

# Dernière exploration du `{tidyverse}` avec le package `{purrr}`

Marie VAUGOYEAU

02/05/2023

## Contents

Le pipe	1
Les données	3
Visualisation rapide des données	3
Analyse univariée	6
Variable qualitative . . . . .	6
Variable quantitative . . . . .	10
En savoir un peu plus sur moi	18

*Ce support, produit pour le live du 27 juin 2023 sur Twitch, est mis à disposition selon les termes de la Licence Creative Commons Attribution 4.0 International.*

## Le pipe

Lors des Rencontres R 2023 à Avignon, j'ai parlé du pipe pendant ma présentation.

Suite aux discussions que j'ai eu dessus après, je préfère détailler à nouveau ici le pipe `%>%` du package `{magrittr}` et le pipe natif `|>` maintenant présent dans le package `{base}` (pour les versions de R > 4.1).

```
# sans pipe
mean(sqrt(c(1:10)*3))
```

```
## [1] 3.89162
```

```
# avec pipe
c(1:10)*3 |>
  sqrt() |>
  mean()
```

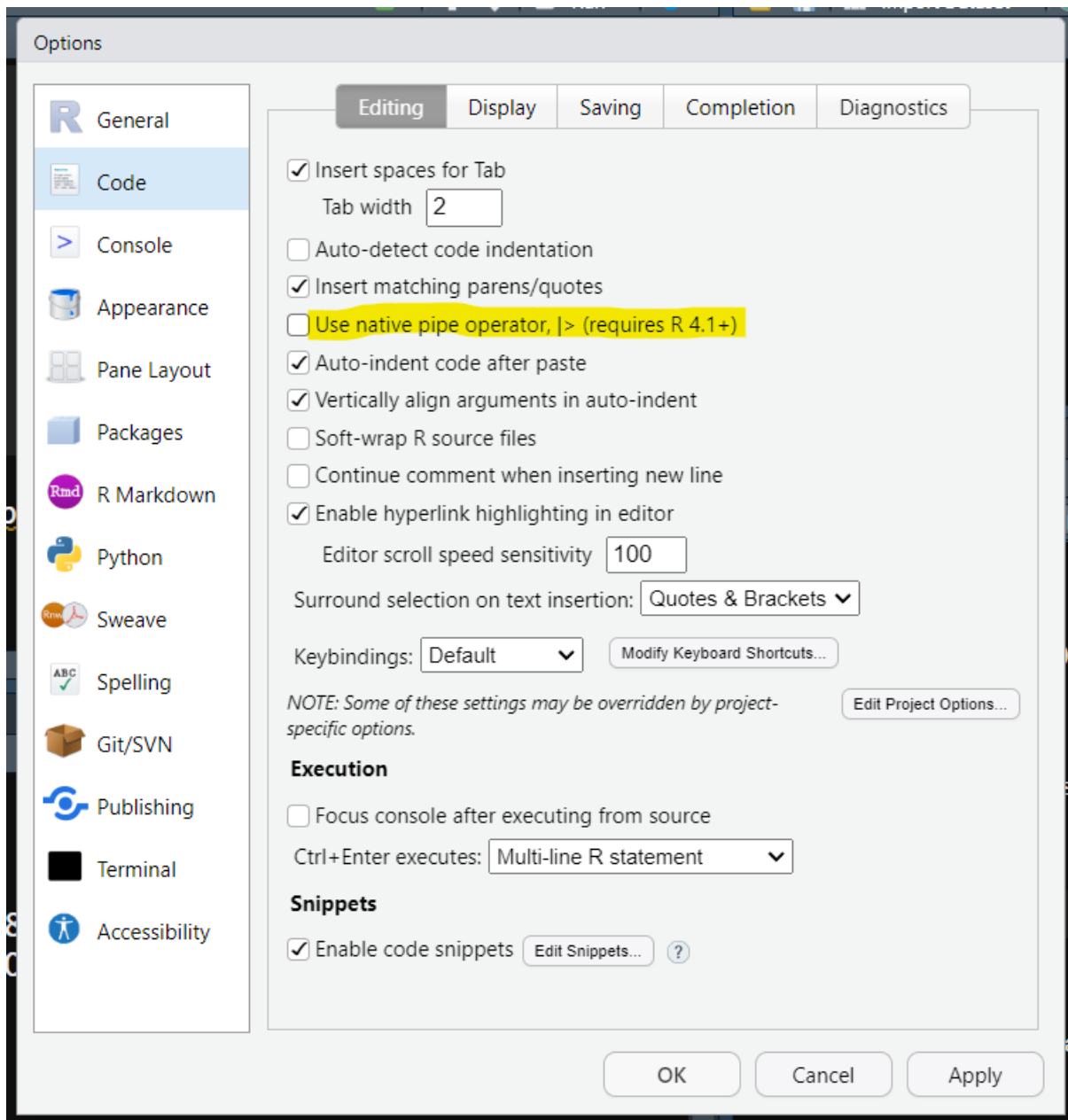
```
## [1] 1.732051 3.464102 5.196152 6.928203 8.660254 10.392305 12.124356
## [8] 13.856406 15.588457 17.320508
```

```
# attention à l'utilisation des parenthèses
(c(1:10)*3) |>
  sqrt() |>
  mean()
```

```
## [1] 3.89162
```

Ici cela fonctionne sans chargé de package car j'utilise le pipe natif `|>`.

Cette option est modifiable dans **Tools > Global Options > Code > Editing**



Si l'option **pipe natif** est décoché, le raccourci clavier **Ctrl + Maj + M** donne le pipe de **{magrittr}** mais ne permet pas de l'utiliser sans charger le package !

**Attention**, lors du chargement de **{tidyverse}** le pipe de **{magrittr}** est automatiquement chargé.

```
(c(1:10)*3) %>%
  sqrt() %>%
  mean()
```

```
## Error in (c(1:10) * 3) %>% sqrt() %>% mean(): impossible de trouver la fonction "%>%"
library(magrittr)
(c(1:10)*3) %>%
  sqrt() %>%
  mean()
```

```
## [1] 3.89162
```

D'un point de vue utilisation, `|>` est plus rapide que `%>%` mais implique d'utiliser une version postérieure à 4.1.

## Les données

Utilisation du jeu de données `penguins` du package `{palmerpenguins}` qui recense les caractéristiques des pingouins de l'archipel de Palmer.

Plus d'informations sur ce jeu de données dans la page d'aide `help(penguins)`.

## Visualisation rapide des données

Avec la fonction très généraliste `plot()` chargée de base dans l'environnement.

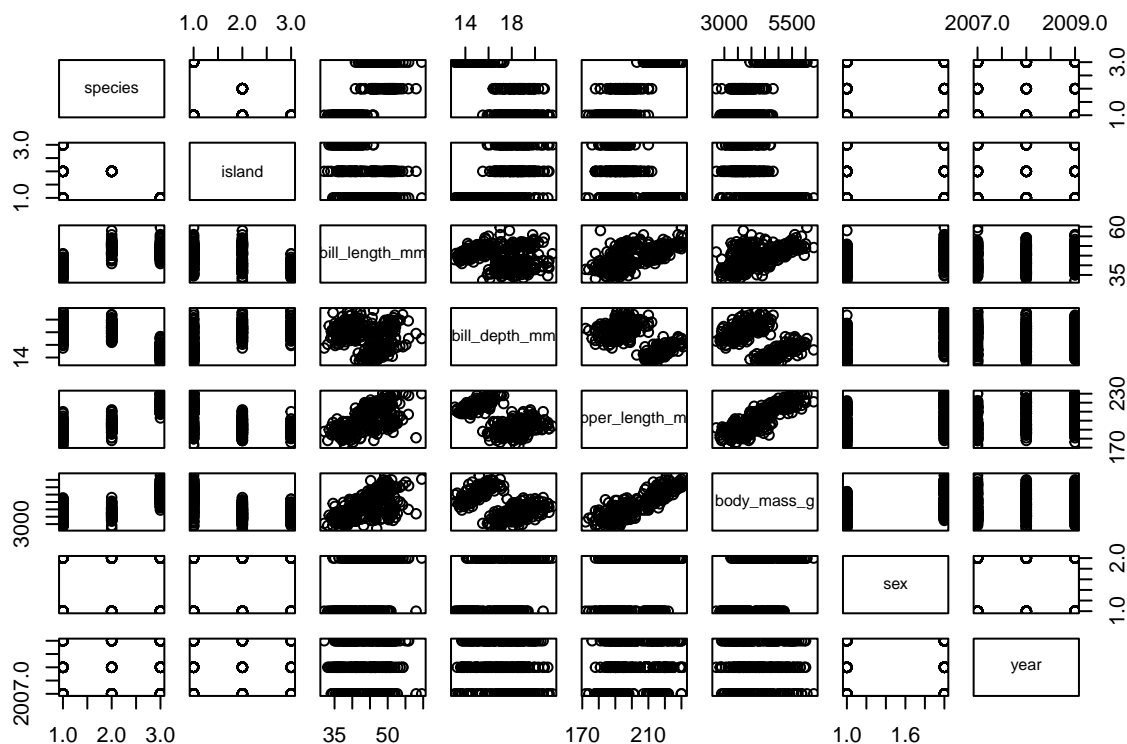
```
library(palmerpenguins)
```

```
penguins
```

```
## # A tibble: 344 x 8
##   species island   bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
##   <fct>   <fct>         <dbl>         <dbl>         <int>         <int>
## 1 Adelie  Torgersen         39.1          18.7          181          3750
## 2 Adelie  Torgersen         39.5          17.4          186          3800
## 3 Adelie  Torgersen         40.3           18          195          3250
## 4 Adelie  Torgersen          NA           NA           NA           NA
## 5 Adelie  Torgersen         36.7          19.3          193          3450
## 6 Adelie  Torgersen         39.3          20.6          190          3650
## 7 Adelie  Torgersen         38.9          17.8          181          3625
## 8 Adelie  Torgersen         39.2          19.6          195          4675
## 9 Adelie  Torgersen         34.1          18.1          193          3475
## 10 Adelie Torgersen         42           20.2          190          4250
## # i 334 more rows
## # i 2 more variables: sex <fct>, year <int>
```

```
View(penguins)
```

```
plot(penguins)
```



Pour avoir un aperçu des données il est intéressant d'utiliser la fonction `summary()` présent dans le package `{base}`.

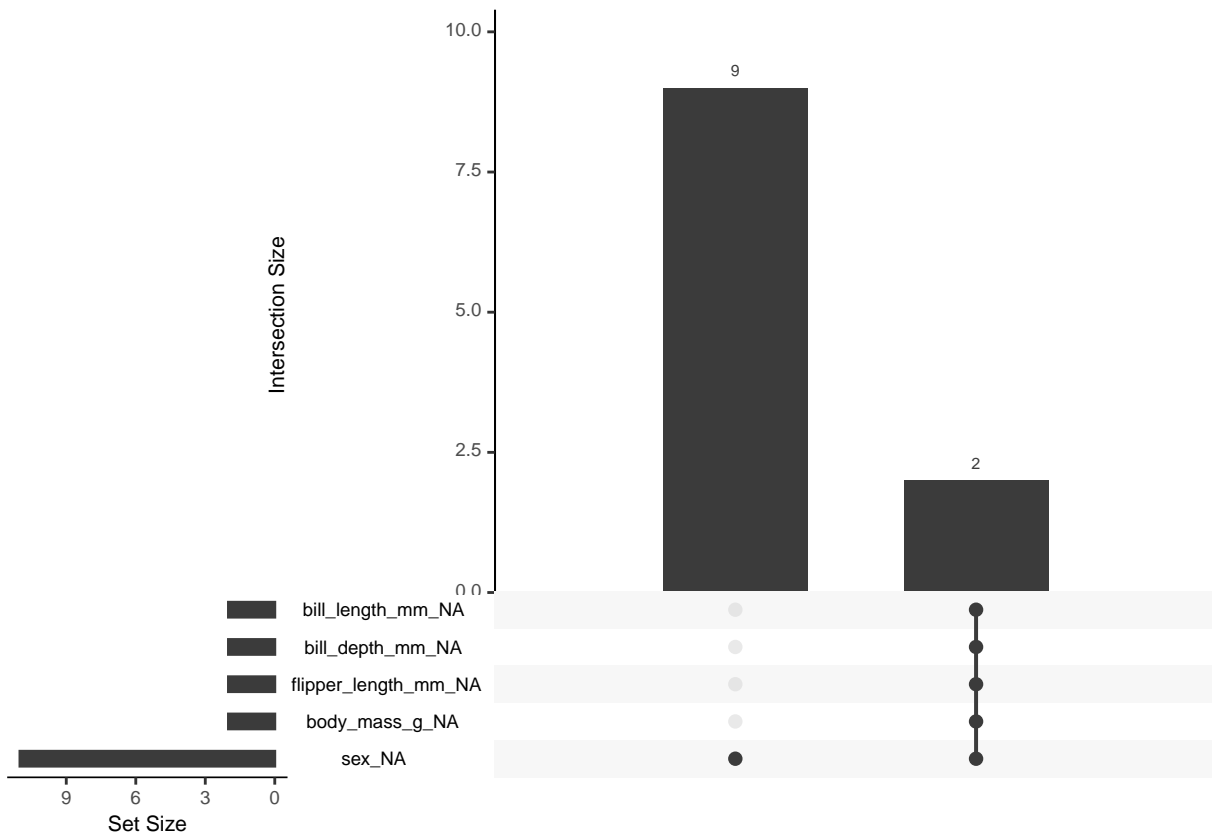
```
summary(penguins)
```

```
##      species      island  bill_length_mm  bill_depth_mm
## Adelie   :152  Biscoe   :168   Min.    :32.10   Min.    :13.10
## Chinstrap: 68  Dream    :124   1st Qu.:39.23   1st Qu.:15.60
## Gentoo   :124  Torgersen: 52   Median :44.45   Median :17.30
##
##                               Mean    :43.92   Mean    :17.15
##                               3rd Qu.:48.50   3rd Qu.:18.70
##                               Max.    :59.60   Max.    :21.50
##                               NA's    :2       NA's    :2
## flipper_length_mm  body_mass_g      sex      year
## Min.    :172.0     Min.    :2700   female:165   Min.    :2007
## 1st Qu.:190.0     1st Qu.:3550   male  :168   1st Qu.:2007
## Median :197.0     Median :4050   NA's  : 11   Median :2008
## Mean    :200.9     Mean    :4202                   Mean    :2008
## 3rd Qu.:213.0     3rd Qu.:4750                   3rd Qu.:2009
## Max.    :231.0     Max.    :6300                   Max.    :2009
## NA's    :2         NA's    :2
```

Il y a des valeurs manquantes, il faut donc les visualiser.

`{naniar}` est un package très performant pour travailler sur les données manquantes.

```
naniar::gg_miss_upset(penguins)
```



Pour visualiser différemment le tableau de données, il est possible d'utiliser la fonction `glimpse()` du `{tidyverse}`. Plus d'information sur le `{tidyverse}` dans le paragraphe ci-dessus Le `{tidyverse}`.

```
library(tidyverse)
```

```
glimpse(penguins)
```

```
## Rows: 344
## Columns: 8
## $ species      <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Adel-
## $ island       <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torgerse-
## $ bill_length_mm <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.9, 39.2, 34.1, ~
## $ bill_depth_mm <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20.6, 17.8, 19.6, 18.1, ~
## $ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 181, 195, 193, 190, 186~
## $ body_mass_g   <int> 3750, 3800, 3250, NA, 3450, 3650, 3625, 4675, 3475, ~
## $ sex          <fct> male, female, female, NA, female, male, female, male~
## $ year         <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007~
```

**Attention** le chargement de certain package remplace des fonctions déjà chargées par celles chargée en dernière.

Par exemple, le chargement du package `{tidyverse}` ou `{dplyr}` remplace la fonction `filter()` du package `{stat}` par la sienne.

# Analyse univariée

## Variable qualitative

Il y a trois variables qualitatives ici : `species`, `island` et `sex`.

Toutes les trois sont finis -> donc on peut réaliser directement des tableaux de contingence.

### Tableau de contingence

```
# fonction `table()` du package `{base}`  
table(penguins$species)
```

```
##  
##      Adelie Chinstrap      Gentoo  
##      152         68       124
```

```
table(penguins$island)
```

```
##  
##      Biscoe      Dream Torgersen  
##      168       124         52
```

```
table(penguins$sex)
```

```
##  
## female   male  
##      165    168
```

*# ne permet pas de voir les NA !*

```
# fonction `count()` du package `{dplyr}`  
count(penguins, sex)
```

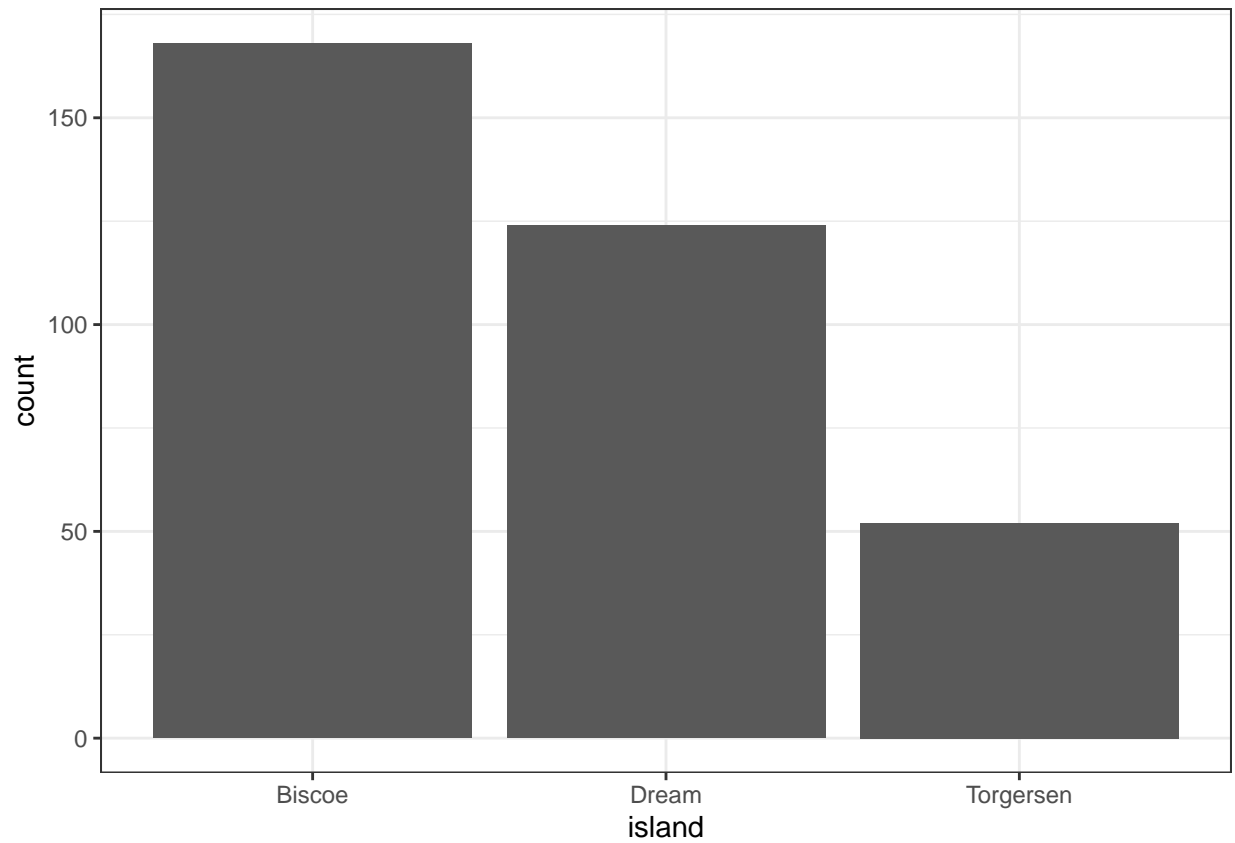
```
## # A tibble: 3 x 2  
##   sex      n  
##   <fct> <int>  
## 1 female  165  
## 2 male    168  
## 3 <NA>    11
```

**Attention :** Il vaut mieux utiliser `count()` de `{dplyr}` pour réaliser les tableaux de contingence car il permet de voir les valeurs manquantes (NA).

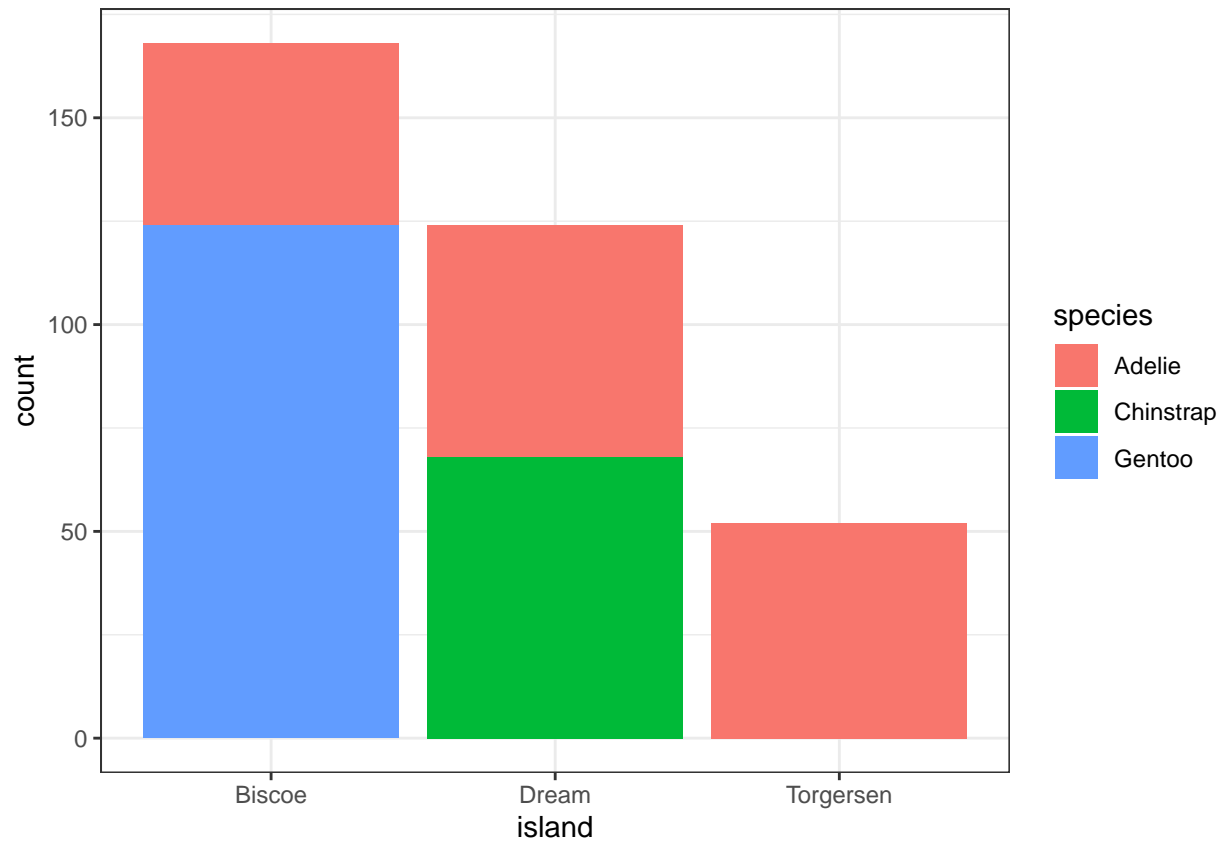
### Représentation graphique

Ressource conseillée pour la réalisation de graphiques : From Data to Viz.

```
# diagramme en barres  
penguins |>  
  ggplot() +  
  aes(x = island) +  
  geom_bar() +  
  theme_bw()
```

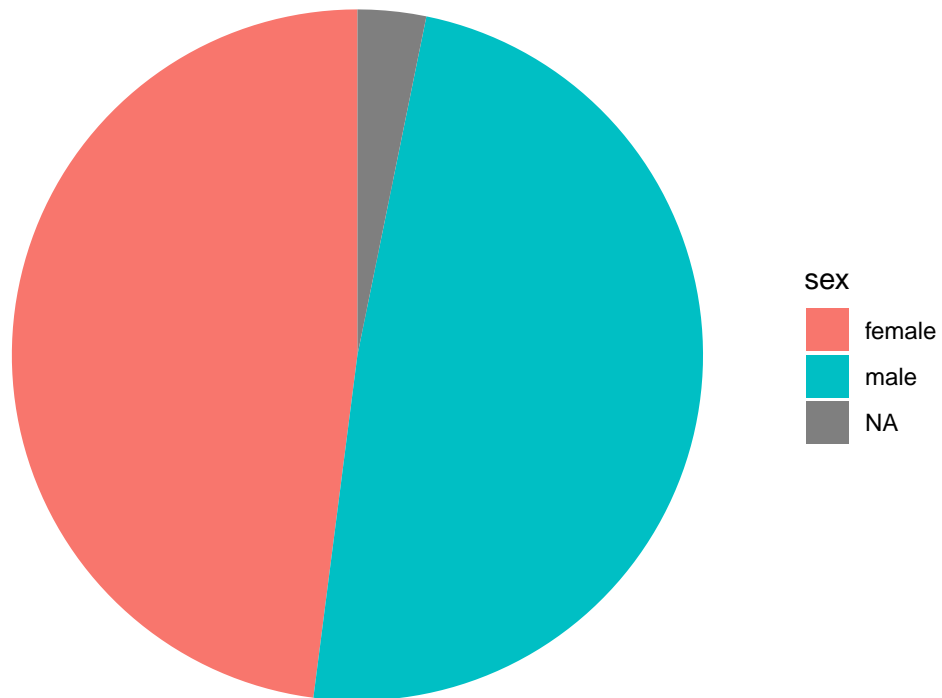


```
# diagramme en barres colorées par l'espèces
penguins |>
  ggplot() +
  aes(x = island, fill = species) +
  geom_bar() +
  theme_bw()
```



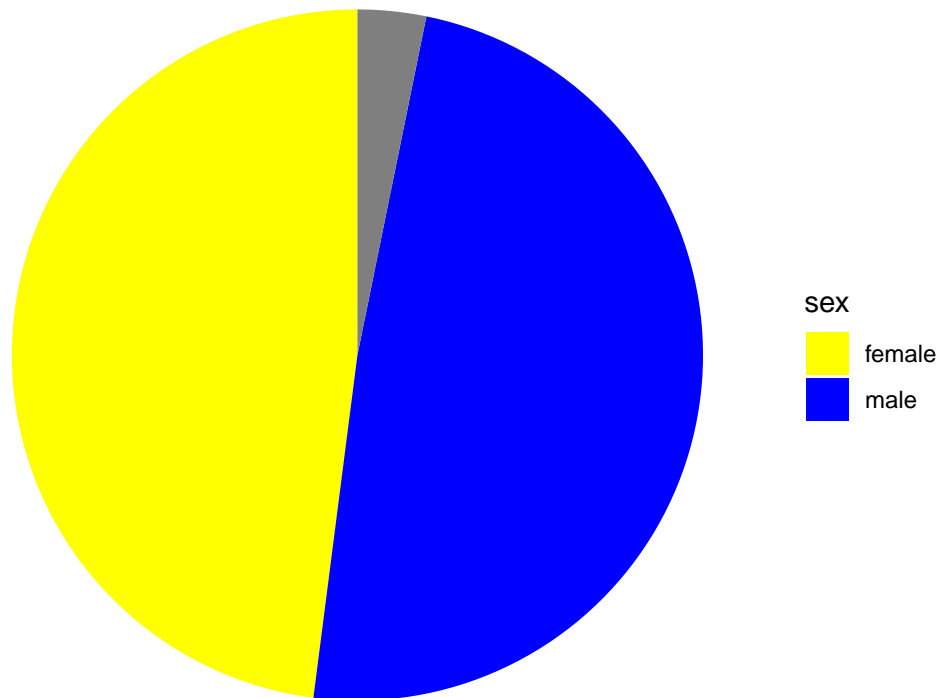
```
# diagramme circulaire
penguins |>
  count(sex) |>
  ggplot() +
  aes(x = "", y = n, fill = sex) +
  geom_bar(stat = "identity") +
  coord_polar("y") +
  theme_void()
```





```
# adapter la couleur
couleur <- c("female" = "yellow", "male" = "blue")

# changer la couleur d'un graphique
penguins |>
  count(sex) |>
  ggplot() +
  aes(x = "", y = n, fill = sex) +
  geom_bar(stat = "identity") +
  coord_polar("y") +
  scale_fill_manual(values = couleur) +
  theme_void()
```



## Variable quantitative

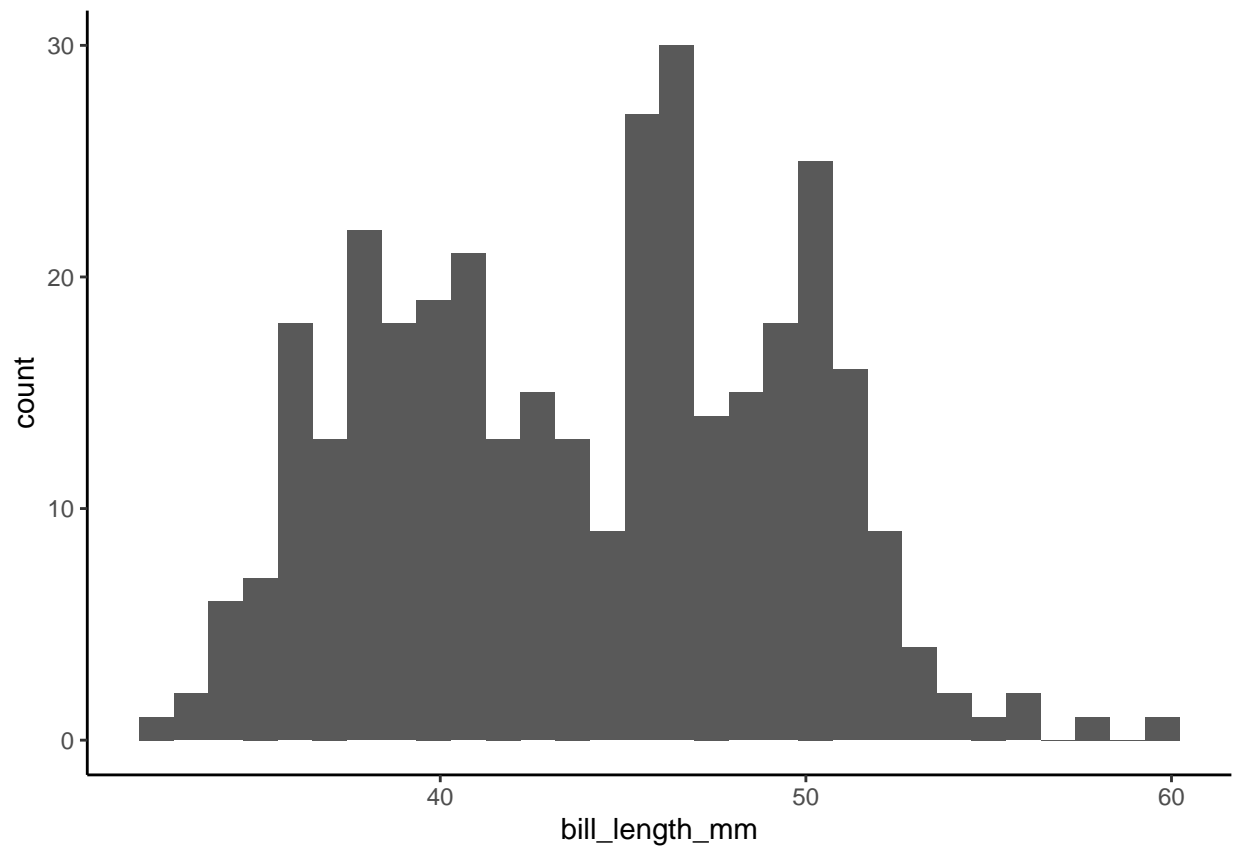
Il y a 5 variables quantitatives, il est possible d'étudier leurs dispersion grâce aux histogrammes ou de calculer les mesures de cette dispersion.

**Attention**, l'une des variables est l'année.

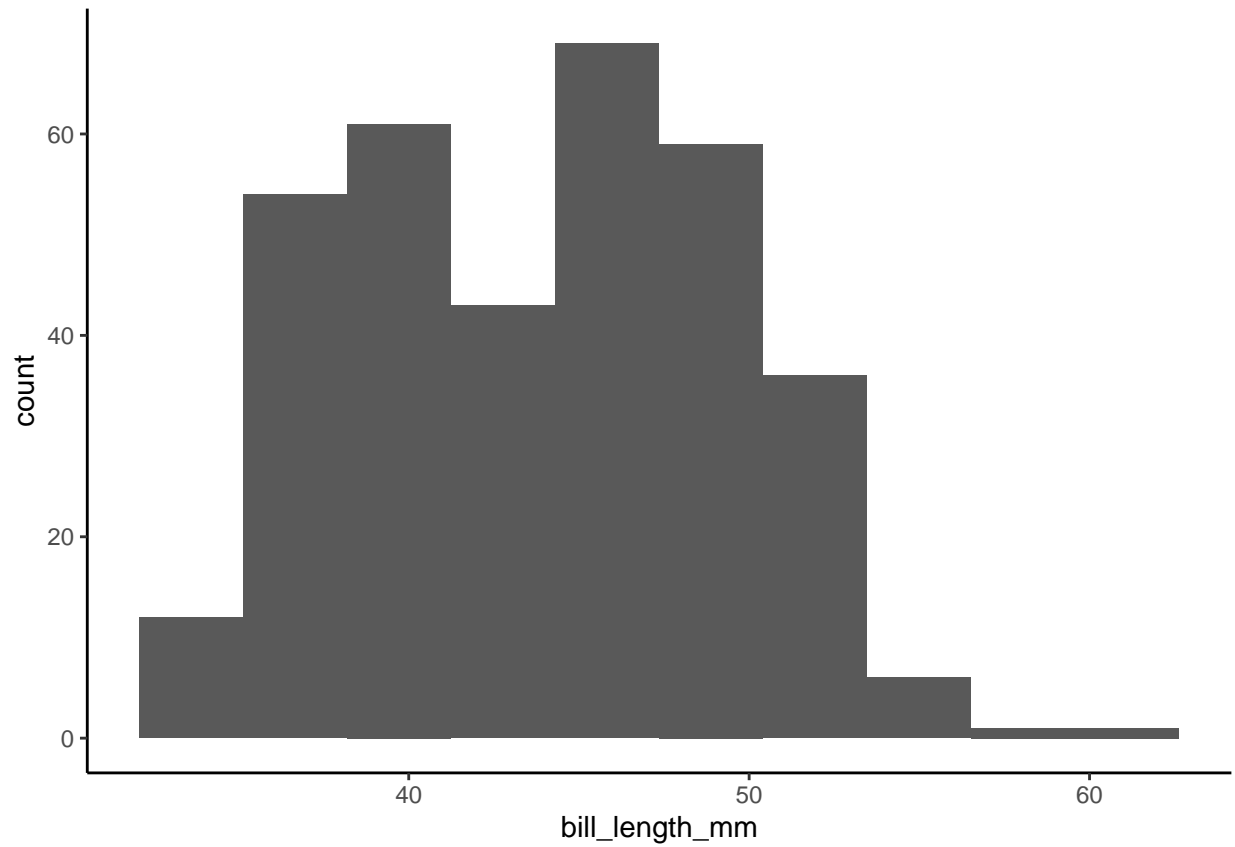
## Histogramme

**Attention** au nombre d'intervalles représenté.

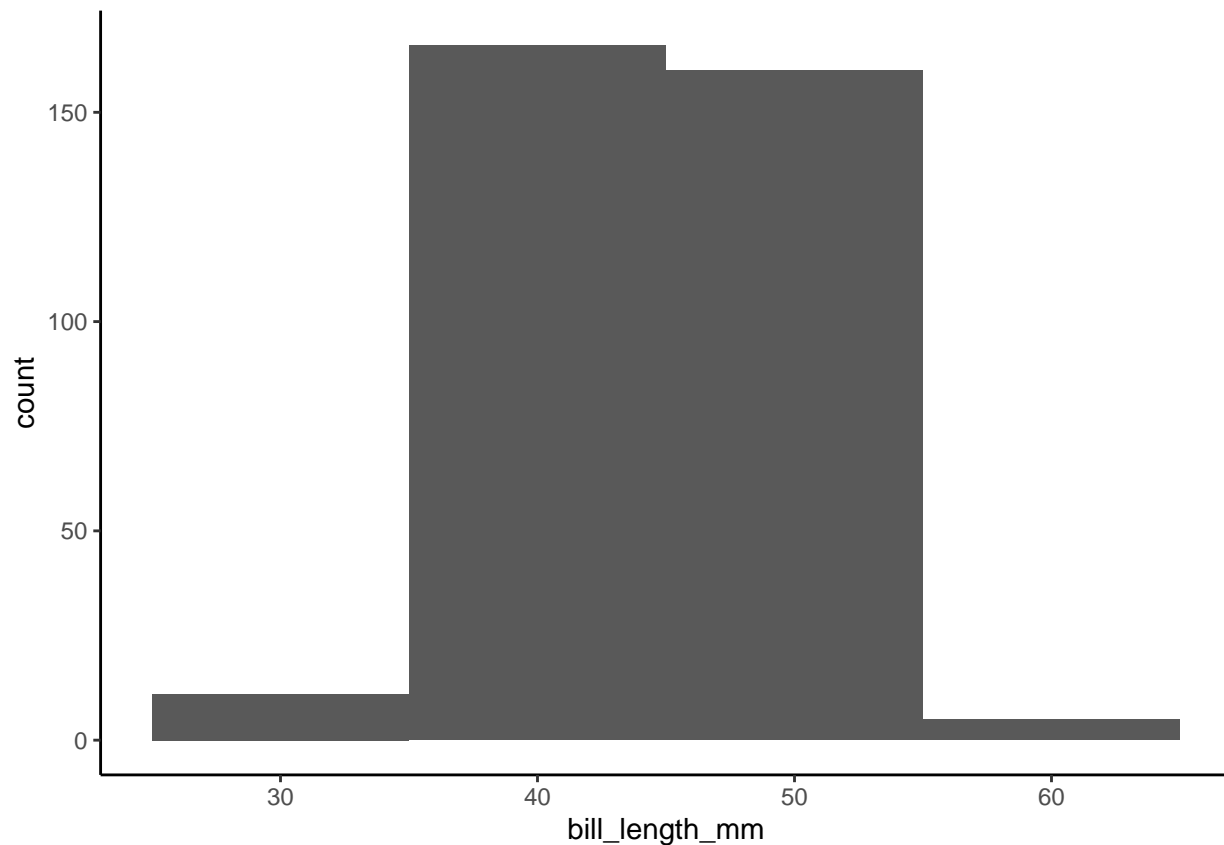
```
penguins |>
  ggplot() +
  aes(x = bill_length_mm) +
  geom_histogram() +
  theme_classic()
```



```
# changement nombre d'intervalles (10)
penguins |>
  ggplot() +
  aes(x = bill_length_mm) +
  geom_histogram(bins = 10) +
  theme_classic()
```



```
# largeur de la barre
penguins |>
  ggplot() +
  aes(x = bill_length_mm) +
  geom_histogram(binwidth = 10) +
  theme_classic()
```



### Calcul des mesures de dispersion

```
summary(penguins)
```

```
##      species      island  bill_length_mm  bill_depth_mm
## Adelie   :152  Biscoe   :168    Min.   :32.10    Min.   :13.10
## Chinstrap: 68  Dream    :124   1st Qu.:39.23   1st Qu.:15.60
## Gentoo   :124  Torgersen: 52   Median :44.45   Median :17.30
##                                     Mean   :43.92   Mean   :17.15
##                                     3rd Qu.:48.50   3rd Qu.:18.70
##                                     Max.   :59.60   Max.   :21.50
##                                     NA's   :2      NA's   :2
## flipper_length_mm  body_mass_g      sex      year
## Min.   :172.0      Min.   :2700    female:165  Min.   :2007
## 1st Qu.:190.0      1st Qu.:3550    male  :168  1st Qu.:2007
## Median :197.0      Median :4050    NA's  : 11  Median :2008
## Mean   :200.9      Mean   :4202                      Mean   :2008
## 3rd Qu.:213.0      3rd Qu.:4750                      3rd Qu.:2009
## Max.   :231.0      Max.   :6300                      Max.   :2009
## NA's   :2          NA's   :2
```

```
mean(penguins$bill_depth_mm)
```

```
## [1] NA
```

```
# NA car présence de valeur manquantes
```

```

mean(penguins$bill_length_mm, na.rm = TRUE)

## [1] 43.92193
max(penguins$bill_length_mm, na.rm = TRUE)

## [1] 59.6
pingouins_sans_na <- penguins |>
  drop_na()

median(pingouins_sans_na$bill_length_mm)

## [1] 44.5
pingouins_sans_na |>
  summarise(
    across(
      .cols = where(is.numeric),
      .fns = list(
        moyenne = ~ mean(.x),
        minimum = ~ min(.x),
        maximum = ~ max(.x)
      ),
      .names = "{col} {fn}"
    )
  ) |>
  pivot_longer(everything()) |>
  separate_wider_delim(
    name,
    delim = " ",
    names = c("variable", "measure")
  ) |>
  pivot_wider(names_from = measure, values_from = value)

## # A tibble: 5 x 4
##   variable      moyenne minimum maximum
##   <chr>         <dbl>   <dbl>   <dbl>
## 1 bill_length_mm    44.0     32.1     59.6
## 2 bill_depth_mm     17.2     13.1     21.5
## 3 flipper_length_mm 201.      172      231
## 4 body_mass_g      4207.    2700     6300
## 5 year             2008.    2007     2009

```

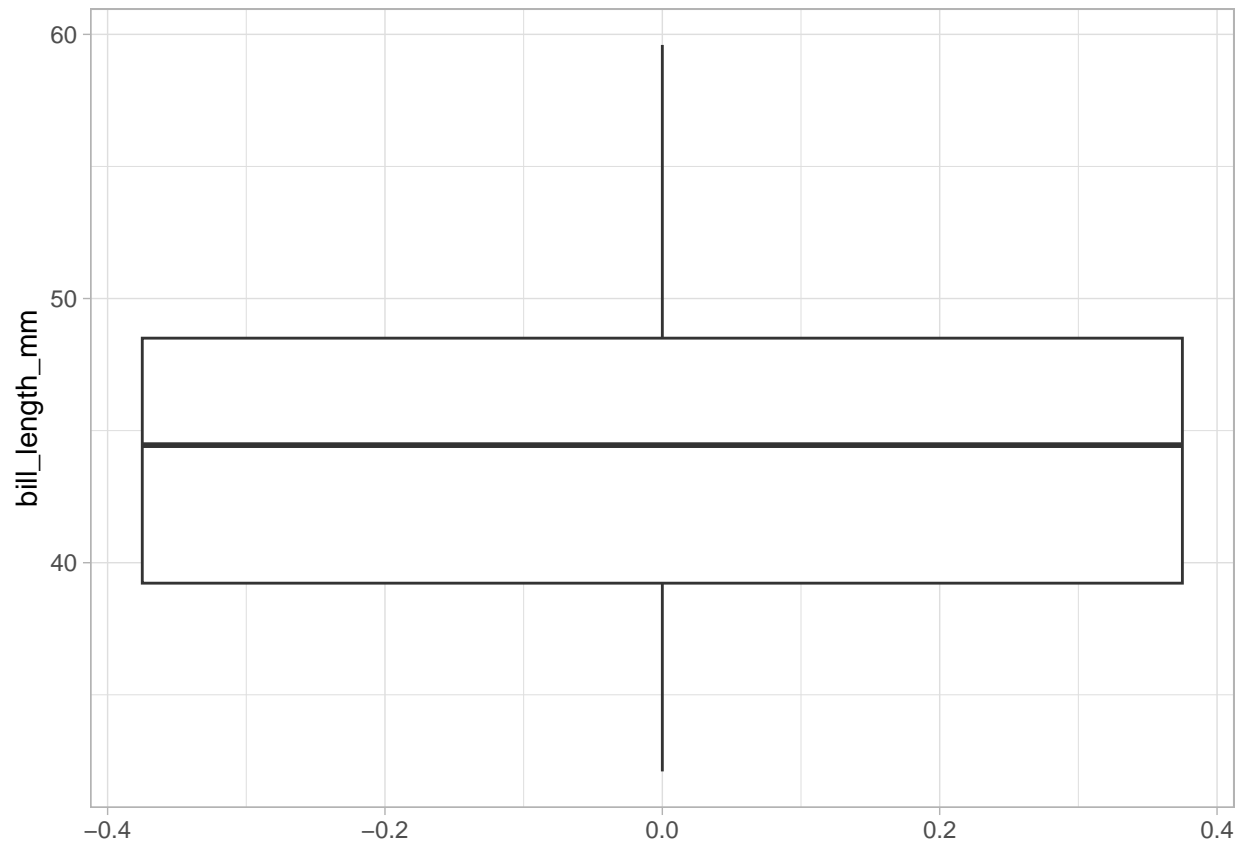
## Boîte à moustaches

Graphique généralisant les données de dispersion.

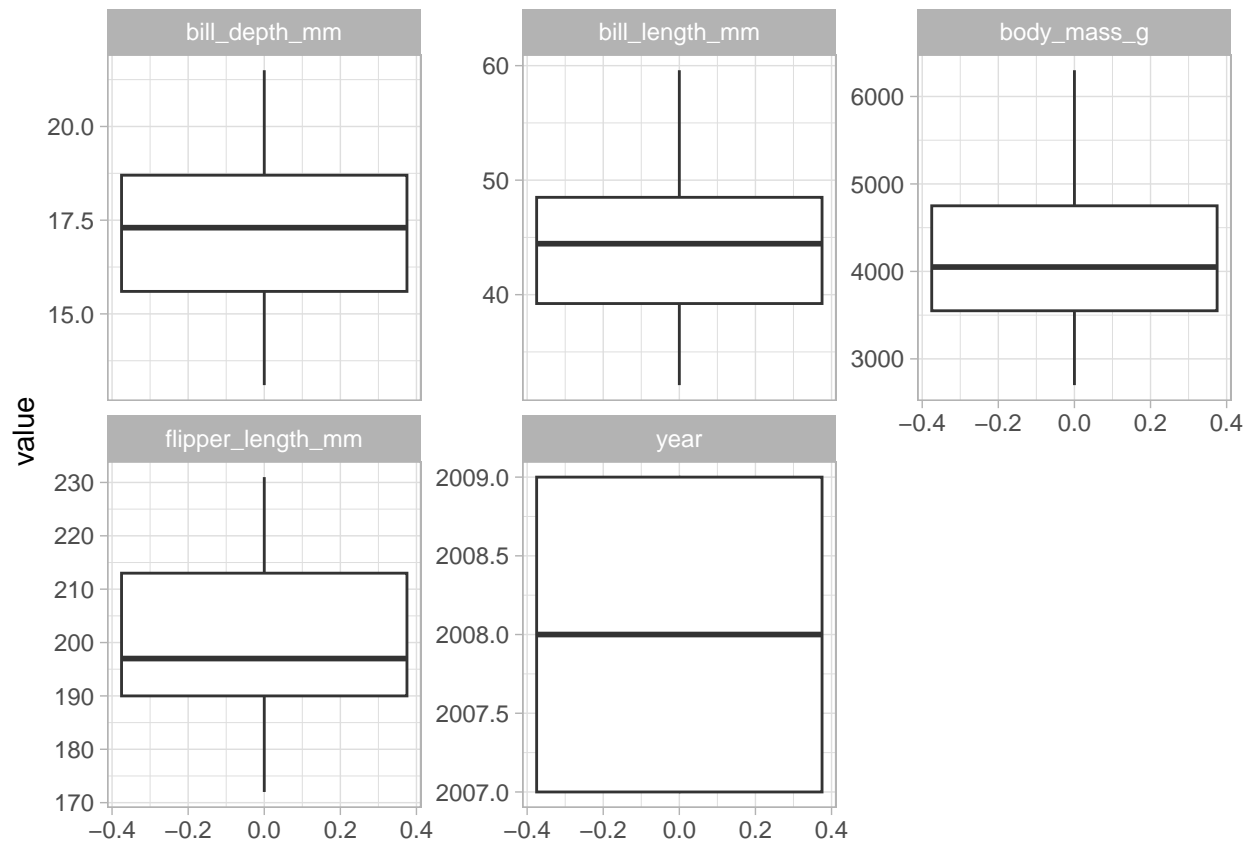
```

penguins |>
  ggplot() +
  aes(y = bill_length_mm) +
  geom_boxplot() +
  theme_light()

```



```
# sur toutes les colonnes numériques
penguins |>
  pivot_longer(
    cols = where(is.numeric)
  ) |>
  ggplot() +
    aes(y = value) +
    facet_wrap(~ name, scales = "free_y") +
    geom_boxplot() +
    theme_light()
```



### Cas particulier des dates

Il est clair que le traitement de l'année est un peu particulier, à mi chemin entre la variable qualitative et la variable quantitative.

S'il y a peu de dates, comme c'est le cas ici autant la traiter comme une variable qualitative, c'est-à-dire réaliser un tableau de contingence pour voir le nombre d'individus capturé par an.

```
# tableau de contingence
```

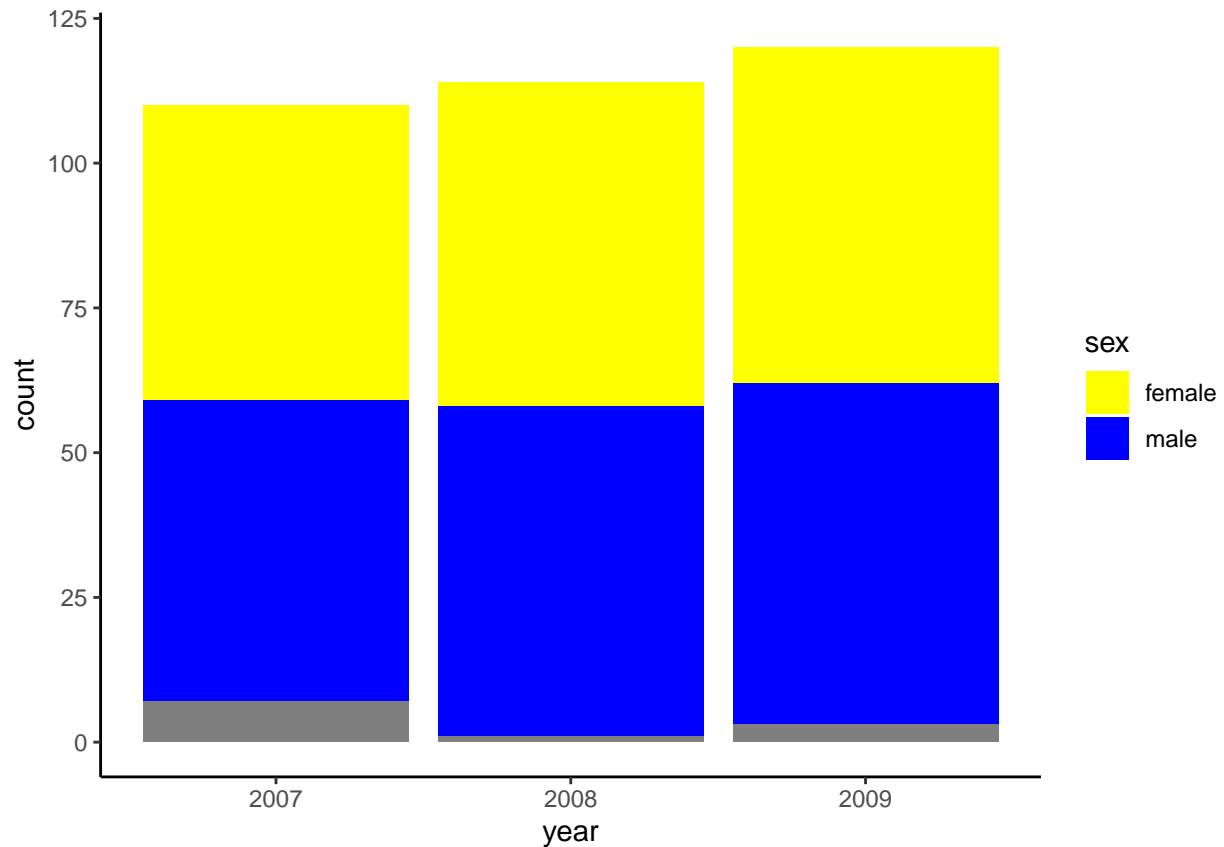
```
penguins |>
  count(year)
```

```
## # A tibble: 3 x 2
##   year     n
##   <int> <int>
## 1  2007   110
## 2  2008   114
## 3  2009   120
```

```
# réalisation d'un diagramme en barres
```

```
penguins |>
  ggplot() +
    aes(x = year, fill = sex) +
    geom_bar() +
    scale_fill_manual(values = couleur) +
    theme_classic()
```

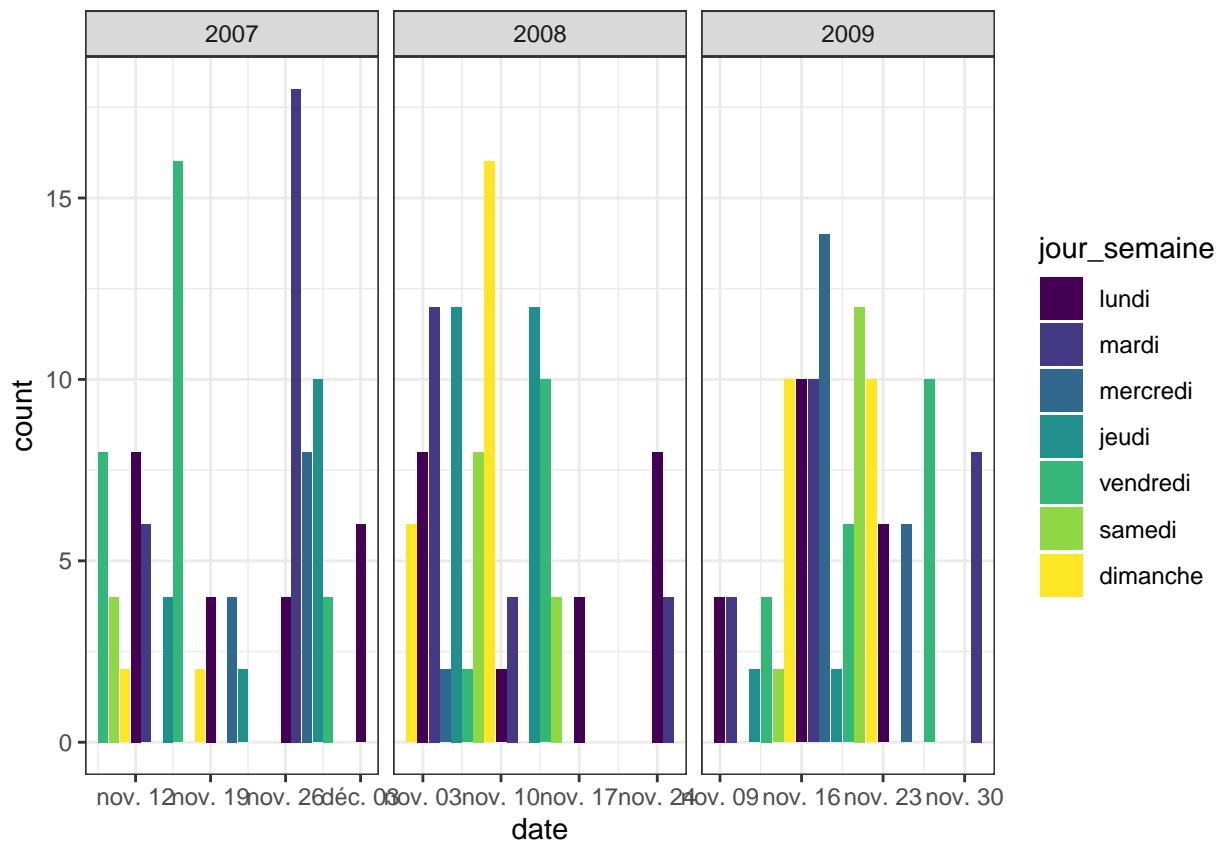




Si c'est une date complète, comment travailler dessus ?

```
# ajout de la date complète depuis le jeu de données `penguins_raw`
pingouins_modifie <- bind_cols(
  penguins,
  date = penguins_raw$`Date Egg`
) |>
  mutate(
    mois = month(date, label = TRUE, abbr = FALSE),
    jour = day(date),
    jour_semaine = wday(date, label = TRUE, abbr = FALSE, week_start = 1)
  )

# représentation en diagramme en barres
pingouins_modifie |>
  ggplot() +
  aes(x = date, fill = jour_semaine) +
  geom_bar() +
  facet_wrap(~ year, scales = "free_x") +
  theme_bw()
```



## En savoir un peu plus sur moi

Bonjour,

Je suis Marie Vaugoyeau et je suis disponible pour des **missions en freelance d'accompagnement à la formation** à R et à l'analyse de données et/ou en **programmation** (reprise de scripts, bonnes pratiques de codage, développement de package).

Ayant un **bagage recherche en écologie**, j'ai accompagné plusieurs chercheuses en biologie dans leurs analyses de données mais je suis ouverte à d'autres domaines.

Vous pouvez retrouver mes offres ici.

**En plus de mes missions de consulting je diffuse mes savoirs en R et analyse de données sur plusieurs plateformes :**

- J'ai écrit un **livre** aux éditions ENI
- Tous les mois je fais un **live sur Twitch** pour parler d'un package de R, d'une analyse
- Je rédige une **newsletter** de manière irrégulière pour parler de mes **inspirations** et transmettre **des trucs et astuces sur R**. Pour s'y inscrire, c'est par là. J'ai aussi un **blog**, en PLS en ce moment, qu'il faut que je reprenne.

Pour en savoir encore un peu plus sur moi, il y a LinkedIn et pour retrouver tous ces liens et plus encore, c'est ici

**N'hésitez pas à me contacter sur [marie.vaugoyeau@gmail.com](mailto:marie.vaugoyeau@gmail.com) !**

Bonne journée

Marie

