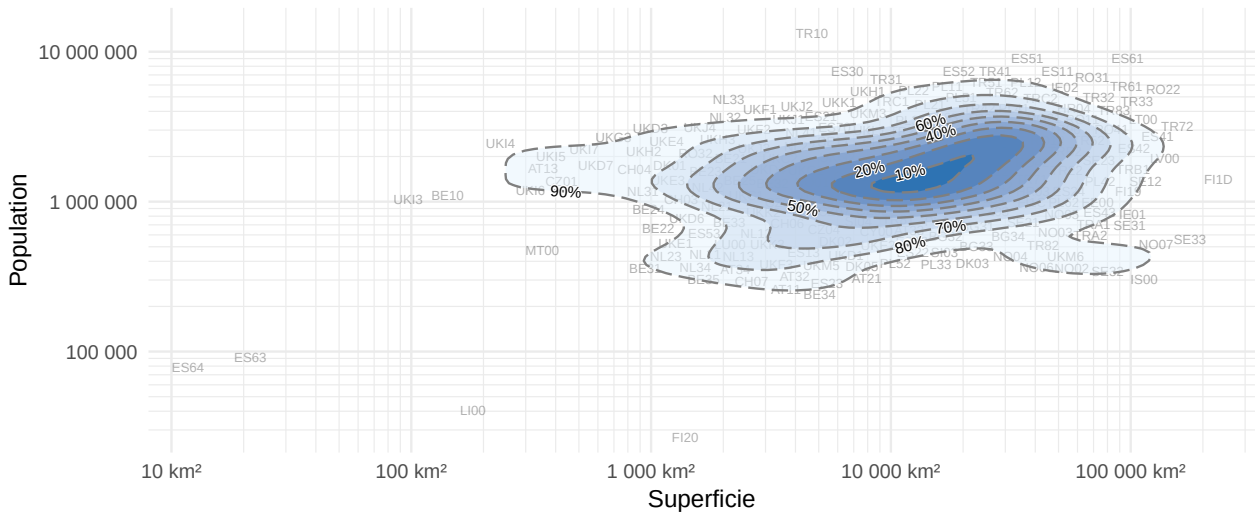
1.2.3 Contours

```
library(metR)
NUTS2_year %>%
  filter(!is.na(population), !is.na(superficie), année == 2015) %>%
  ggplot(aes(x=superficie, y=population)) +
  stat_density_2d(aes(fill = stat(level)), geom = "polygon", bins=40, alpha=0.5) +
  geom_point()+
  # geom_contour(aes(z = ..level..)) +
  # # geom_text(aes(label=id_anc), size=2) +
  scale_fill_gradient(low='#e7f4fe', high='#0864aa') +
  scale_x_continuous(
    labels = function(x){str_c(scales:::number(x), ' km²')},
    limits = c(-70000, 250000)
  ) +
  scale_y_continuous(
    labels = scales:::number,
    limits = c(-600000, 15000000)
  ) +
  theme_minimal() +
  labs(
    x        = 'Superficie',
    y        = 'Population',
    title    = "Istanbul (Turquie) surpeuplée, Oulu (Finlande) isolée",
    subtitle = "Population et superficie en Europe (2015)",
    caption  = "Source: Eurostat (niveau NUTS 2)"
  )+
  coord_cartesian(xlim = c(0, 225000), ylim = c(0, 15000000))
```



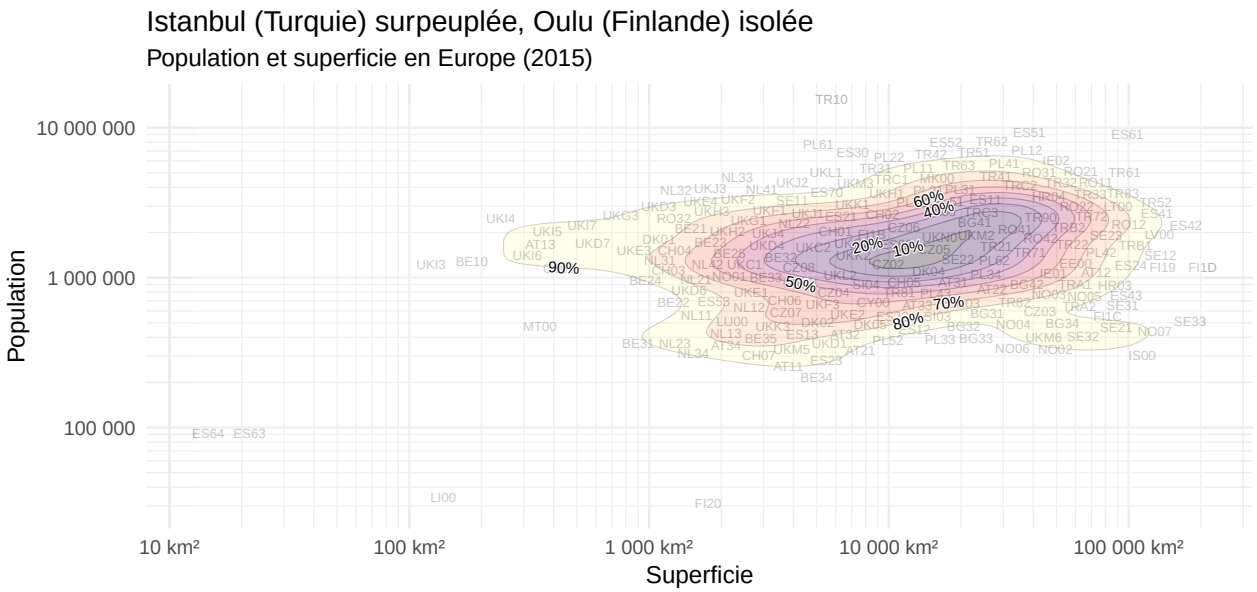

```
library(metR)
library(ggrepel)
NUTS2_year %>%
  filter(!is.na(population), !is.na(superficie), année == 2015) %>%
  ggplot(aes(x=superficie, y=population)) +
  geom_text_repel(aes(label=id_anc), alpha=0.3, size=2, force=1, segment.colour = NA, box.padding=0)+
  geom_contour_fill(aes(fill=stat(level)), stat='density_2d', binwidth=0.1, alpha=0.5) +
  geom_density2d(binwidth=0.1, color='gray50', linetype=5) +
  geom_text_contour(aes(label=str_c(100*(1-stat(level)), '%')), stat='density_2d', binwidth=0.1, stroke
  # stat_density_2d(aes(fill = stat(level)), geom = "polygon", bins=40, alpha=0.5) +
  scale_fill_gradient(low='#e7f4fe', high='#0864aa', guide='none') +
  scale_x_log10(
    minor_breaks = rep(1:10, times=10)*10^rep(1:10, each=10),
    labels = function(x){str_c(scales:::number(x), ' km²')}
  ) +
  scale_y_log10(
    minor_breaks = rep(1:10, times=10)*10^rep(1:10, each=10),
    labels = scales:::number
  ) +
  theme_minimal() +
  labs(
    x        = 'Superficie',
    y        = 'Population',
    title    = "Istanbul (Turquie) surpeuplée, Oulu (Finlande) isolée",
    subtitle = "Population et superficie en Europe (2015)",
    caption  = "Source: Eurostat (niveau NUTS 2)"
  )
```

```
## Warning: Ignoring unknown parameters: breaks, na.fill
```



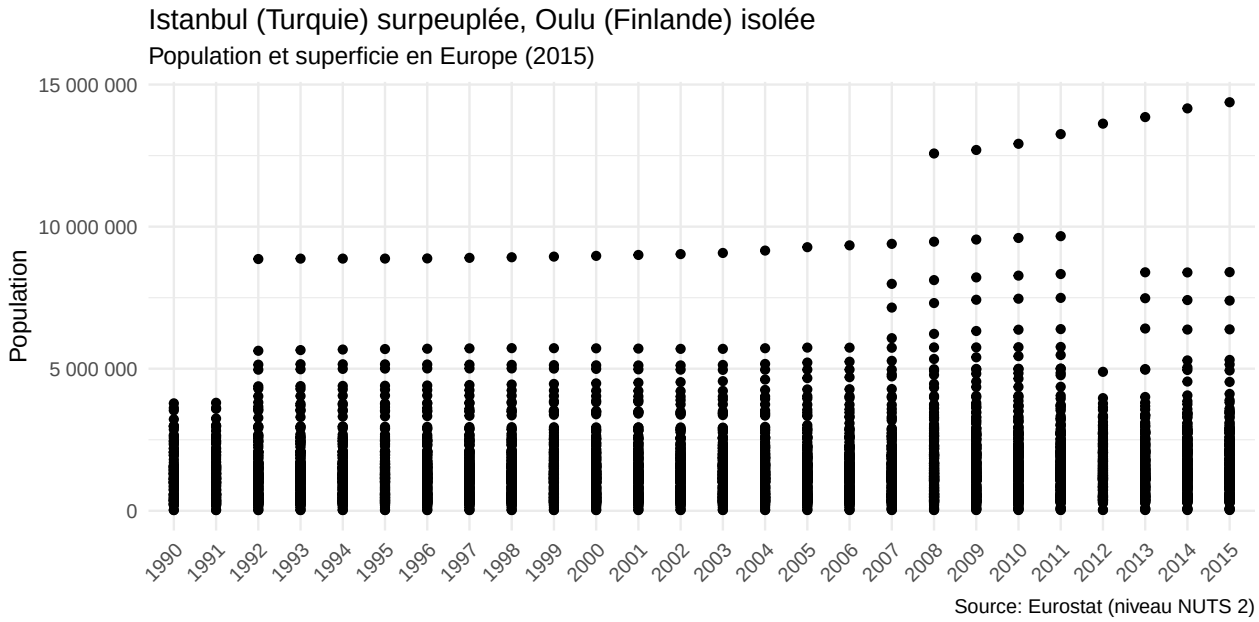

```
library(ggrepel)
library(ggisoband)  # devtools::install_github("clauswilke/ggisoband")

NUTS2_year %>%
  filter(!is.na(population), !is.na(superficie), année == 2015) %>%
  ggplot(aes(x=superficie, y=population)) +
  geom_text_repel(aes(label=id_anc), alpha=0.3, size=2, force=1, segment.colour = NA, box.padding=0)+
  # geom_contour_fill(aes(fill=stat(level)), stat='density_2d', binwidth=0.1, alpha=0.5) +
  geom_density_bands(aes(fill = stat(density) %>% ifelse(.<0.1, NA, .)), alpha=0.3, color = "gray40", s
  geom_text_contour(aes(label=str_c(100*(1-stat(level)), '%')), stat='density_2d', binwidth=0.1, stroke
  # stat_density_2d(aes(fill = stat(level)), geom = "polygon", bins=40, alpha=0.5) +
  # scale_fill_hue() +
  scale_fill_viridis_c(guide = "none", option='A', direction=-1, na.value='transparent') +
  # scale_fill_gradient(low='#e7f4fe', high='#0864aa', guide='none') +
  scale_x_log10(
    minor_breaks = rep(1:10, times=10)*10^rep(1:10, each=10),
    labels = function(x){str_c(scales:::number(x), ' km²')}
  ) +
  scale_y_log10(
    minor_breaks = rep(1:10, times=10)*10^rep(1:10, each=10),
    labels = scales:::number
  ) +
  theme_minimal() +
  labs(
    x        = 'Superficie',
    y        = 'Population',
    title    = "Istanbul (Turquie) surpeuplée, Oulu (Finlande) isolée",
    subtitle = "Population et superficie en Europe (2015)",
    caption  = "Source: Eurostat (niveau NUTS 2)"
  )
```



1.3. variable continue vs. variable discrète

```
NUTS2_year %>%
  filter(!is.na(population), !is.na(superficie)) %>%
  ggplot(aes(x=as.factor(année), y=population)) +
  geom_point() +
  scale_y_continuous(
    labels = scales:::number
  ) +
  scale_fill_gradient(low='#56b1f7', high='black') +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(
    x        = NULL,
    y        = 'Population',
    title    = "Istanbul (Turquie) surpeuplée, Oulu (Finlande) isolée",
    subtitle = "Population et superficie en Europe (2015)",
    caption  = "Source: Eurostat (niveau NUTS 2)"
  )
```

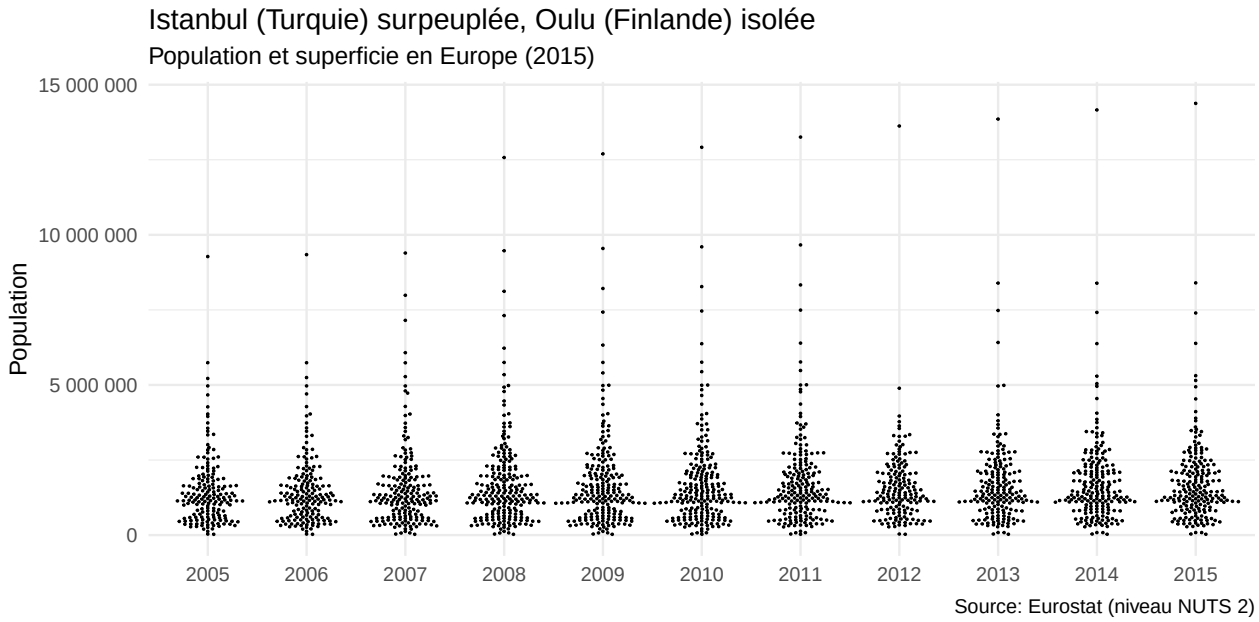


La transparence,

1.3.0 Aglutination

Appelé dans ce contexte, “essaims” ou “beeswarms”.

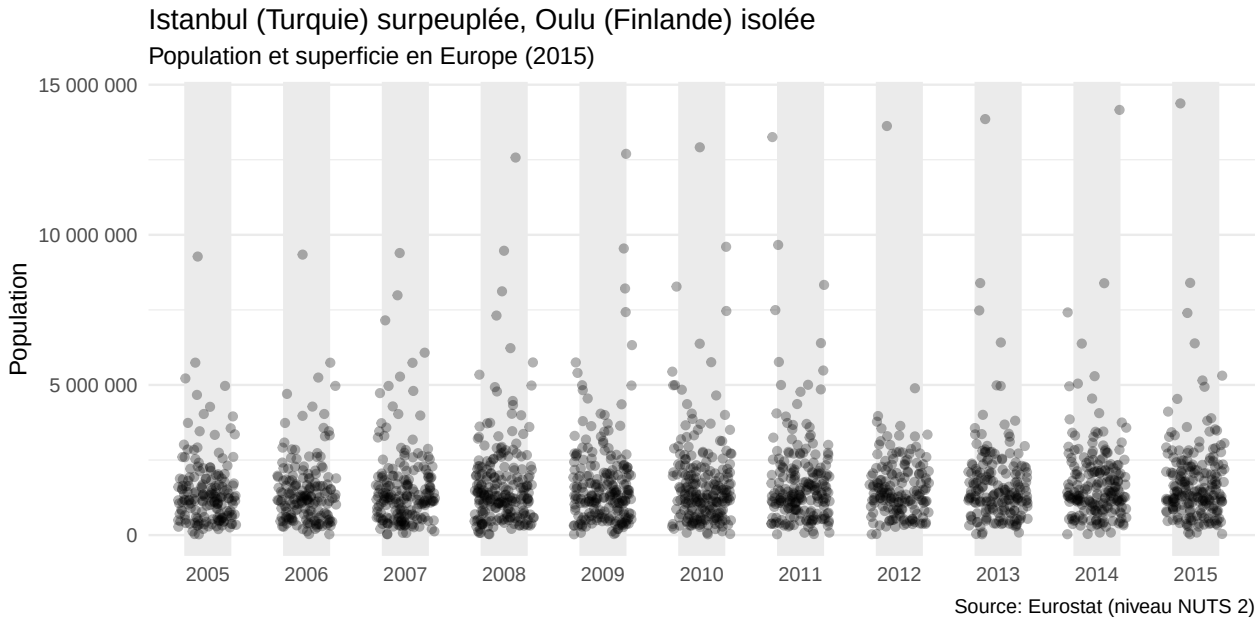
```
library(ggbeeswarm)
NUTS2_year %>%
  filter(année %in% 2005:2015) %>%
  filter(!is.na(population), !is.na(superficie)) %>%
  ggplot(aes(x=as.factor(année), y=round(population,-3))) +
  geom_beeswarm(size=0.1, priority='density', cex=0.55) +
  scale_y_continuous(
    labels = scales:::number
  ) +
  scale_x_discrete()+
  # scale_fill_gradient(low='#56b1f7', high='black') +
  theme_minimal() +
  labs(
    x        = NULL,
    y        = 'Population',
    title    = "Istanbul (Turquie) surpeuplée, Oulu (Finlande) isolée",
    subtitle = "Population et superficie en Europe (2015)",
    caption  = "Source: Eurostat (niveau NUTS 2)"
  )
```



1.3.1 jittering

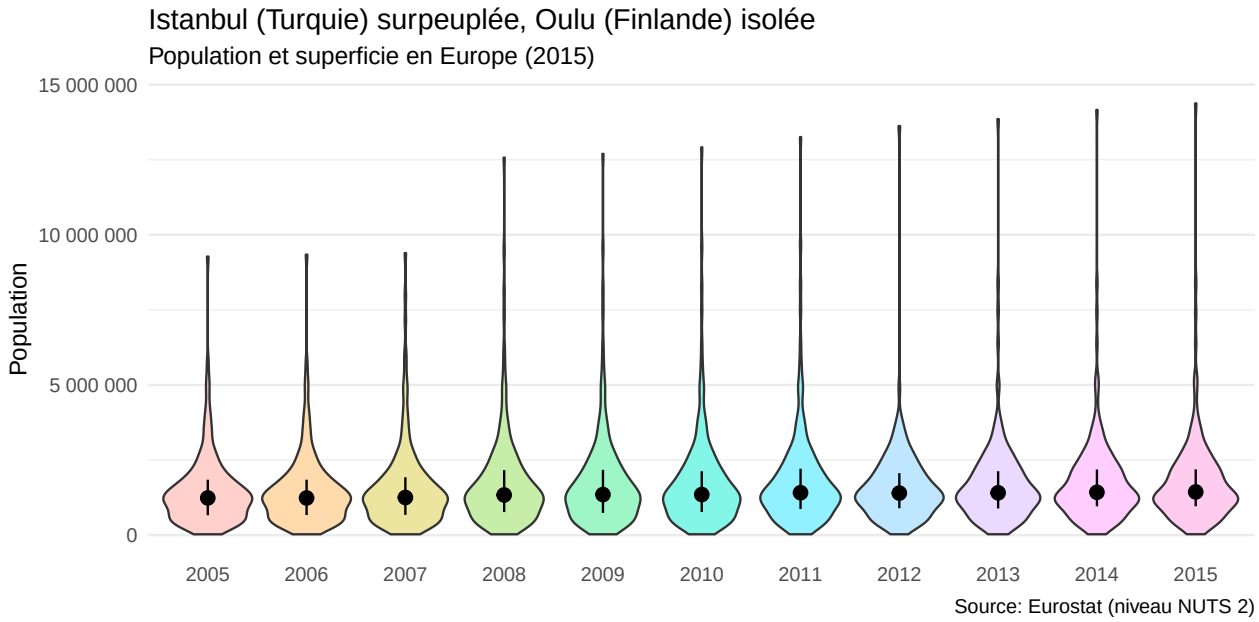
Facile à mettre en œuvre. Parfois très efficace.

```
NUTS2_year %>%
  filter(année %in% 2005:2015) %>%
  filter(!is.na(population), !is.na(superficie)) %>%
  ggplot(aes(x=as.factor(année), y=population)) +
  geom_point(
    position=position_jitter(width = 0.3, height = 0), alpha=0.3) +
  scale_y_continuous(
    labels = scales:::number
  ) +
  scale_x_discrete()+
  # scale_fill_gradient(low='#56b1f7', high='black') +
  theme_minimal() +
  theme(
    panel.grid.major.x = element_line(size=10)
  ) +
  labs(
    x        = NULL,
    y        = 'Population',
    title    = "Istanbul (Turquie) surpeuplée, Oulu (Finlande) isolée",
    subtitle = "Population et superficie en Europe (2015)",
    caption  = "Source: Eurostat (niveau NUTS 2)"
  )
```

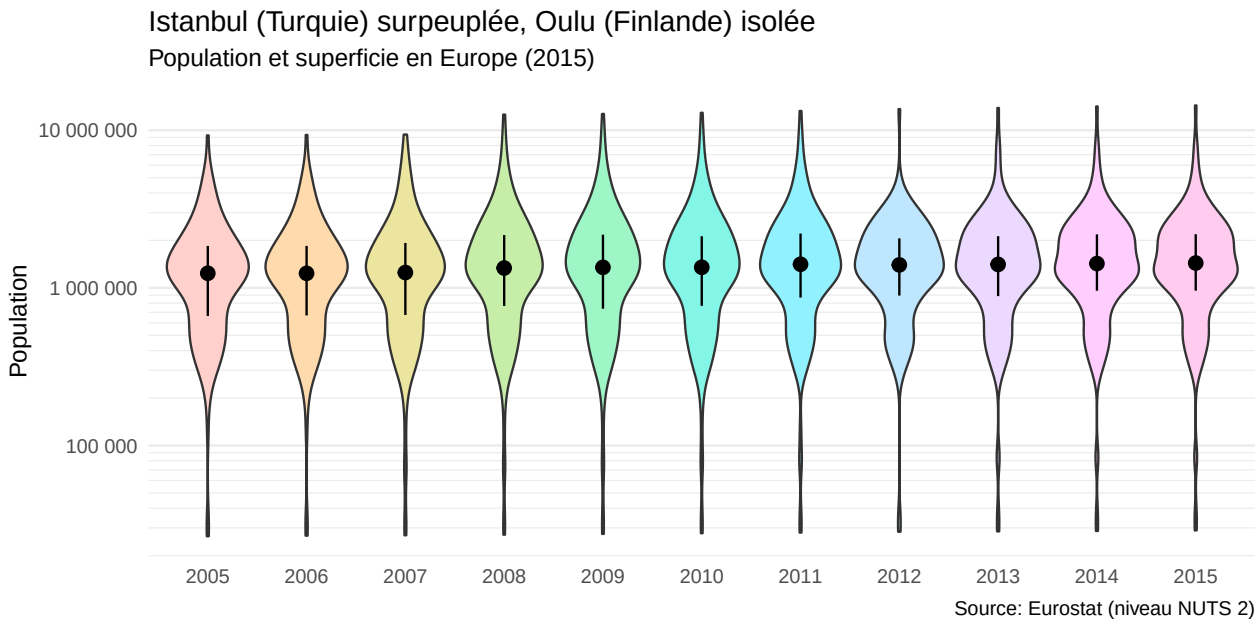


1.3.2 violinplots

```
NUTS2_year %>%
  filter(!is.na(population), !is.na(superficie)) %>%
  filter(année %in% 2005:2015) %>%
  ggplot(aes(x=as.factor(année), y=population)) +
  geom_violin(aes(fill=as.factor(année))) +
  scale_y_continuous(
    labels = scales:::number
  ) +
  geom_pointrange(
    stat = "summary",
    fun.ymin = . %>% quantile(0.25),
    fun.ymax = . %>% quantile(0.75),
    fun.y = median
  ) +
  scale_fill_discrete(guide='none', c=50, l=90) +
  theme_minimal() +
  theme(
    panel.grid.major.x = element_blank()
  ) +
  labs(
    x        = NULL,
    y        = 'Population',
    title    = "Istanbul (Turquie) surpeuplée, Oulu (Finlande) isolée",
    subtitle = "Population et superficie en Europe (2015)",
    caption  = "Source: Eurostat (niveau NUTS 2)"
  )
```



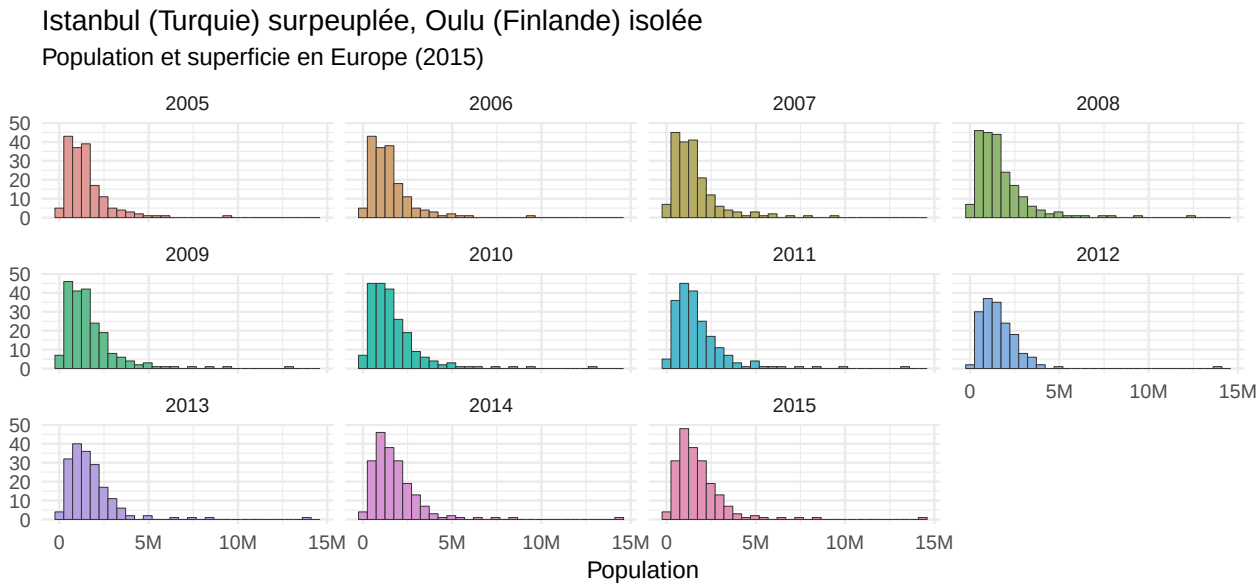

```
NUTS2_year %>%
  filter(!is.na(population), !is.na(superficie)) %>%
  filter(année %in% 2005:20015) %>%
  ggplot(aes(x=as.factor(année), y=population)) +
  geom_violin(aes(fill=as.factor(année))) +
  scale_y_log10(
    minor_breaks = rep(1:10, times=10)*10^rep(1:10, each=10),
    labels = scales:::number
  ) +
  geom_pointrange(
    stat = "summary",
    fun.ymin = . %>% quantile(0.25),
    fun.ymax = . %>% quantile(0.75),
    fun.y = median
  ) +
  scale_fill_discrete(guide='none', c=50, l=90) +
  theme_minimal() +
  theme(
    panel.grid.major.x = element_blank()
  ) +
  labs(
    x        = NULL,
    y        = 'Population',
    title    = "Istanbul (Turquie) surpeuplée, Oulu (Finlande) isolée",
    subtitle = "Population et superficie en Europe (2015)",
    caption  = "Source: Eurostat (niveau NUTS 2)"
  )
```



1.3.3 small multiples

```
NUTS2_year %>%
  filter(!is.na(population), !is.na(superficie)) %>%
  filter(année %in% 2005:2015) %>%
  ggplot(aes(x=population)) +
  geom_histogram(aes(fill=as.factor(année)), color='grey20', size=0.1) +
  theme_minimal() +
  facet_wrap(~année) +
  scale_fill_discrete(guide='none', c=50, l=70) +
  scale_x_continuous(
    # minor_breaks = rep(1:10, times=10)*10^rep(1:10, each=10),
    labels = function(x) ifelse(x==0, x, str_c(scales:::number(x/1000000), 'M'))
  ) +
  labs(
    x        = "Population",
    y        = NULL,
    title    = "Istanbul (Turquie) surpeuplée, Oulu (Finlande) isolée",
    subtitle = "Population et superficie en Europe (2015)",
    caption  = "Source: Eurostat (niveau NUTS 2)"
  )
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#   geom_pointrange(
#     stat = "summary",
#     fun.ymin = . %>% quantile(0.25),
#     fun.ymax = . %>% quantile(0.75),
#     fun.y = median
#   ) +
#   # scale_fill_gradient(low='#56b1f7', high='black') +
#   theme_minimal() +
#   theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
```

1.4. 2 variables discrètes

=> représenter des comptes

1.3.1 Heatmap 1.3.2 Marimekko plot 1.3.3 Disques alignés sur une grille 1.3.4 small multiples

Scénarisation d'un graphique

La scénarisation d'un graphique (en anglais: *story-telling*) est le fait de guider la lecture d'un graphique. Il ne faut pas y voir de la manipulation! Au contraire, il s'agit de faciliter l'apropriation du graphique. Plus le graphique est complexe / original, plus l'aide doit être poussée.

Critique de graphiques