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.

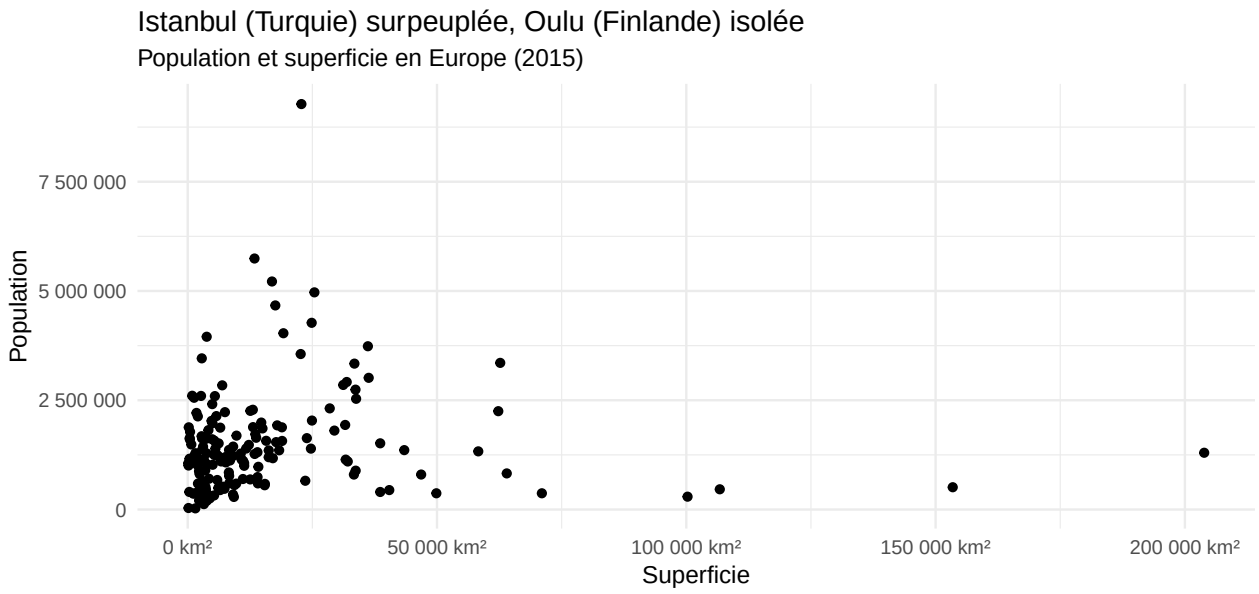
Visualisation de données denses

Trois grands principes: - ruser - g'en'eraliser -

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.

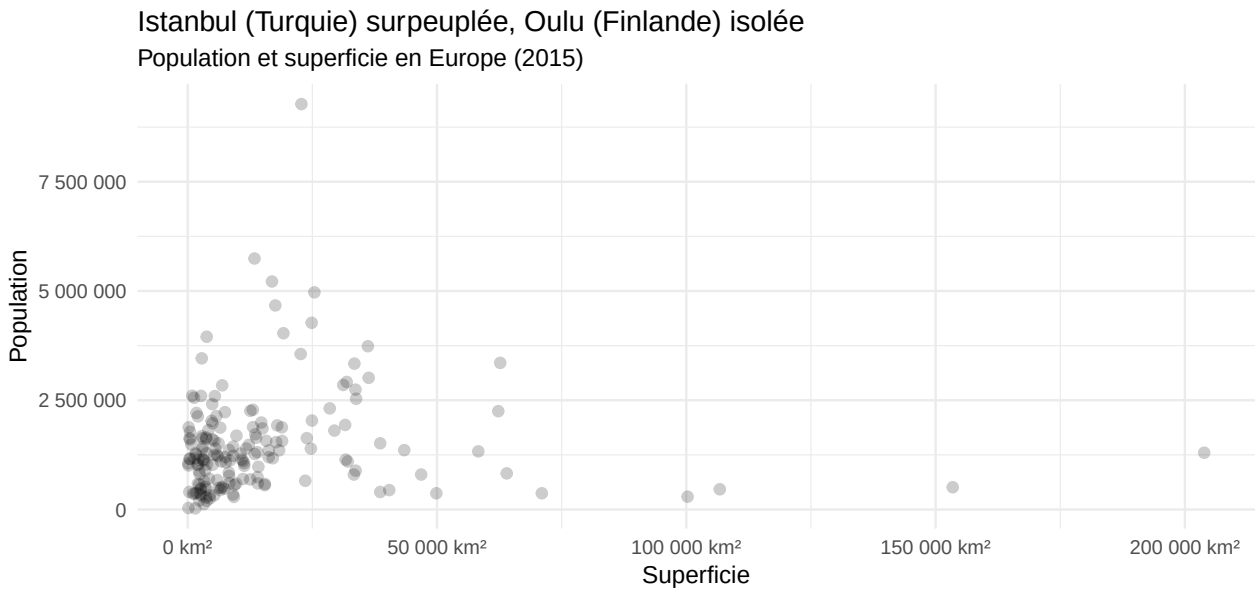
1.2. 2 variables continues

```
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)"
  )
```



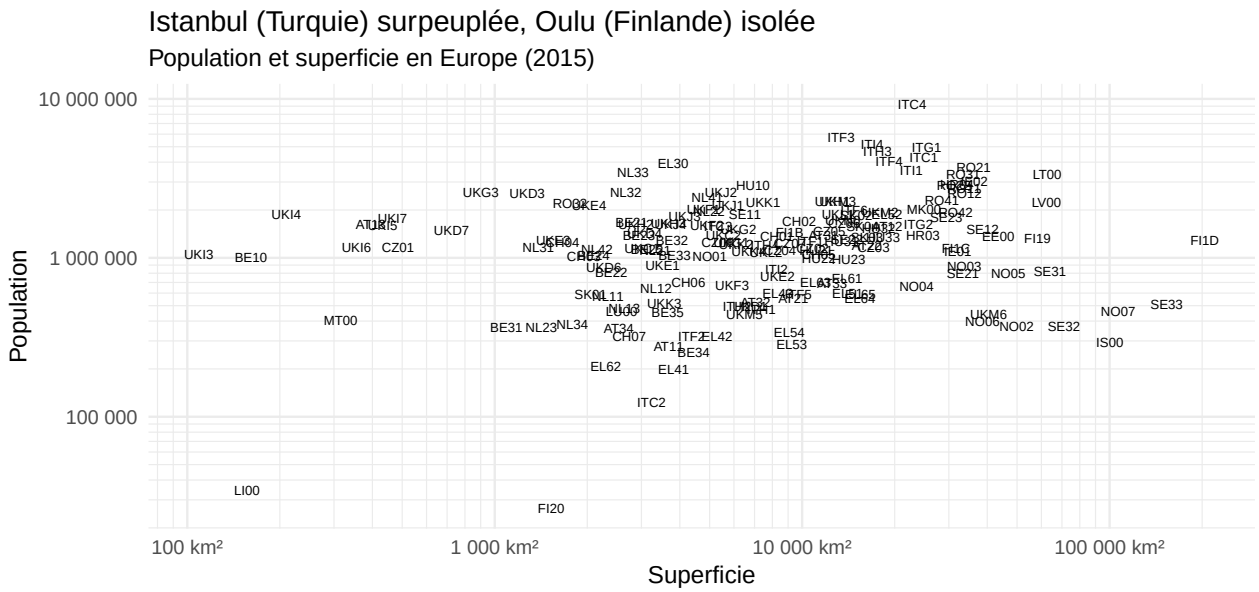
1.2.0 Transparence

```
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)"
  )
```



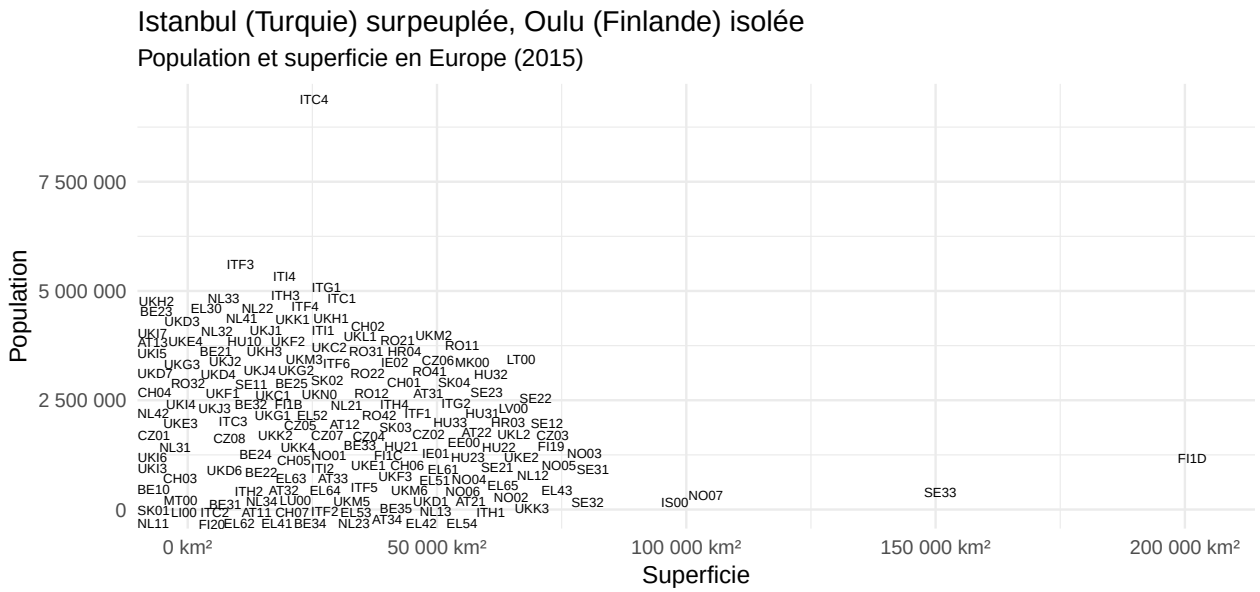
1.2.1 Transformation des axes

```
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)"
  )
```



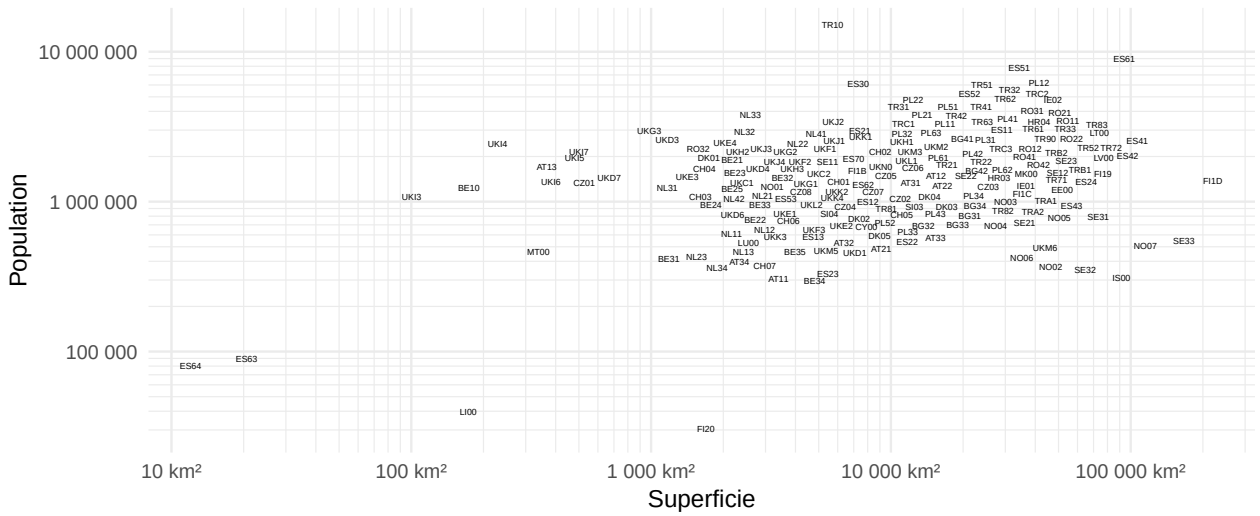
1.2.1 Agglutination

```
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)"
  )
```



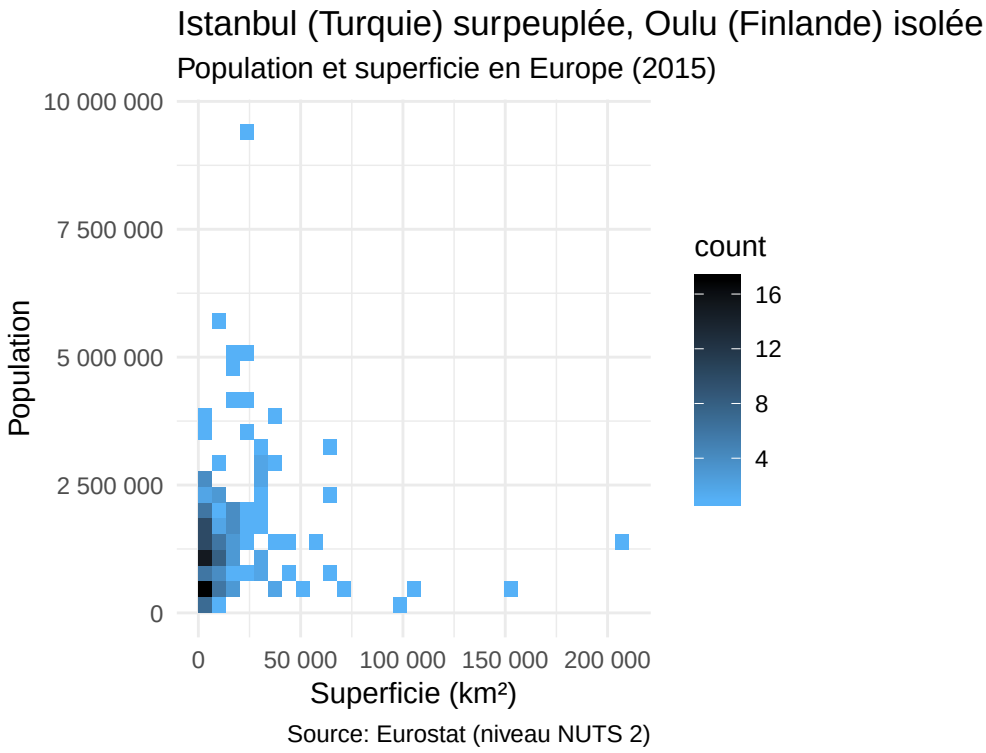
Source: Eurostat (niveau NUTS 2)

```
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)"
  )
```

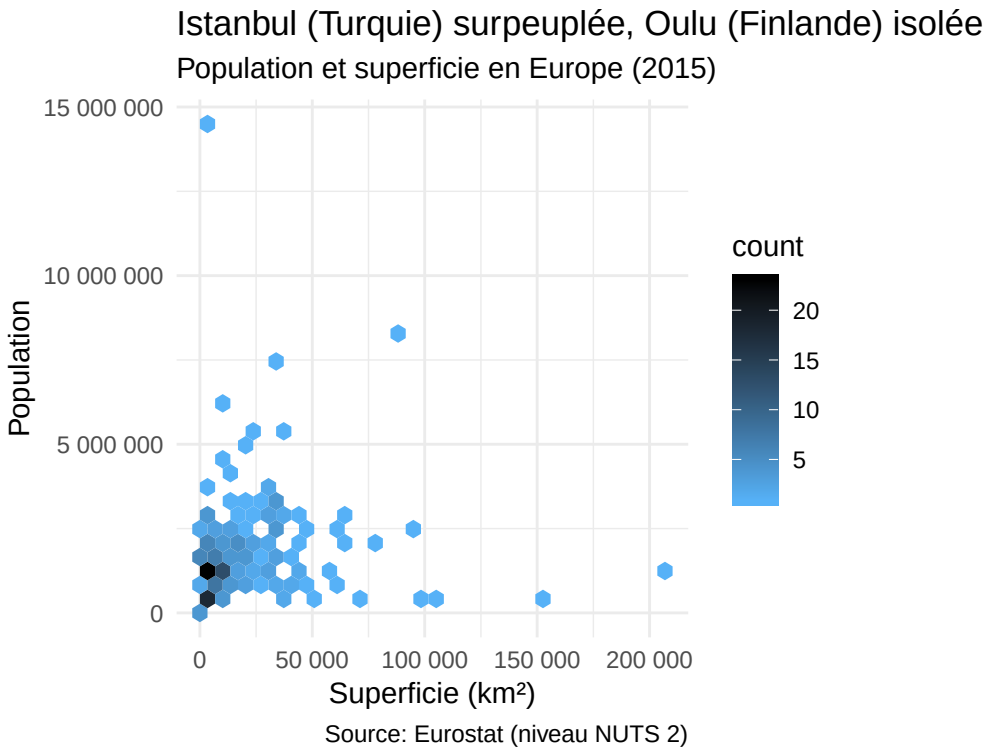


1.2.2 Heatmaps (incl hexbins)

```
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)
```

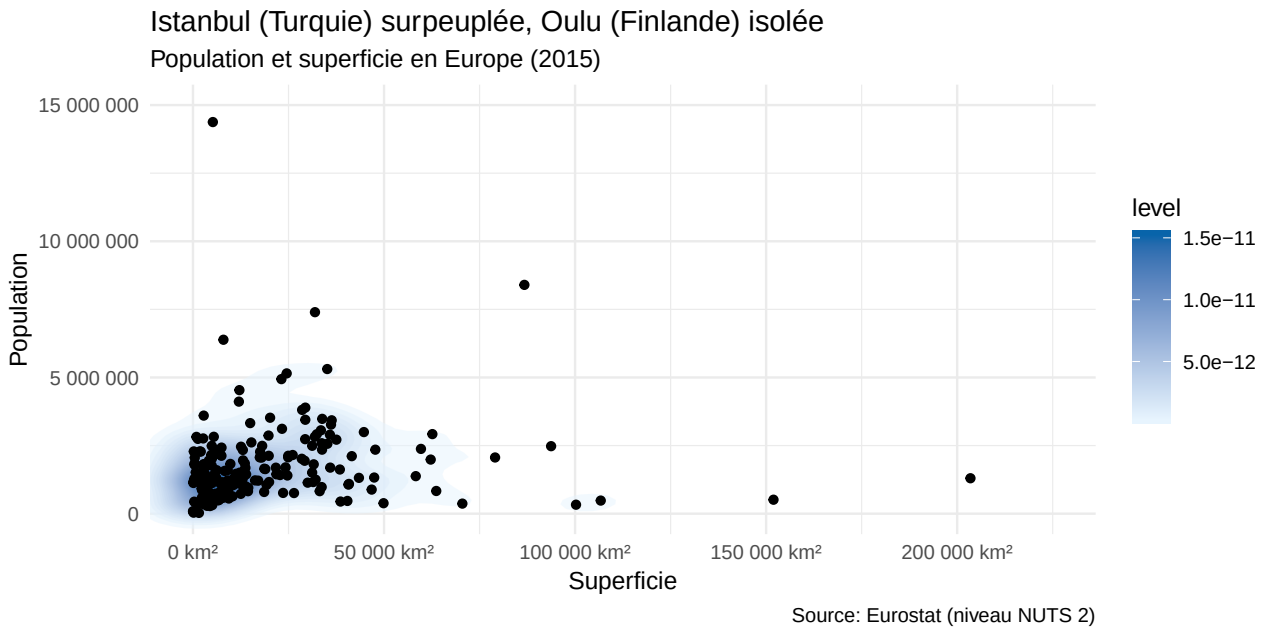



```
NUTS2_year %>%
  filter(!is.na(population), !is.na(superficie), année == 2015) %>%
  ggplot(aes(x=superficie, y=population)) +
  geom_hex() +
  scale_x_continuous(
    labels = scales:::number
  ) +
  scale_y_continuous(
    labels = scales:::number
  ) +
  scale_fill_gradient(low='#56b1f7', high='black') +
  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)
```



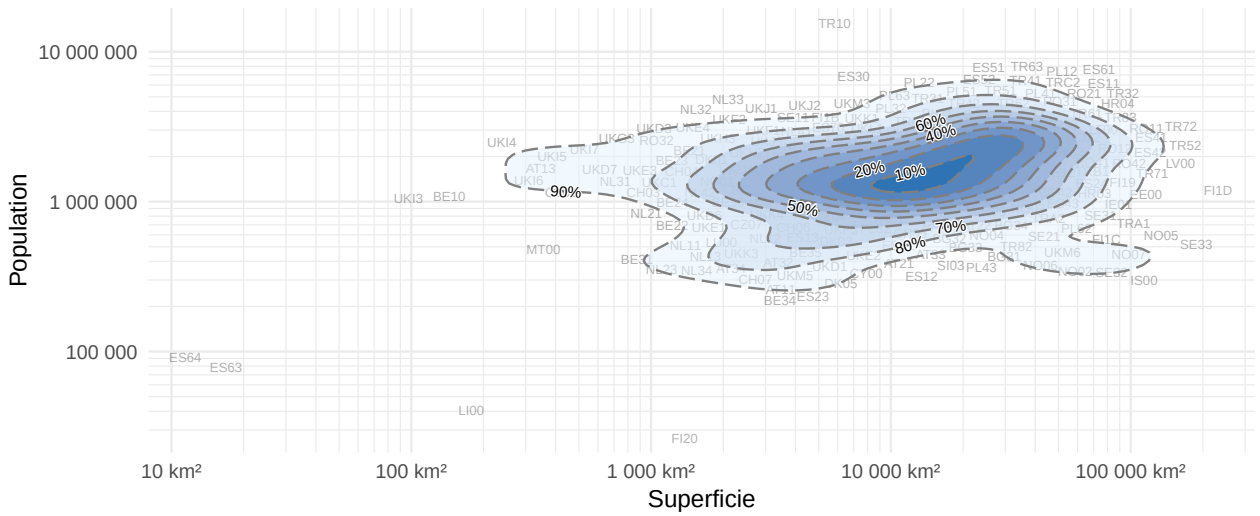
1.2.3 Contours

```
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)"
  )
```

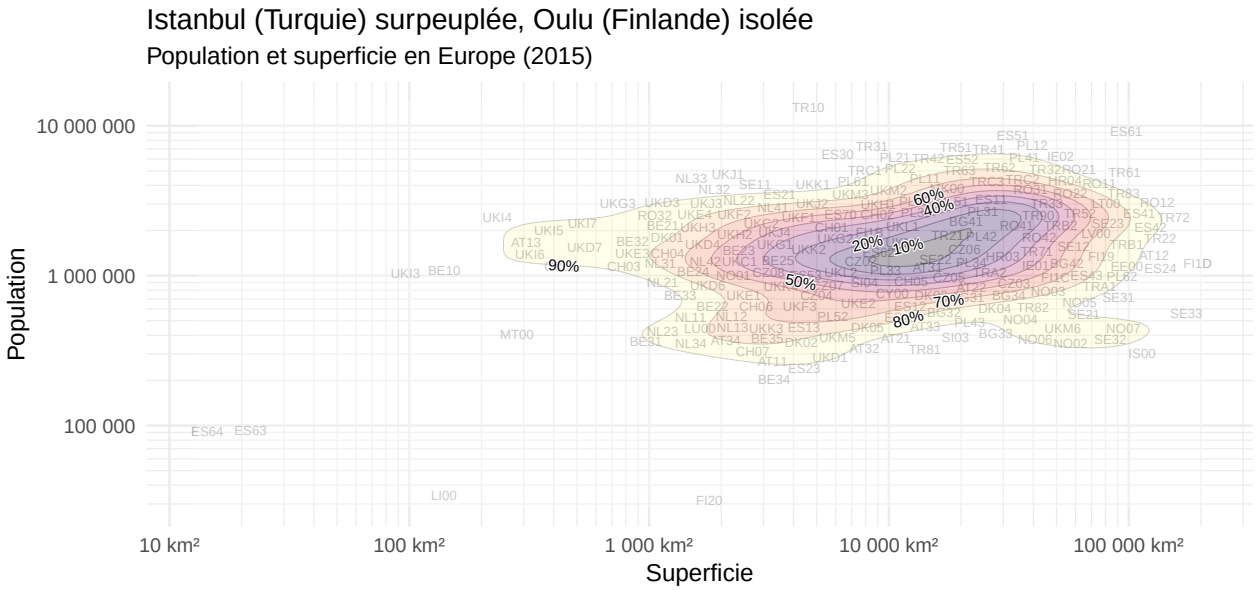
```
## Warning: Ignoring unknown parameters: breaks, na.fill
```



Source: Eurostat (niveau NUTS 2)

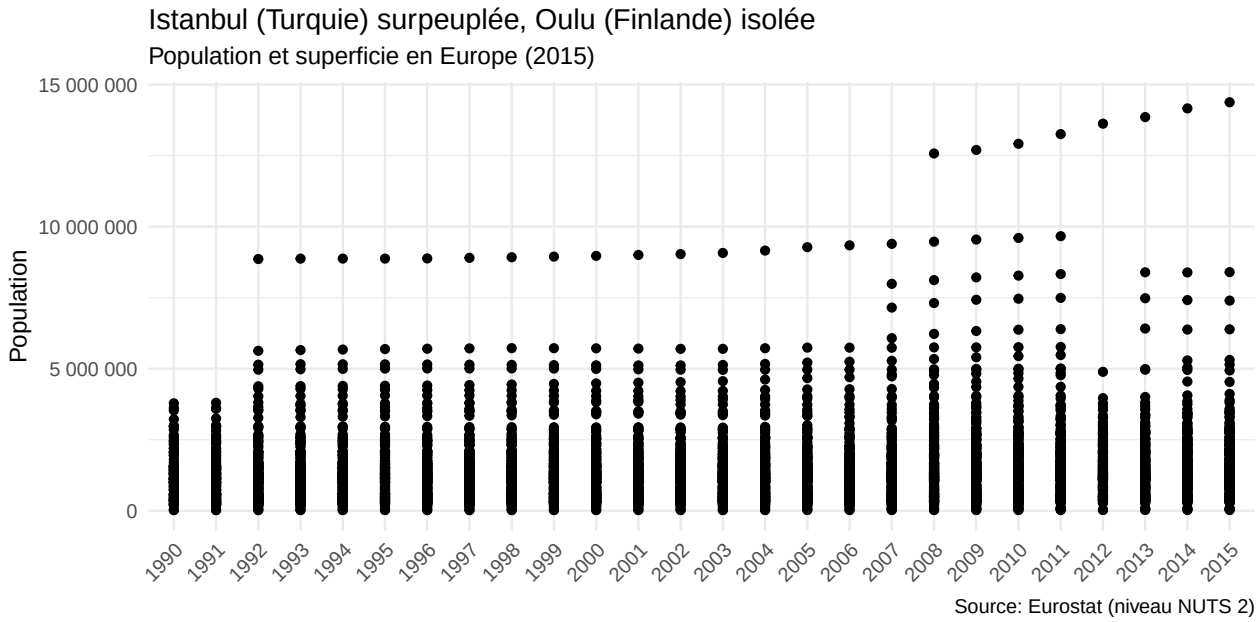
```
library(ggrepel)
library(ggisoband)  # devtools::install_github("clauswilke/ggisoband")

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)"
  )
```

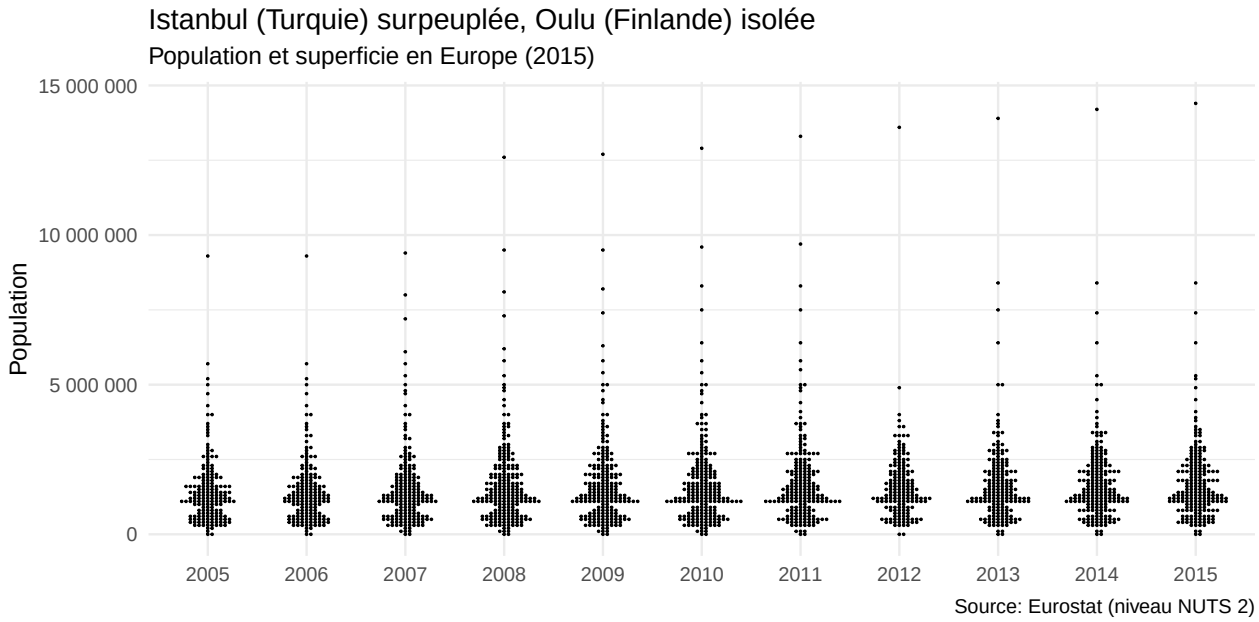


La transparence,

1.3.0 Aglutination

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

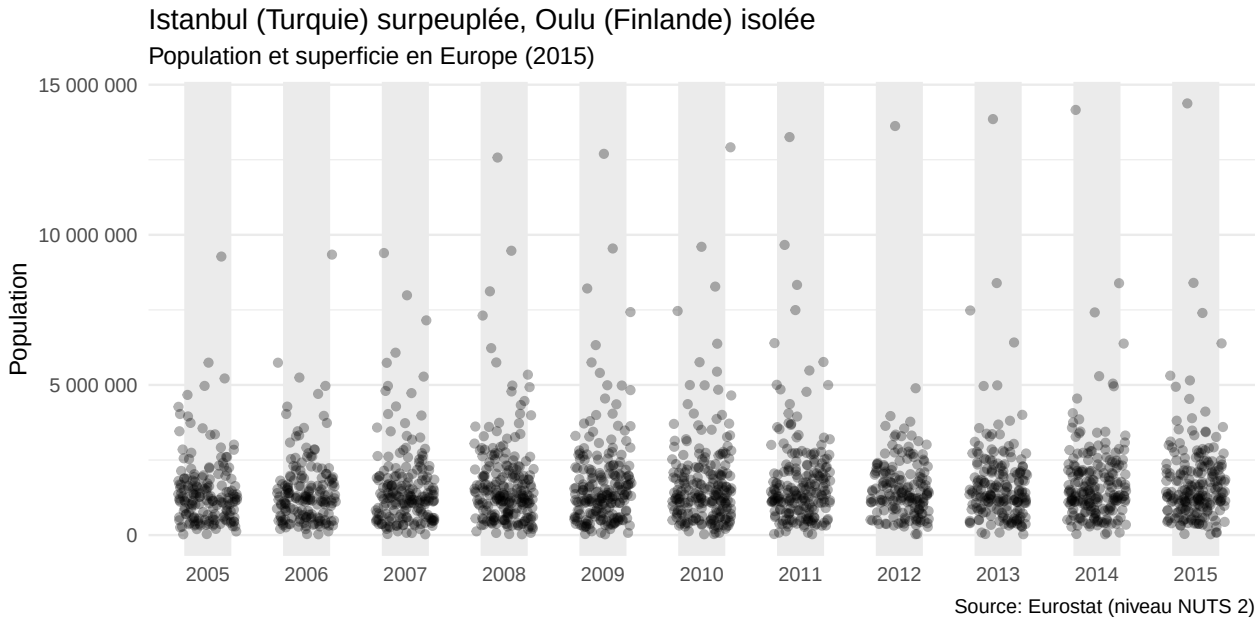
```
library(ggbeeswarm)
NUTS2_year %>%
  filter(année %in% 2005:2015) %>%
  filter(!is.na(population), !is.na(superficie)) %>%
  ggplot(aes(x=as.factor(année), y=round(population,-5))) +
  geom_beeswarm(size=0.1, priority='density', cex=0.4) +
  scale_y_continuous(
    labels = scales:::number
  ) +
  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

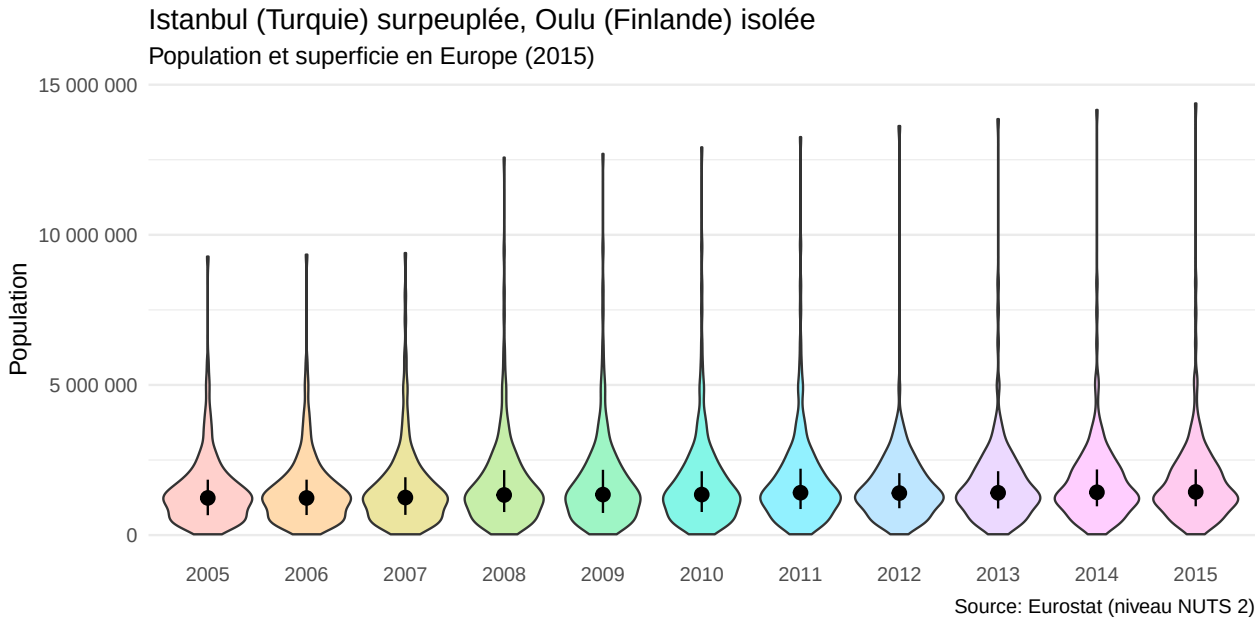
Facile à mettre en œuvre. Parfois très efficace.

```
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)"
  )
```



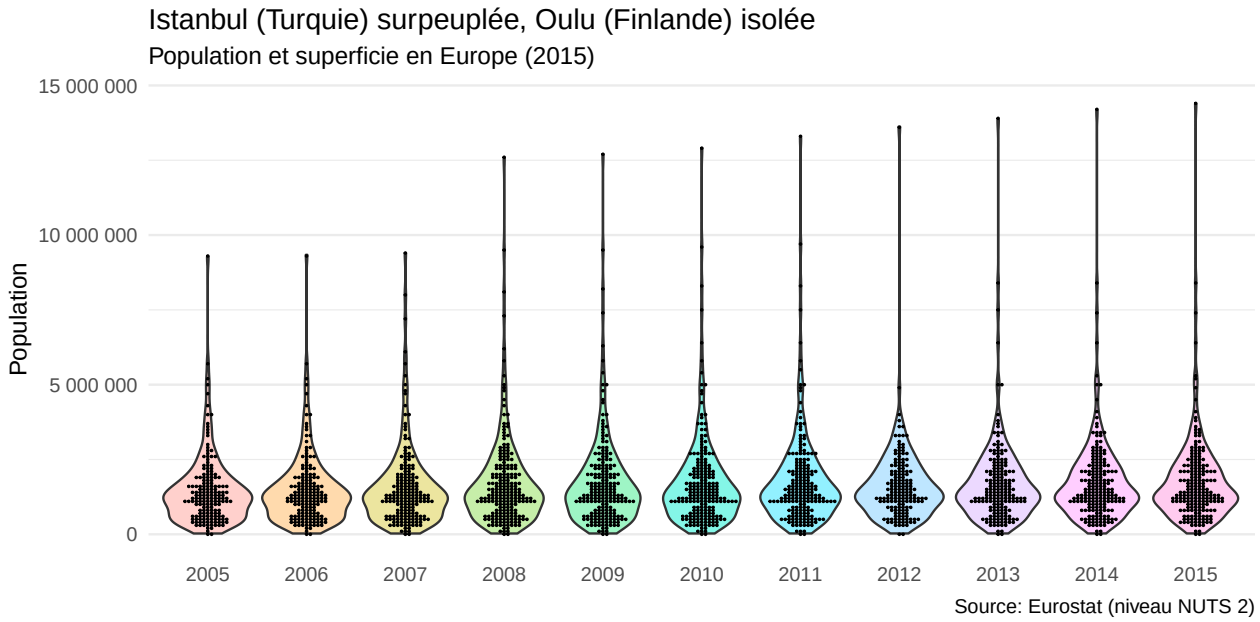
1.3.2 violinplots

```
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)"
  )
```

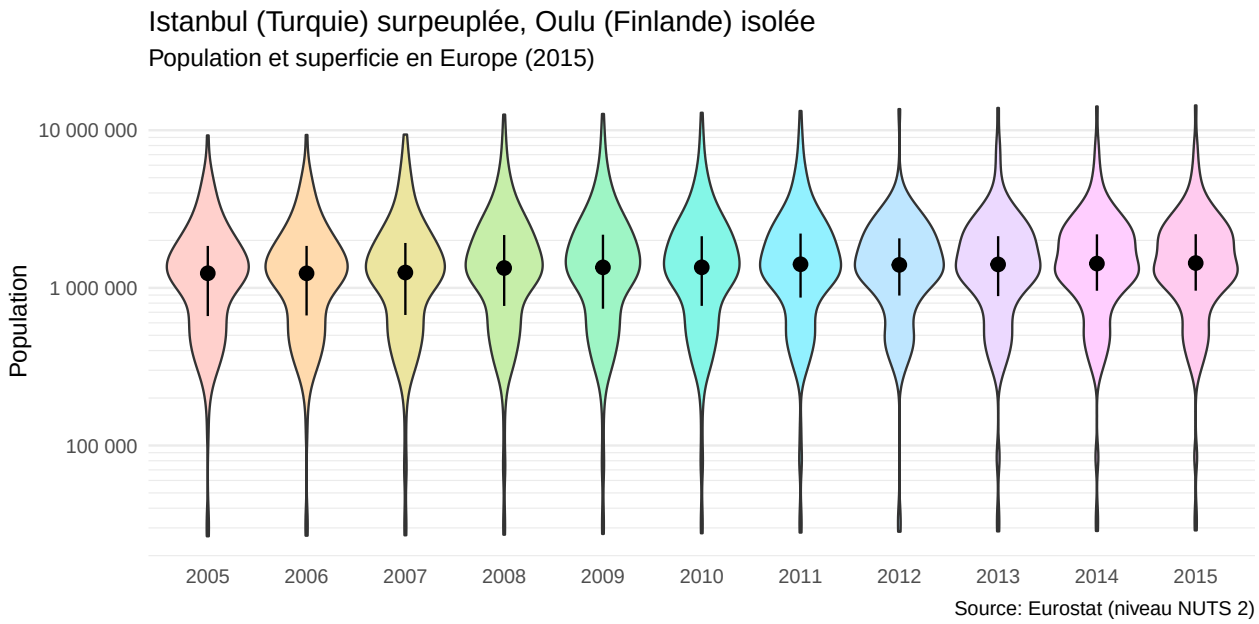


Alternativement, avec le beeswarm à l'intérieur:

```
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_beeswarm(aes(y=round(population,-5)), size=0.1, priority='density', cex=0.35) +
  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)"
  )
```



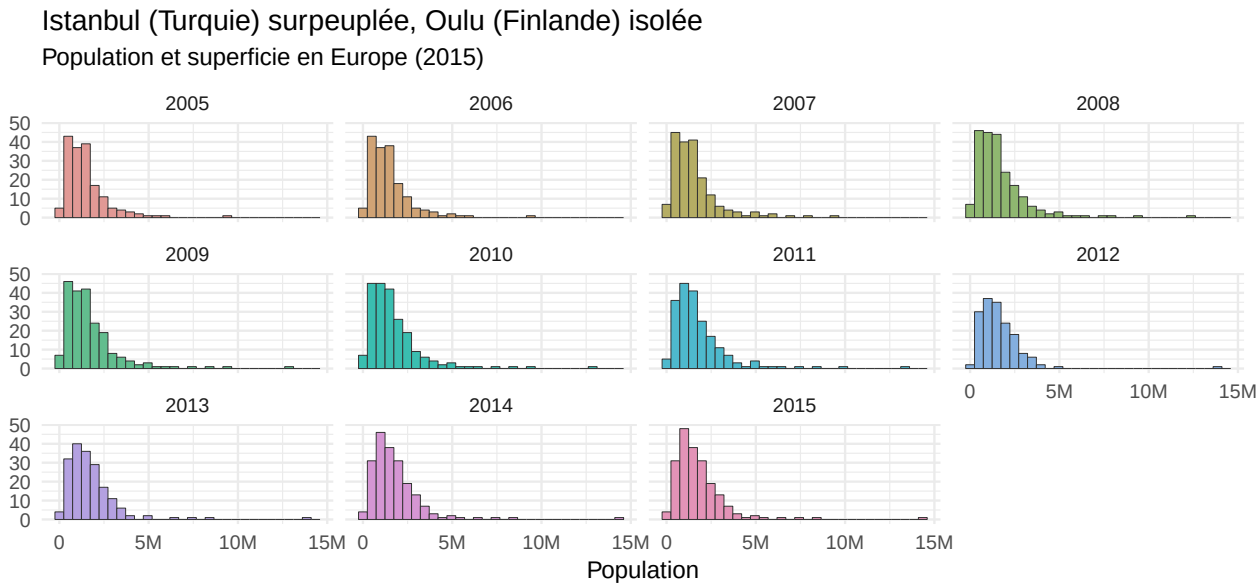

```
NUTS2_year %>%
  filter(!is.na(population), !is.na(superficie)) %>%
  filter(année %in% 2005:20015) %>%
  ggplot(aes(x=as.factor(année), y=population)) +
  geom_violin(aes(fill=as.factor(année))) +
  scale_y_log10(
    minor_breaks = rep(1:10, times=10)*10^rep(1:10, each=10),
    labels = scales:::number
  ) +
  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)"
  )
```

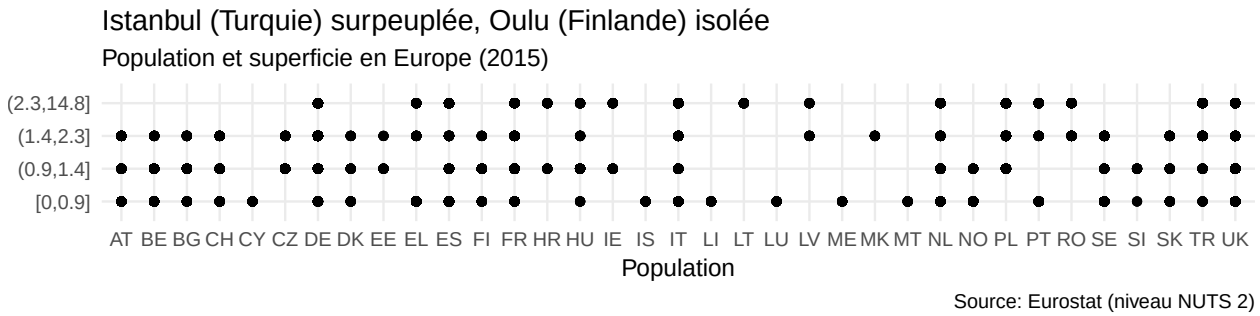
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#   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)) +
```

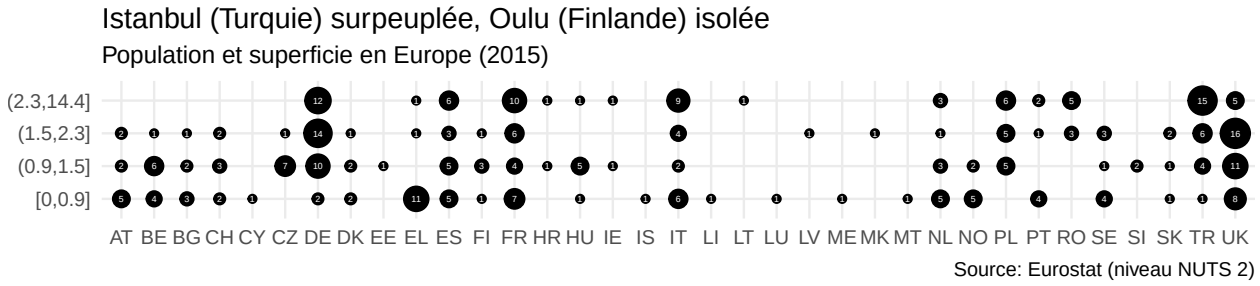
1.4. 2 variables discrètes

```
NUTS2_year %>%
  filter(!is.na(population)) %>%
  mutate(
    population = (population/1000000) %>% round(1) %>% cut(., breaks=quantile(.), include.lowest =TRUE)
    pays       = str_sub(id_anc, end=2)
  ) %>%
  ggplot(aes(y=population, x=pays)) +
  geom_point() +
  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)"
  ) +
  theme_minimal() +
  coord_equal()
```



1.3.0 Représenter des comptes

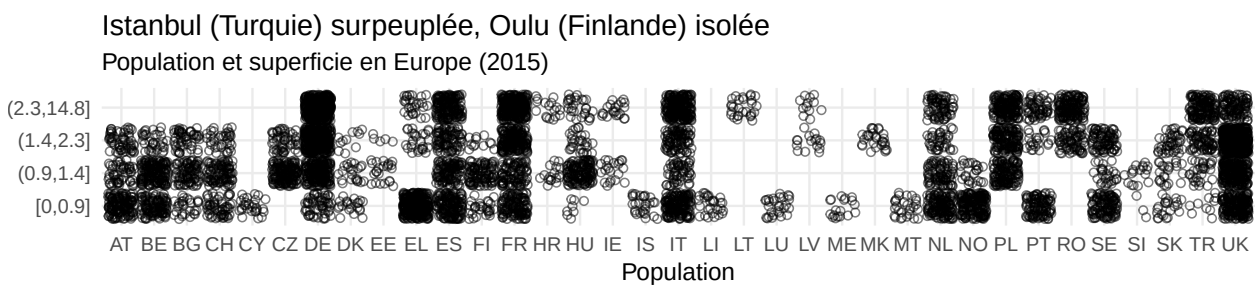
```
NUTS2_year %>%
  filter(!is.na(population), année==2015) %>%
  mutate(
    population = (population/1000000) %>% round(1) %>% cut(., breaks=quantile(.), include.lowest = TRUE)
    pays       = str_sub(id_anc, end=2)
  ) %>%
  group_by(pays, population) %>% summarize(n=n()) %>%
  ggplot(aes(y=population, x=pays)) +
  geom_point(aes(size=n)) +
  geom_text(aes(label=n), col='white', size=1.5) +
  scale_size_area(breaks=c(1, 4, 16), guide='none') +
  labs(
    x        = NULL,
    y        = NULL,
    title    = "Istanbul (Turquie) surpeuplée, Oulu (Finlande) isolée",
    subtitle = "Population et superficie en Europe (2015)",
    caption  = "Source: Eurostat (niveau NUTS 2)"
  ) +
  theme_minimal() +
  coord_equal()
```



1.3.1 Jittering

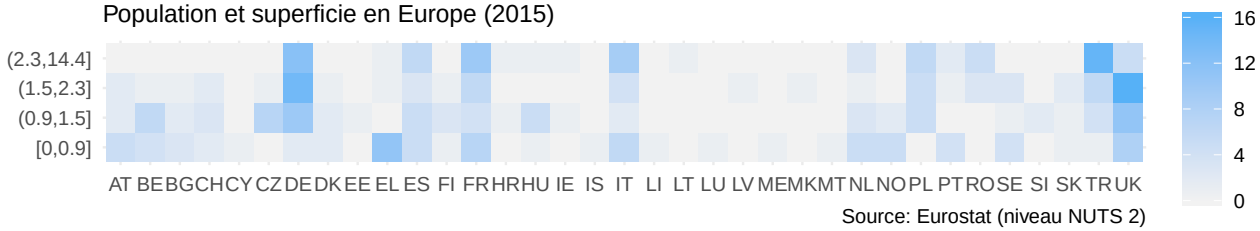
```
NUTS2_year %>%
  filter(!is.na(population)) %>%
  mutate(
    population = (population/1000000) %>% round(1) %>% cut(., breaks=quantile(.), include.lowest =TRUE)
    pays       = str_sub(id_anc, end=2)
  ) %>%
  ggplot(aes(y=population, x=pays)) +
  geom_point(position=position_jitter(), alpha=0.5, shape=21) + # ou geom_jitter
  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)"
  ) +
  theme_minimal() +
  coord_equal()
```



1.3.1 Heatmap

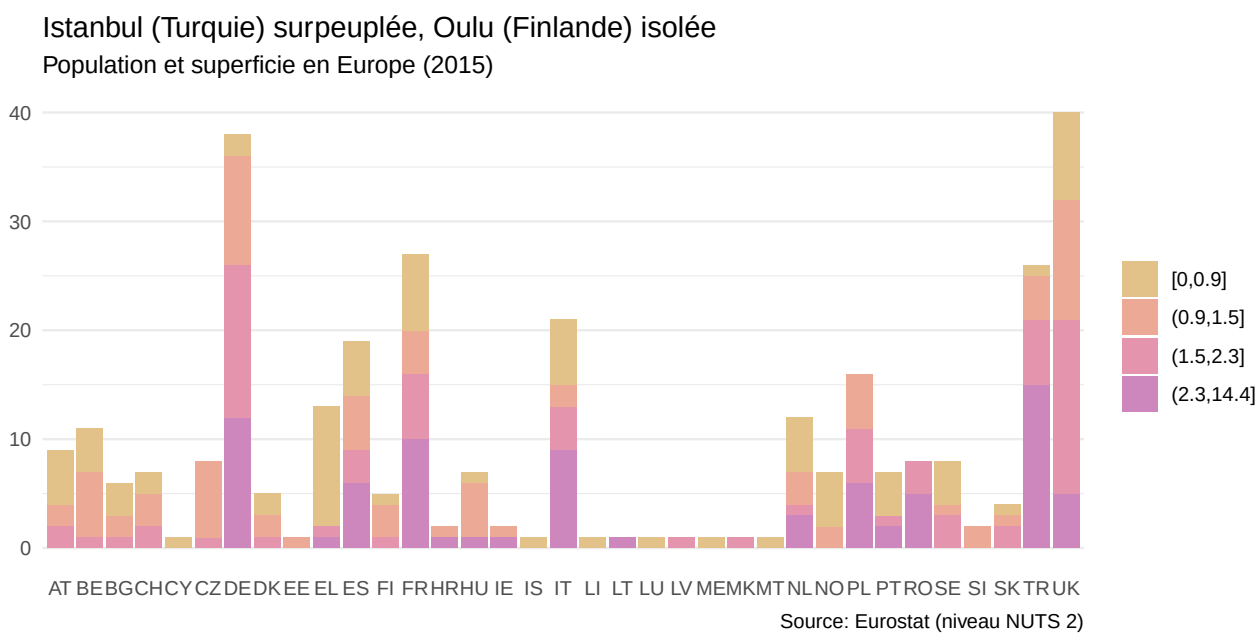
```
NUTS2_year %>%
  filter(!is.na(population), année==2015) %>%
  mutate(
    population = (population/1000000) %>% round(1) %>% cut(., breaks=quantile(.), include.lowest =TRUE),
    pays       = str_sub(id_anc, end=2)
  ) %>%
  group_by(pays, population) %>% summarize(n=n()) %>%
  complete(pays, population, fill=list(n=0)) %>%
  ggplot(aes(y=population, x=pays)) +
  geom_tile(aes(fill=n)) +
  scale_fill_continuous(low='grey95')+
  labs(
    fill     = NULL,
    x        = NULL,
    y        = NULL,
    title    = "Istanbul (Turquie) surpeuplée, Oulu (Finlande) isolée",
    subtitle = "Population et superficie en Europe (2015)",
    caption  = "Source: Eurostat (niveau NUTS 2)"
  ) +
  theme_minimal() +
  coord_equal()
```



1.3.2 Barres empilées, barres empilées standardisées, Marimekko plot

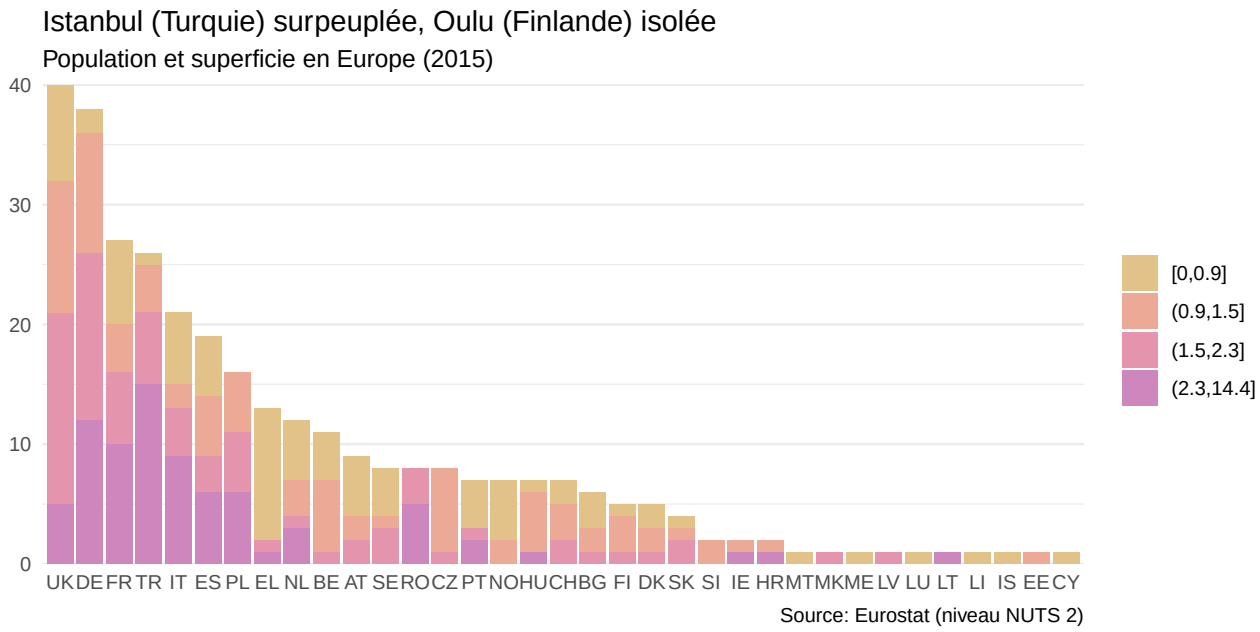
Pas d'échelle logarithmique avec les barres!!!!!!!!

```
NUTS2_year %>%
  filter(!is.na(population), année==2015) %>%
  mutate(
    population = (population/1000000) %>% round(1) %>% cut(., breaks=quantile(.), include.lowest =TRUE)
    pays       = str_sub(id_anc, end=2)
  ) %>%
  arrange() %>%
  ggplot(aes(x=pays, group=population, fill=population)) +
  geom_bar(stat='count', position='stack') +
  scale_fill_hue(h=c(60,-40), l=c(80,75,70,65), c=50)+
  scale_y_continuous(limits = c(0,NA))+
  labs(
    fill     = NULL,
    x        = NULL,
    y        = NULL,
    title    = "Istanbul (Turquie) surpeuplée, Oulu (Finlande) isolée",
    subtitle = "Population et superficie en Europe (2015)",
    caption  = "Source: Eurostat (niveau NUTS 2)"
  ) +
  theme_minimal()+
  theme(
    panel.grid.major.x = element_blank()
  )
```




```
NUTS2_year %>%
  filter(!is.na(population), année==2015) %>%
  mutate(
    population = (population/1000000) %>% round(1) %>% cut(., breaks=quantile(.), include.lowest =TRUE)
    pays       = str_sub(id_anc, end=2)
  ) %>%
  group_by(population, pays) %>% summarise(n=n()) %>% ungroup %>%
  mutate(pays = pays %>% fct_reorder(n, sum) %>% fct_rev) %>%
  ggplot(aes(x=pays, group=population, fill=population)) +
  geom_bar(aes(y=n), stat='identity', position='stack') + # position = position_stack(reverse = TRUE)
```

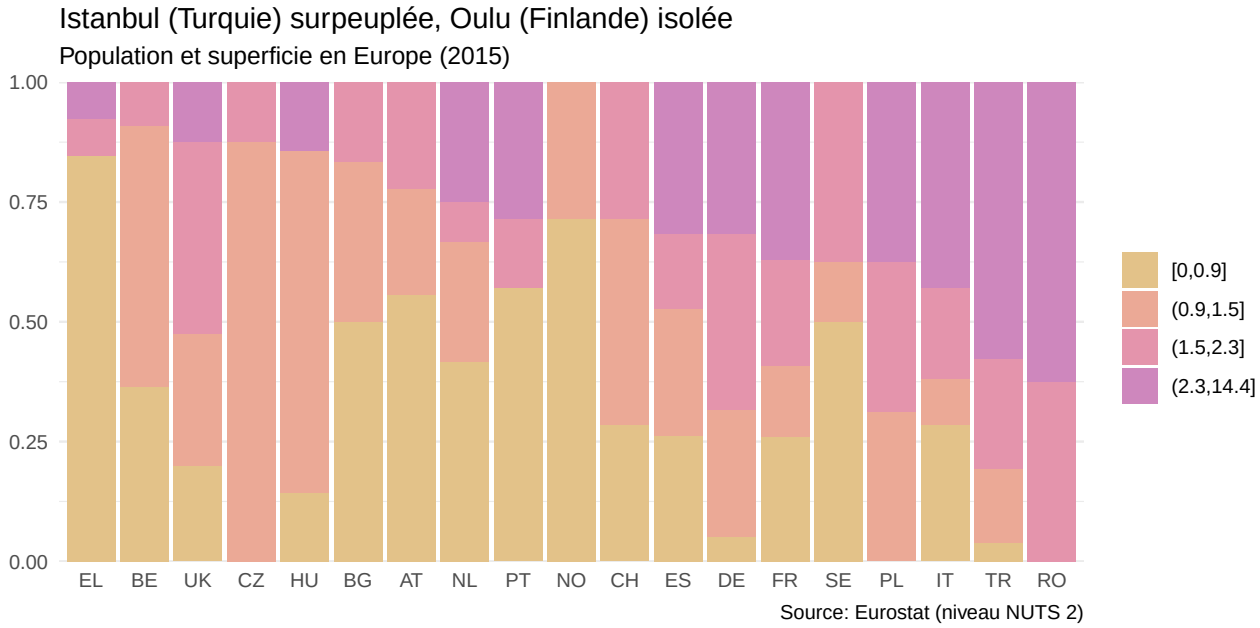
```
  scale_y_continuous(expand = c(0, 0))+
  scale_fill_hue(h=c(60,-40), l=c(80,75,70,65), c=50)+
  labs(
    fill     = NULL,
    x        = NULL,
    y        = NULL,
    title    = "Istanbul (Turquie) surpeuplée, Oulu (Finlande) isolée",
    subtitle = "Population et superficie en Europe (2015)",
    caption  = "Source: Eurostat (niveau NUTS 2)"
  ) +
  theme_minimal()+
  theme(
    panel.grid.major.x = element_blank()
  )
```



On peut normaliser:

```
NUTS2_year %>%
  filter(!is.na(population), année==2015) %>%
  mutate(
    population = (population/1000000) %>% round(1) %>% cut(., breaks=quantile(.), include.lowest =TRUE)
    pays       = str_sub(id_anc, end=2)
  ) %>%
  group_by(population, pays) %>% summarise(n=n()) %>% ungroup %>%
  group_by(pays) %>% mutate(n_pays=sum(n)) %>% ungroup %>%
  filter(n_pays > 5) %>%
  mutate(pays = pays %>% fct_reorder2(population, n/n_pays) %>% fct_rev) %>%
  ggplot(aes(x=pays, group=population, fill=population)) +
  geom_bar(aes(y=n), stat='identity', position=position_fill(reverse = TRUE)) + # position = position_s
  scale_y_continuous(expand = c(0, 0))+
  scale_fill_hue(h=c(60,-40), l=c(80,75,70,65), c=50)+
  labs(
    fill     = NULL,
    x        = NULL,
```

```
    y        = NULL,
    title    = "Istanbul (Turquie) surpeuplée, Oulu (Finlande) isolée",
    subtitle = "Population et superficie en Europe (2015)",
    caption  = "Source: Eurostat (niveau NUTS 2)"
  ) +
  theme_minimal()+
  theme(
    panel.grid.major.x = element_blank()
  )
```



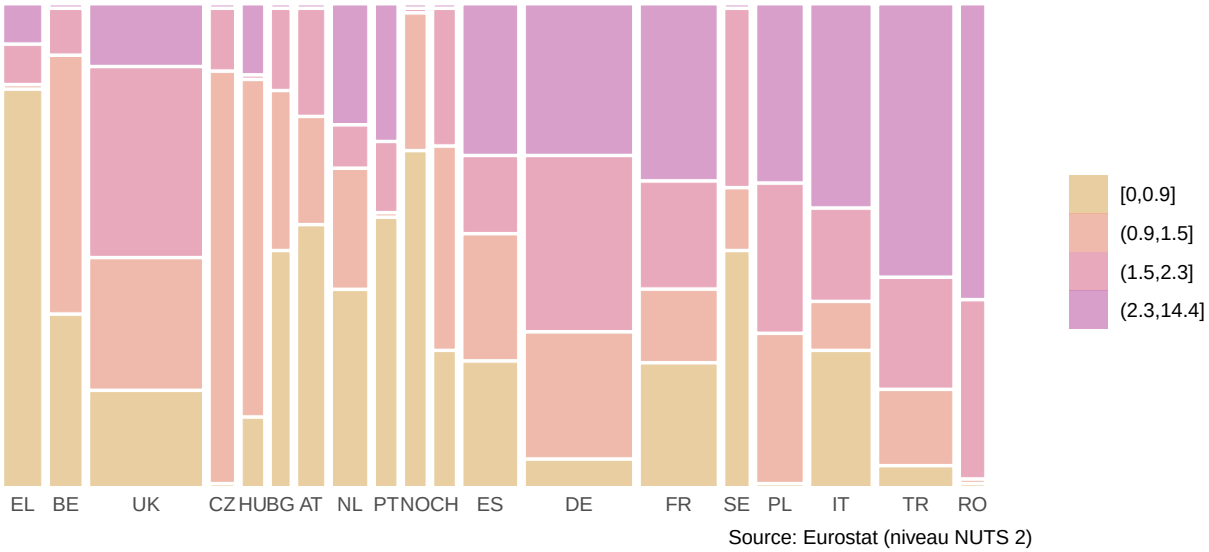
Pour avoir une idée de la proportion relative entre les pays, on peut avoir envie de modifier le graphique précédent.

```
library(ggmosaic)
NUTS2_year %>%
  filter(!is.na(population), année==2015) %>%
  mutate(
    population = (population/1000000) %>% round(1) %>% cut(., breaks=quantile(.), include.lowest =TRUE)
    pays       = str_sub(id_anc, end=2)
  ) %>%
  group_by(pays, population) %>% mutate(n=n()) %>% ungroup %>%
  group_by(pays) %>% mutate(n_pays=n()) %>% ungroup %>% filter(n_pays > 5) %>%
  mutate(pays = pays %>% fct_reorder2(population, n/n_pays) %>% fct_rev) %>%
  ggplot()+
  geom_mosaic(aes(x=product(population, pays), fill=population))+
  scale_fill_hue(h=c(60,-40), l=c(80,75,70,65), c=50)+
  scale_y_continuous(expand = c(0, 0), breaks=NULL)+
  scale_x_productlist(expand = c(0, 0))+
  labs(
    fill     = NULL,
    x        = NULL,
    y        = NULL,
    title    = "Istanbul (Turquie) surpeuplée, Oulu (Finlande) isolée",
    subtitle = "Population et superficie en Europe (2015)",
```

```
    caption  = "Source: Eurostat (niveau NUTS 2)"
  ) +
  theme_minimal()+
  theme(
    panel.grid.major = element_blank()
  )
```

Istanbul (Turquie) surpeuplée, Oulu (Finlande) isolée

Population et superficie en Europe (2015)



Source: Eurostat (niveau NUTS 2)

Scénarisation d'un graphique (11h00-12h00)

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. Voyons comment nous y prendre avec un exemple précis.

La scénarisation est l'aboutissement de toute la recherche graphique. Une fois qu'on exploré de nombreuses possibilités et qu'on a choisi une représentation pertinente, reste à accompagner le spectateur dans la lecture du graphique, pour susciter de l'intérêt et lui permettre d'exercer son jugement critique.

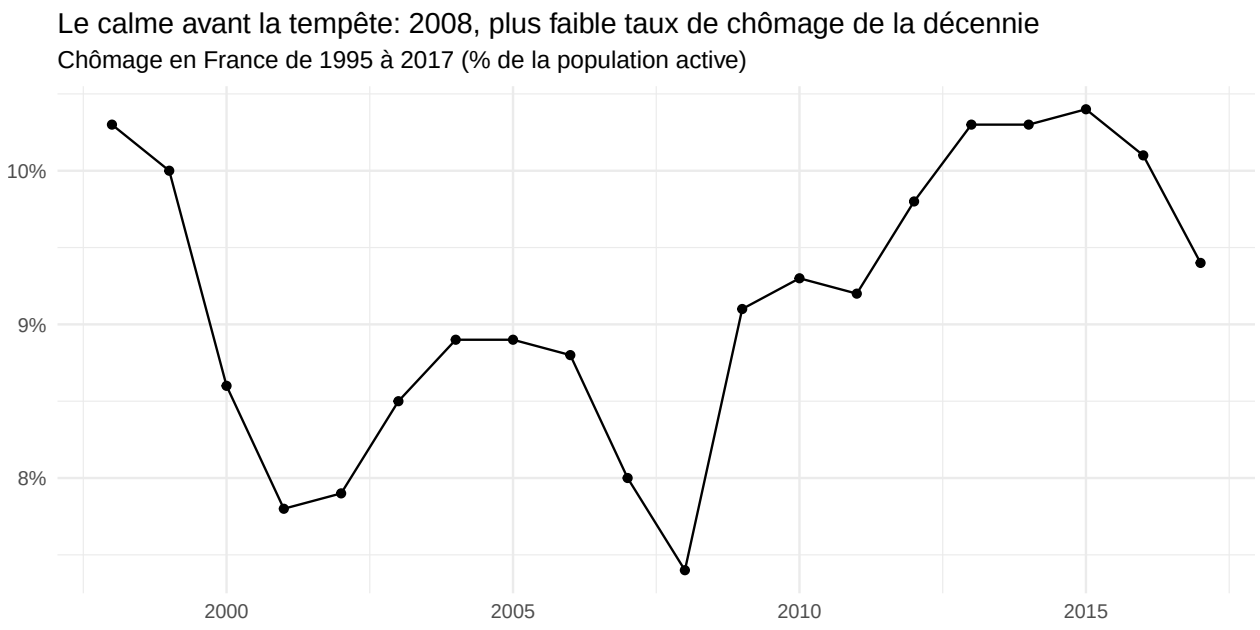
Essayons avec les données du chômage.

```
load("chomage.RData")
```

1. Représentez le chômage au cours du temps

```
chomage %>%
  ggplot(aes(.....................................................)) +
  .....................................................................
  .....................................................................
  .....................................................................
  .....................................................................
  .....................................................................
  .....................................................................
```

```
  ..................................................................
  ..................................................................
  ..................................................................
  ..................................................................
theme_minimal() +
labs(
  ..................................................................
  ..................................................................
  ..................................................................
  ..................................................................
  ..................................................................
  ..................................................................
)
```



2. Qu'est-ce que le chômage au sens du Bureau internationnal du travail (BIT)? Envisagez des situations extrêmement distinctes, mais étant carractérisées par un même taux de chômage fictif de 0%.

3. Notre base de donnée contient deux autres variables. Offrent-elles un regard complémentaire sur la nature de l'activité en France dans la précédente décennie?

4. Proposez un ou plusieurs graphiques synthétisant ces différents aspects

5. Voici une proposition (le graphique est disponible au format .png). Quels sont les avantages d'une telle visualisation? Habillez ce graphique à l'aide de *Libre Office Draw*.

En plus des habituels titre, source, etc., votre habillage comprendra notamment une scénarisation à l'aide de petites notes de lecture numérotées, pour aide le lecteur à s'approprier ce graphique complexe. (En principe, vous pouriez également placer à la main les étiquettes des années de façon plus harmonieuse que ne le fait `ggplot2` automatiquement.)

```
library(ggrepel)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(cowplot)
```

```
## Warning: package 'cowplot' was built under R version 3.5.2
```

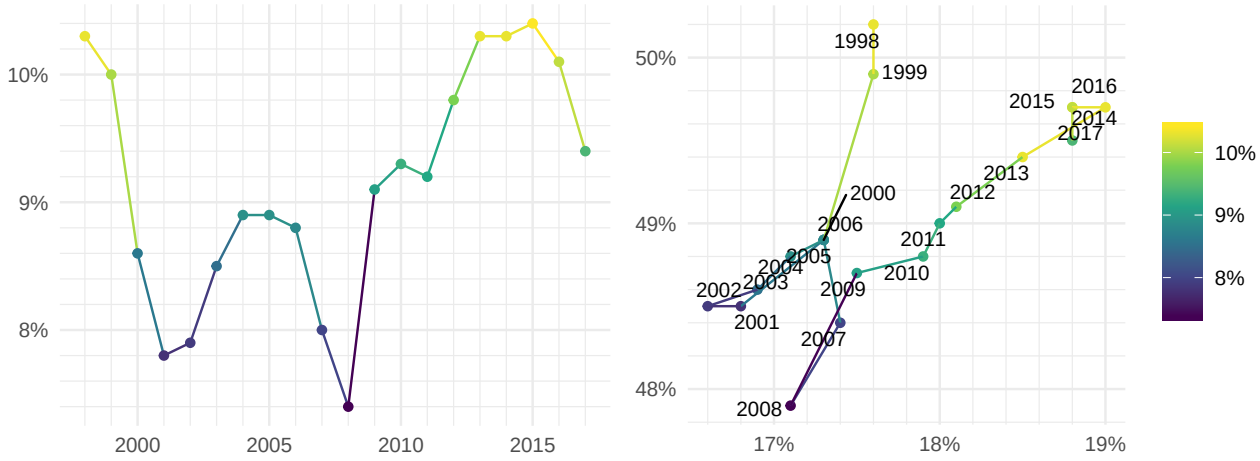
```
##
## Attaching package: 'cowplot'
```

```
## The following object is masked from 'package:ggplot2':
##
##     ggsave
```

```
plot1 <- chomage %>%
  ggplot(aes(x=Année, y=Chômage)) +
  geom_path(aes(group=1, col=Chômage)) +
  geom_point(aes(col=Chômage)) +
  scale_x_continuous(minor_breaks = 0:3000)+
  scale_y_continuous("", label=scales:::percent_format(accuracy =1), minor_breaks=seq(0,1,0.002)) +
  scale_colour_continuous(type='viridis', breaks=seq(0,1,0.01), label=scales:::percent_format(accuracy =
  theme_minimal() +
  coord_fixed(ratio=485) +
  labs(x=NULL, y=NULL)

plot2 <- chomage %>% ggplot(aes(x=`Temps partiel`, y=Inactivité)) +
  geom_point(aes(col=Chômage)) +
  geom_path(aes(group=1, col=Chômage)) +
  geom_text_repel(aes(label=Année), force=2, size=3) +
  scale_x_continuous(breaks=seq(0,1,0.01), minor_breaks=seq(0,1,0.002), label=scales:::percent_format(a
  scale_y_continuous(breaks=seq(0,1,0.01), minor_breaks=seq(0,1,0.002), label=scales:::percent_format(a
  scale_color_viridis_c(breaks=seq(0,1,0.01), label=scales:::percent_format(accuracy = 1))+
  theme_minimal() +
  coord_equal() +
  labs(x=NULL, y=NULL, color=NULL)

plot_grid(plot1, plot2)
```



ß

Critique de graphiques (12h15-12h45)

Graphique 1

ß

Graphique 2

Graphique 3

- Histogramme bizarre des données fiscales
-
-