6 Exploration statistique d
données

6.1 Pré-requis

La première étape de toute analyse de données est l'exploration. Avant de se lancer dans des tests statis-
tiques et des procédures complexes, et à supposer q
les données dont vous disposez sont déjà dans un form
approprié, il est toujours très utile :

1. d'explorer visuellement les données dont on dispos
faisant des graphiques nombreux et variés, afin de c
prendre, notamment quelle est la distribution des
riables numériques, quelles sont les catégories l
représentées pour les variables qualitatives (fa
quelles sont les relations les plus marquantes ent
riables numériques et/ou catégorielles, etc. Vous
appris au Chapitre 3 comment produire toutes sortes de
graphiques avec ggplot2. Il va maintenant
falloir vous poser la question du choix des graphi
à produire du point de vue de l'exploration statist
de données inconnues.

2. d'explorer les données en calculant des indices de
tistiques descriptives. Ces indices relèvent en g
de 2 catégories : les indices de position (e.g. moye
médianes, quartiles…) et les indices de dispers
(e.g. variance, écart-type, intervalle inter-qua
Nous avons déjà vu comment utiliser la fonction
summarise() et son argument .by pour calculer des
moyennes ou des effectifs pour plusieurs sous-grou
de nos jeux de données (Section 5.8). Dans ce chapitre,
nous irons plus loin, et nous découvrirons d'une p
comment calculer d'autres indices statistiques p
nents, et comment utiliser d'autres fonctions en
plus utiles que summarise().

Nous verrons ensuite dans le Chapitre 7 comment calculer des indices d'incertitude (Section 7.4 et Section 7.5). Attention, il ne faudra pas confondre indices de dispersion et indices d'incertitude. Et enfin, avant de passer aux tests statistiques, nous verrons comment visualiser dispersion et incertitude (Chapitre 8).

Afin d'explorer ces questions, nous aurons besoin des packages suivants :

```
library(tidyverse)
library(skimr)
library(palmerpenguins)
library(nycflights13)
```

Comme vous le savez maintenant, les packages du tidyverse (Wickham 2023) permettent de manipuler facilement des tableaux de données et de réaliser des graphiques. Charger le tidyverse permet d'accéder, entre autres, aux packages readr (Wickham, Hester, et Bryan 2024), pour importer facilement des fichiers .csv au format tibble, tidyr (Wickham, Vaughan, et Girlich 2024) et dplyr (Wickham et al. 2023) pour manipuler des tableaux de données, ou encore ggplot2 (Wickham et al. 2024) pour produire des graphiques. Le package skimr (Waring et al. 2022) permet de calculer des résumés de données très informatifs. Les packages palmerpenguins (Horst, Hill, et Gorman 2022) et nycflights13 (Wickham 2021) fournissent des jeux de données qui seront faciles à manipuler pour illustrer ce chapitre (et les suivants).

> **Important**
>
> Si vous avez installé le tidyverse ou dplyr avant le printemps 2023, ré-installez dplyr avec install.packages("dplyr"). Ce package a en effet été mis à jour courant 2023, et nous aurons besoin de sa version v1.1.0 ou d'une version plus récente pour utiliser certaines fonctions. Chargez-le ensuite en mémoire avec library(dplyr).

> **Attention**
>
> Pensez à installer tous les packages listés ci-dessu… de les charger en mémoire si vous ne l'avez pas déjà fa… Si vous ne savez plus comment faire, consultez d'urge… la Section 1.4

Pour travailler dans de bonnes conditions, et puisqu… abordons maintenant les statistiques à proprement pa… vous conseille de créez un nouveau script dans le même d… que votre Rproj. Là encore, si vous ne savez plus comment faire consultez la Section 1.3

6.2 Créer des résumés avec la fonction reframe()

Dans la Section 5.8, nous avons vu comment utiliser la fonction summarise(), éventuellement .by pour argument pour calculer des statistiques descriptives variées. N'h… relire cette section si vous n'êtes pas sûr d'avoir tou… ou tout retenu. Les calculs que nous pouvons faire grâ… fonction summarise() impliquent des fonctions statistiques qui ne renvoient qu'une valeur à la fois lorsqu'on leu… une série de valeurs. Par exemple, si on dispose d'un v… numérique (les entiers compris entre 1 et 100 pour l'ex… :

```r
1:100
```

```
  [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
 [19]  19  20  21  22  23  24  25  26  27  28  29  30  31  32
 [37]  37  38  39  40  41  42  43  44  45  46  47  48  49  50
 [55]  55  56  57  58  59  60  61  62  63  64  65  66  67  68
 [73]  73  74  75  76  77  78  79  80  81  82  83  84  85  86
 [91]  91  92  93  94  95  96  97  98  99 100
```

la fonction mean() renvoie qu'une valeur, la moyenne des 100 nombres contenus dans le vecteur :

```r
mean(1:100)
```

[1] 50.5

De même pour les fonctions sd(), median(), ou toutes les autres fonctions listées à la fin de la Section 5.8.4

```r
sd(1:100)
```

[1] 29.01149

```r
median(1:100)
```

[1] 50.5

Il existe toutefois des fonctions qui renvoient plus
leur à la fois. Par exemple la fonction quantile() que nous avons utilisée dans un autre contexte à la Section 5.9.1 renvoie par défaut 5 éléments :

1. la valeur minimale contenue dans le vecteur (ou quan
   0%) : c'est la valeur la plus faible contenue dans la
   de données
2. le premier quartile du vecteur (Q1 ou quantile 25%
   est la valeur coupant l'échantillon en deux : 25% d
   observations du vecteur y sont inférieures et 75% y
   supérieures
3. la médiane du vecteur (Q2 ou quantile 50%) est la va
   leur coupant l'échantillon en deux : 50% des obser
   tions du vecteur sont inférieures à cette valeur et
   y sont supérieures
4. le troisième quartile du vecteur (Q3 ou quantile 7
   est la valeur coupant l'échantillon en deux : 75% d
   observations du vecteur y sont inférieures et 25% y
   supérieures
5. la valeur maximale contenue dans le vecteur (ou qua
   tile 100%) : c'est la valeur la plus élevée contenue
   la série de données.

Par exemple, toujours avec le vecteur des entiers co
entre 1 et 100 :

```r
quantile(1:100)
```

```
  0%    25%    50%    75%   100% 
  1.00  25.75  50.50  75.25 100.00 
```

L'objet obtenu est un vecteur dont chaque élément por
nom. Pour transformer cet objet en tibble, on utilise
tion enframe()

```r
enframe(quantile(1:100))
```

```
# A tibble: 5 x 2
  name  value
  <chr> <dbl>
1 0%     1   
2 25%    25.8
3 50%    50.5
4 75%    75.2
5 100%  100  
```

Il peut être très utile de calculer ces diférentes val
plusieurs variables à la fois, ou pour plusieurs sous
d'un jeu de données. Le problème est que nous ne pouvo
pas utiliser summarise() car la fonction quantile() ren-
voie pas qu'une unique valeur. Par exemple, pour calcu
quantiles des longueurs de becs pour chaque espèce de
chots, on pourrait être tenté de taper ceci :

```r
penguins |>
  summarise(Indices = quantile(bill_length_mm, na.rm = TRUE),
            .by = species)
```

```
Warning: Returning more (or less) than 1 row per `su
dplyr 1.1.0.
i Please use `reframe()` instead.
i When switching from `summarise()` to `reframe()`, 
  always returns an ungrouped data frame and adjust a
```

```
# A tibble: 15 x 2
   species Indices
   <fct>   <dbl>
 1 Adelie 32.1
 2 Adelie 36.8
```

```
 3 Adelie 38.8
 4 Adelie 40.8
 5 Adelie 46
 6 Gentoo 40.9
 7 Gentoo 45.3
 8 Gentoo 47.3
 9 Gentoo 49.6
10 Gentoo 59.6
11 Chinstrap  40.9
12 Chinstrap  46.3
13 Chinstrap  49.6
14 Chinstrap  51.1
15 Chinstrap  58
```

C'est dans ces situations que la fonction reframe() est utile. Elle joue le même rôle que summarise() dans les situation où les fonctions statistiques renvoient plus d'u
à la fois :

```r
penguins |>
  reframe(Indices = quantile(bill_length_mm, na.rm = TRUE),
          .by = species)
```

```
# A tibble: 15 x 2
  species  Indices
  <fct>    <dbl>
 1 Adelie 32.1
 2 Adelie 36.8
 3 Adelie 38.8
 4 Adelie 40.8
 5 Adelie 46
 6 Gentoo 40.9
 7 Gentoo 45.3
 8 Gentoo 47.3
 9 Gentoo 49.6
10 Gentoo 59.6
11 Chinstrap  40.9
12 Chinstrap  46.3
13 Chinstrap  49.6
14 Chinstrap  51.1
15 Chinstrap  58
```

Au contraire de summarise(), reframe() ne renvoie pas de message d'avertissement dans cette situation. Da[illegible] exemple, on ne sait malheureusement pas à quoi corr[illegible] pondent les chiffres renvoyés puisque l'informatio[illegible] quartiles a disparu (quelles valeurs correspondent[illegible] médianes ou aux premiers quartiles par exemple). Po[illegible] y remédier, on doit transformer le vecteur renvoyé [illegible] quantile() en tibble. Nous avons déjà vu comment le faire grâce à enframe(). Par ailleurs, puisque la fonction va maintenant renvoyer un tableau, on n'a[illegible] besoin de lui fournir de nom de colonnes (je retire d[illegible] Indices = de mon code) :

```
penguins |>
  reframe(enframe(quantile(bill_length_mm, na.rm = TRUE)),
          .by = species)
```

```
# A tibble: 15 x 3
   species   name  value
   <fct>     <chr> <dbl>
 1 Adelie    0%     32.1
 2 Adelie    25%    36.8
 3 Adelie    50%    38.8
 4 Adelie    75%    40.8
 5 Adelie    100%   46
 6 Gentoo    0%     40.9
 7 Gentoo    25%    45.3
 8 Gentoo    50%    47.3
 9 Gentoo    75%    49.6
10 Gentoo    100%   59.6
11 Chinstrap 0%     40.9
12 Chinstrap 25%    46.3
13 Chinstrap 50%    49.6
14 Chinstrap 75%    51.1
15 Chinstrap 100%   58
```

Enfn, comme décrit à la section 4.3.2, il est possible de modifier la forme de ce tableau avec pivot_wider() pour le lire plus facilement et éventuellement l'intégrer dans un[illegible] ou compte-rendu :

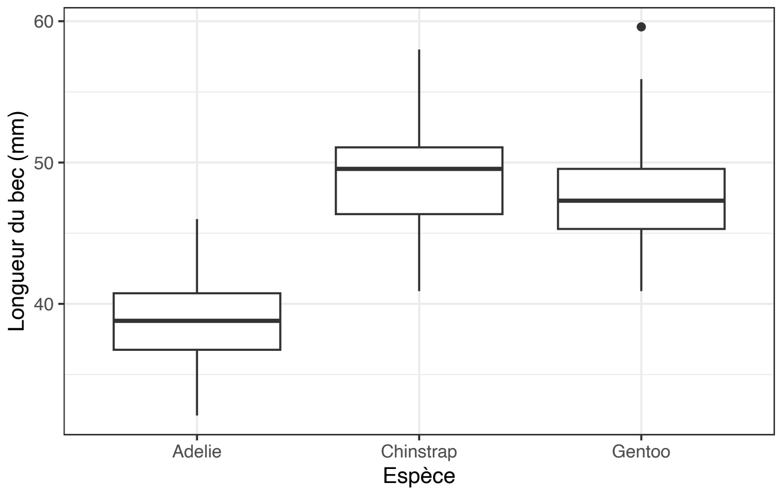
```r
penguins |>
  reframe(enframe(quantile(bill_length_mm, na.rm = TRUE)),
          .by = species) |>
  pivot_wider(names_from = species,
              values_from = value)
```

```
# A tibble: 5 x 4
  name  Adelie Gentoo Chinstrap
  <chr>  <dbl>  <dbl>     <dbl>
1 0%      32.1   40.9      40.9
2 25%     36.8   45.3      46.3
3 50%     38.8   47.3      49.6
4 75%     40.8   49.5      51.1
5 100%    46     59.6      58
```

Ces statistiques nous permettent de constater que le
chots de l'espèce Adélie semblent avoir des becs plus
que les 2 autres espèces (les 5 quantiles le confrmen
manchots Gentoo et Chinstrap ont en revanche des becs
longueur à peu près similaires, bien que ceux des Chi
soient peut-être très légèrement plus longs (Q1, méd
Q3 supérieurs à ceux des Gentoo). On peut vérifier tou
graphiquement avec des boîtes à moustaches, puisque
valeurs de quantiles sont justement celles qui sont u
pour tracer les boîtes à moustaches :

```r
penguins |>
  ggplot(aes(x = species, y = bill_length_mm)) +
  geom_boxplot() +
  labs(x = "Espèce", y = "Longueur du bec (mm)") +
  theme_bw()
```

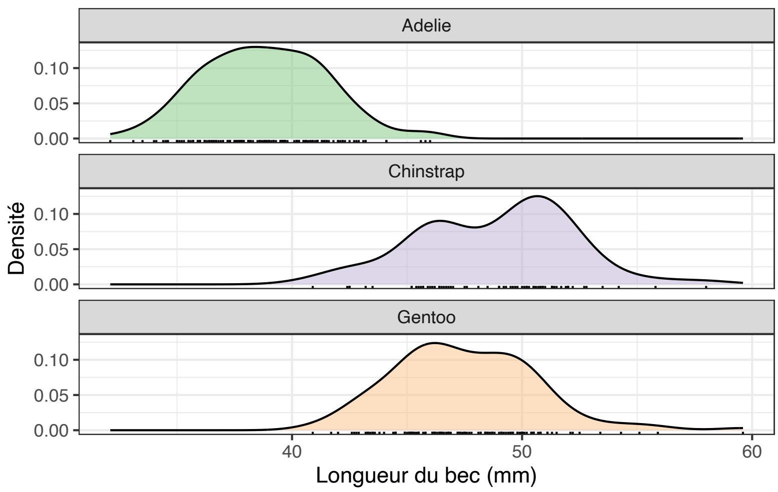
```
Warning: Removed 2 rows containing non-finite outsi
(`stat_boxplot()`).
```



Ou avec un graphique de densité :

```
penguins |>
  ggplot(aes(x = bill_length_mm, fill = species)) +
  geom_density(alpha = 0.5, show.legend = FALSE) +
  geom_rug() +
  labs(x = "Longueur du bec (mm)", y = "Densité") +
  facet_wrap(~species, ncol = 1) +
  scale_fill_brewer(palette = "Accent") +
  theme_bw()
```

```
Warning: Removed 2 rows containing non-finite outsi
(`stat_density()`).
```



À ce stade, vous devriez être capables de créer (e… d'interpréter !) ce type de graphiques. Si ce n'est… le cas, relisez d'urgence les sections 3.5, 13.5, 9.3

> **À retenir**
>
> - la fonction `summarise()` s'utilise avec des fonctions statistiques qui ne renvoient qu'une valeur (par exemple `mean()`, `median()`, `sd()`, `var()`...)
> - la fonction `reframe()` s'utilise avec des fonctions statistiques qui renvoient plusieurs valeurs (p… exemple `quantile()`, `range()`...)

Les fonctions `summarise()` et `reframe()`, avec leur argument `.by()` ou la fonction `group_by()` permettent donc de calculer n'importe quel indice de statistique descripti… tableau de données entier ou sur des modalités ou comb… sons de modalités de facteurs. Il existe par ailleurs… breuses fonctions, disponibles dans **R base** ou dans des packages spécifiques, qui permettent de fournir des r… plus ou moins automatiques de tout ou partie des varia… d'un jeu de données. Nous allons maintenant en décri… mais il en existe beaucoup d'autres : à vous d'explor… possibilités et d'utiliser les fonctions qui vous pa… plus pertinentes, les plus simples à utiliser, les pl… ou les plus complètes.

6.3 Créer des résumés de données avec la fonction `summary()`

La fonction `summary()` (à ne pas confondre avec `summarise()`) permet d'obtenir des résumés de données pour tous types d'objets dans R. Selon la classe des objets que l'on transmet à `summary()`, la nature des résultats obtenus changera. Nous verrons ainsi aux chapitres 12 et 14 que cette fonction peut être utilisée pour examiner les r… d'analyses de variances ou de modèles de régressions l… Pour l'instant, nous nous intéressons à 3 situations

1. ce que renvoie la fonction quand on lui fournit un v… teur

2. ce que renvoie la fonction quand on lui fournit un f
teur
3. ce que renvoie la fonction quand on lui fournit un t
bleau

6.3.1 Variable continue : vecteur numérique

Commençons par fournir un vecteur numérique à la fon
tion summary(). Nous allons pour cela extraire les données
de masses corporelles des manchots du tableau penguins.

```
penguins$body_mass_g
```

```
  [1] 3750 3800 3250   NA 3450 3650 3625 4675 3475 4250
 [16] 3700 3450 4500 3325 4200 3400 3600 3800 3950 38
 [31] 3250 3900 3300 3900 3325 4150 3950 3550 3300 46
 [46] 4600 3425 2975 3450 4150 3500 4300 3450 4050 29
 [61] 3150 4400 3600 4050 2850 3950 3350 4100 3050 44
 [76] 4250 3700 3900 3550 4000 3200 4700 3800 4200 33
 [91] 3550 4300 3400 4450 3300 4300 3700 4350 2900 41
[106] 3550 3750 3900 3175 4775 3825 4600 3200 4275 39
[121] 3150 3500 3450 3875 3050 4000 3275 4300 3050 40
[136] 3900 3175 3975 3400 4250 3400 3475 3050 3725 30
[151] 3700 4000 4500 5700 4450 5700 5400 4550 4800 52
[166] 5850 4200 5850 4150 6300 4800 5350 5700 5000 44
[181] 4600 5550 5250 4700 5050 6050 5150 5400 4950 52
[196] 4750 5550 4900 4200 5400 5100 5300 4850 5300 44
[211] 4450 5550 4200 5300 4400 5650 4700 5700 4650 58
[226] 5200 4700 5800 4600 6000 4750 5950 4625 5450 47
[241] 4875 5550 4950 5400 4750 5650 4850 5200 4925 48
[256] 5500 4725 5500 4700 5500 4575 5500 5000 5950 46
[271] 4925   NA 4850 5750 5200 5400 3500 3900 3650 352
[286] 3700 3800 3775 3700 4050 3575 4050 3300 3700 34
[301] 3300 4150 3400 3800 3700 4550 3200 4300 3350 41
[316] 4500 3950 3650 3550 3500 3675 4450 3400 4300 32
[331] 3350 3450 3250 4050 3800 3525 3950 3650 3650 40
```

Nous avons donc 344 valeurs de masses en grammes qui co
respondent aux 344 manchots du jeu de données. La fonc
summary() renvoie le résumé suivant lorsqu'on lui fournit
valeurs :

```r
summary(penguins$body_mass_g)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
   2700    3550    4050    4202    4750    6300       2 
```

Nous obtenons ici 7 valeurs, qui correspondent aux ci
leurs renvoyées par la fonction `quantile()` (voir Section 6.2),
ainsi que la moyenne et le nombre de valeurs manquant
Dans l'ordre, on a donc :

- la valeur minimale observée dans le vecteur. Ici,
  manchot le plus léger de l'échantillon pèse donc 2
  grammes.
- la valeur du premier quartile du vecteur. Ici, 25%
  manchots de l'échantillon (soit 86 individus) ont
  masse inférieure à 3550 grammes, et 75% des individ
  de l'échantillon (soit 258 individus) ont une mass
  périeure à 3550 grammes.
- la valeur de médiane du vecteur. La médiane est
  le deuxième quartile. Ici, 50% des manchots de
  l'échantillon (soit 172 individus) ont une mas
  inférieure à 4050 grammes, et 50% des individus
  de l'échantillon (soit 172 individus) ont une ma
  supérieure à 4050 grammes.
- la moyenne du vecteur. Ici, les manchots des 3 espèc
  du jeu de données ont en moyenne une masse 4202
  grammes.
- la valeur du troisième quartile du vecteur. Ici, 75
  manchots de l'échantillon (soit 258 individus) on
  masse inférieure à 4700 grammes, et 25% des individ
  de l'échantillon (soit 86 individus) ont une masse
  rieure à 4750 grammes.
- la valeur maximale observée dans le vecteur. Ici,
  manchot le plus lourd de l'échantillon pèse donc 63
  grammes.
- le nombre de données manquantes. Ici, 2 manchots
  n'ont pas été pesés et présentent donc la mention `NA`
  (comme Not Available) pour la variable `body_mass_g`

Utiliser la fonction `quantile()` fournit donc presque les
mêmes informations :

```r
quantile(penguins$body_mass_g, na.rm = TRUE)
```

```
  0%  25%  50%  75% 100%
2700 3550 4050 4750 6300
```

Attention, contrairement à ce que nous avons vu plus h
la fonction summary() ne possède pas d'argument by().
il n'est pas possible de l'utiliser avec la fonction group_by()
Par conséquent, il n'est pas possible de se servir de
fonction pour avoir des valeurs pour chaque modalités
facteur (pour chaque espèce par exemple).

Les diférents indices statistiques fournis nous rens
la fois sur la position de la distribution et sur la di
des données.

- La position correspond à la tendance centrale et in-
  dique quelles sont les valeurs qui caractérisent l
  grand nombre d'individus. La moyenne et la médiane
  sont les deux indices de position les plus fréquemm
  utilisés. Lorsqu'une variable a une distribution p
  ment symétrique, la moyenne et la médiane sont stri
  tement égales. Mais lorsqu'une distribution est a
  trique, la moyenne et la médiane difèrent. En parti
  lier, la moyenne est beaucoup plus sensible aux val
  extrêmes que la médiane. Cela signife que quand une
  distribution est très asymétrique, la médiane est
  vent une meilleure indication des valeurs les plus
  quemment observées.

L'histogramme de la Figure 6.1 montre la distribution de la
taille des manchots (toutes espèces confondues). Ce
tribution présente une asymétrie à droite. Cela sign
la distribution n'est pas symétrique et que la "queue
tribution" est plus longue à droite qu'à gauche. La p
des individus ont une masse comprise entre 3500 et 3
grammes, au niveau du pic principal du graphique. La
diane, en orange et qui vaut 4050 grammes est plus pro
du pic que la moyenne, en rouge, qui vaut 4202 grammes.
la diférence entre moyenne et médiane n'est pas énorme
elle peut le devenir si la distribution est vraiment t
trique, par exemple, si quelques individus seulement

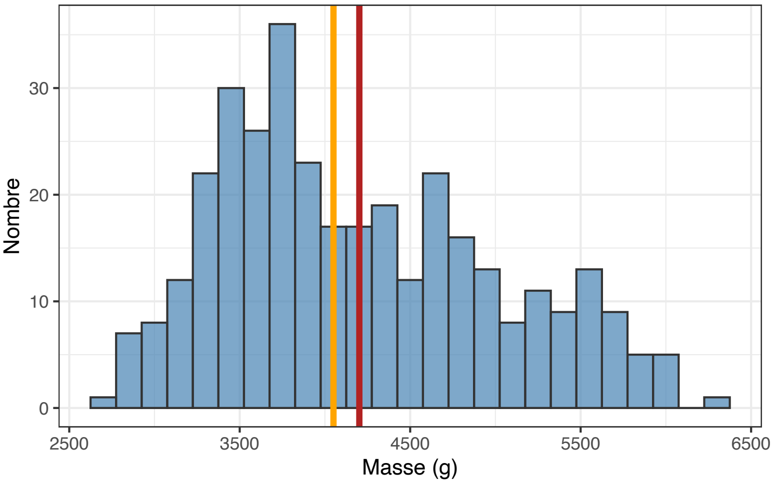


Figure 6.1: Distribution des masses corporelles des mancho

une masse supérieure à 7000 grammes, la moyenne serai
rée vers la droite du graphique alors que la médiane ne
presque pas afectée. La moyenne représenterait alors
moins fdèlement la tendance centrale.

Si l'on revient à notre `summary()`, on trouve des valeurs
proches pour la moyenne et la médiane nous indique don
degré de symétrie de la distribution.

- La dispersion des données nous renseigne sur la dis-
  persion des points autour des indices de position.
  quartiles et les valeurs minimales et maximales r
  voyées par `summary()` renseignent sur
  l'étalement des points. Les valeurs situées entre
  mier et le troisième quartile correspondent aux 50%
  valeurs de l'échantillon les plus centrales. Plus l
  entre ces quartiles (notée IQR pour "intervalle i
  quartile") sera grande, plus la dispersion sera i
  tante. D'ailleurs, lorsque la dispersion est très
  tante, les moyennes et médianes ne renseignent que f
  moyennement sur la tendance centrale. Les indices
  position sont surtout pertinents lorsque la dispe
  des points autour de cette tendance centrale n'est
  trop large. Par exemple, si la distribution des dor
  ressemblait à celle-ci (Figure 6.2), la moyenne et la mé-
  diane seraient fort peu utiles car très éloignées
  plupart des observations :

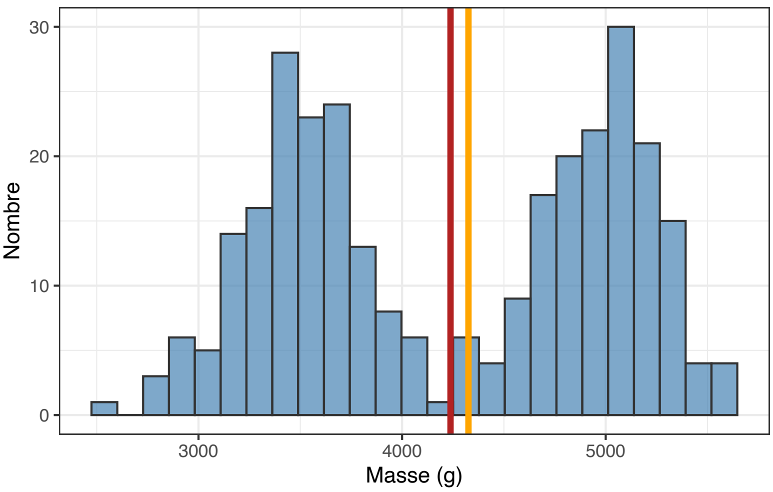


Figure 6.12: Distribution des masses corporelles (données f
tives)

On comprend donc l'importance de considérer les indic
dispersion en plus des indices de position pour carac
et comprendre une série de données numériques. L'inte
interquartile est toujours utile pour connaître l'ét
données qui correspondent aux 50% des observations le
centrales. Les autres indices de dispersion très fréq
utilisés, mais qui ne sont pas proposés par défaut par
tion summary(), sont la variance et l'écart-type. Il est pos
sible de calculer tous les indices renvoyés par la f
summary() et ceux qui nous manquent grâce à la fonction
summarise()

```
penguins |>
  summarise(min = min(body_mass_g, na.rm = TRUE),
            Q1 = quantile(body_mass_g, 0.25, na.rm = TRUE),
            med = median(body_mass_g, na.rm = TRUE),
            moy = mean(body_mass_g, na.rm = TRUE),
            Q3 = quantile(body_mass_g, 0.75, na.rm = TRUE),
            max = max(body_mass_g, na.rm = TRUE),
            var = var(body_mass_g, na.rm = TRUE),
            et = sd(body_mass_g, na.rm = TRUE))
```

```
# A tibble: 1 x 8
    min    Q1   med   moy    Q3   max     var    et
  <int> <dbl> <dbl> <dbl> <dbl> <int>   <dbl> <dbl>
```

```
1 2700 3550 4050 4202. 4750 6300 643131. 802.
```

Vous notez que le code est beaucoup plus long, et qu'ut summary() peut donc faire gagner beaucoup de temps, même si cette fonction ne nous fournit ni la variance ni l'é Mais comme souvent sous R, il est toujours possible de calculer à la main toutes ces valeurs si besoin. N'oubliez pas no qu'avec summarise(), on pourrait utiliser l'argument .by ou la fonction group_by() pour calculer très rapidement ces indices pour chaque espèce de manchot, ou pour chaque esp et chaque sexe. Comme indiqué plus haut, les fonction vantes peuvent être utilisées :

- mean() permet de calculer la moyenne.
- median() permet de calculer la médiane.
- min() et max() permettent de calculer les valeurs minimales et maximales respectivement.
- quantile() permet de calculer les quartiles. Vous notez que contrairement aux exemples de la partie précéde on utilise ici la fonction quantile() en précisant une valeur supplémentaire pour n'obtenir qu'une valeu la fois : 0.25 pour le premier quartile, et 0.75 pour le troisième.
- sd() permet de calculer l'écart-type.
- var() permet de calculer la variance.

Pour toutes ces fonctions, l'argument na.rm = TRUE permet d'obtenir les résultats même si certaines valeur manquantes. Enfin, IQR() permet de calculer l'intervalle inter-quartiles :

```r
IQR(penguins$body_mass_g, na.rm = TRUE)
```

```
[1] 1200
```

Ici, les 50% des valeurs les plus centrales de l'échant situées dans un intervalle de 1200 grammes autour de l diane.

6.3.2 Variable quantitative : facteur

Si l'on fournit une variable catégorielle à summary(), le résultat obtenu sera naturellement diférent : calcu moyennes, médianes ou quartiles n'aurait en efet pas d lorsque la variable fournie ne contient que des catégo

```
summary(penguins$species)
```

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

Pour les facteurs, summary() compte simplement le nombre d'observations pour chaque modalité. Ici, la variable species est un facteur qui compte 3 modalités. La fonction summary() nous indique donc le nombre d'individus pour chaque m lité : notre jeu de données se compose de 152 individu l'espèce Adélie, 68 individus de l'espèce Chinstrap individus de l'espèce Gentoo.

Comme pour les vecteurs numériques, si le facteur pré des données manquantes, summary() compte égale-lement leur nombre :

```
summary(penguins$sex)
```

```
female   male   NA's 
   165    168     11 
```

Pour les facteurs, la fonction summary() est donc tout à fait équivalente à la fonction count()

```
penguins |>
  count(species)
```

```
# A tibble: 3 x 2
  species       n
  <fct>     <int>
1 Adelie      152
2 Chinstrap    68
3 Gentoo      124
```

L'avantage de la fonction count() est qu'il est possible d'utiliser plusieurs facteurs pour compter le nom d'observations de toutes les combinaisons de modal (par exemple, combien d'individus de chaque sexe po chaque espèce), ce qui n'est pas possible avec la for summary()

6.3.3 Les tableaux data.frame ou tibble

L'avantage de summary() par rapport à la fonction count() apparaît lorsque l'on souhaite obtenir des infor tions sur toutes les variables d'un tableau à la fois :

```r
summary(penguins)
```

```
    species          island    bill_length_mm  bill_depth_mm
 Adelie   :152   Biscoe   :168   Min.   :32.10   Min.   :13
 Chinstrap: 68   Dream    :124   1st Qu.:39.23   1st Qu.
 Gentoo   :124   Torgersen: 52   Median :44.45   Median
                                 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.   :200
 1st Qu.:190.0     1st Qu.:3550   male  :168   1st Qu.:2
 Median :197.0     Median :4050   NA's  : 11   Median :20
 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
```

Ici, on obtient un résumé pour chaque colonne du tabl Les colonnes numériques sont traitées comme les vecte mériques (on obtient alors les minimas et maximas, les tiles, les moyennes et médianes) et les colonnes conte variables catégorielles sont traitées comme des fac on obtient alors le nombre d'observations pour chaque lité).

On constate au passage que la variable year est considérée ici comme une variable numérique, alors qu'elle devrait

être considérée comme un facteur, ce qui nous permet
de savoir combien d'individus ont été échantillonnés
année :

```r
penguins |>
  mutate(year = factor(year)) |>
  summary()
```

```
      species          island    bill_length_mm  bill_depth_mm
 Adelie   :152   Biscoe   :168   Min.   :32.10   Min.   :13
 Chinstrap: 68   Dream    :124   1st Qu.:39.23   1st Qu.:
 Gentoo   :124   Torgersen: 52   Median :44.45   Median
                                 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   2007:110
 1st Qu.:190.0     1st Qu.:3550   male  :168   2008:114
 Median :197.0     Median :4050   NA's  : 11   2009:120
 Mean   :200.9     Mean   :4202
 3rd Qu.:213.0     3rd Qu.:4750
 Max.   :231.0     Max.   :6300
 NA's   :2         NA's   :2
```

Au final, la fonction summary() est très utile dans certaines situations, notamment pour avoir rapidement accès à d
tistiques descriptives simples sur toutes les colonn
bleau. Elle reste cependant limitée car d'une part,
fournit pas les variances ou les écarts-types pour les
numériques, et il est impossible d'avoir des résumés p
pour chaque modalité d'un facteur par exemple. Ici,
rait en effet intéressant d'avoir des informations syn
concernant les mesures biométriques des manchots, e
par espèce, plutôt que toutes espèces confondues. C'
que la fonction skim() intervient.

6.4 Créer des résumés de données avec la fonction `skim()`

La fonction `skim()` fait partie du package `skimr`. Avant de pouvoir l'utiliser, pensez donc à l'installer et à le charger en mémoire si ce n'est pas déjà fait. Comme pour la fonction `summary()`, on peut utiliser `skim()` sur plusieurs types d'objets. Nous nous contenterons d'examiner ici le cas le plus fréquent : celui des tableaux, groupés avec `group_by()` ou non.

6.4.1 Tableau non groupé

Commençons par examiner le résultat avec le tableau `penguins` non groupé :

```r
skim(penguins)
```

Table 6.1: Data summary

Name	penguins
Number of rows	344
Number of columns	8
_______________________	
Column type frequency:	
factor	3
numeric	5
________________________	
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
species	0	1.00	FALSE	3	Ade: 152, Gen: 124, Chi: 68
island	0	1.00	FALSE	3	Bis: 168, Dre: 124, Tor: 52
sex	11	0.97	FALSE	2	mal: 168, fem: 165

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
bill_length_mm	2	0.99	43.92	5.46	32.1	39.23	44.45	48.5	5	
bill_depth_mm	2	0.99	17.15	1.97	13.1	15.60	17.30	18.7	2	
flipper_length_mm	2	0.99	200.92	14.06	172.0	190.00	197.00	213.0	231.	
body_mass_g	2	0.99	4201.75	801.95	2700.0	3550.00	4050.00	47		
year	0	1.00	2008.03	0.82	2007.0	2007.00	2008.00	2009.0	20	

Les résultats obtenus grâce à cette fonction sont nom
La première section nous donne des informations sur l
bleau :

- son nom, son nombre de lignes et de colonnes
- la nature des variables qu'il contient (ici 3 facte variables numériques)
- la présence de variables utilisées pour faire des r pements (il n'y en a pas encore à ce stade)

Ensuite, un bloc apporte des informations sur chaque f présent dans le tableau :

- le nom de la variable (skim_variable)
- le nombre de données manquantes (n_missing) et le taux de “données complètes” (complete_rate)
- des informations sur le nombre de modalités (n_unique)
- le nombre d'observations pour les modalités les p représentées (top_counts)

En un coup d'œil, on sait donc que 3 espèces sont prése (et on connait leurs efectifs), on sait que les manch été échantillonnées sur 3 îles, et on sait que le sexe individus sur 344 (soit 3%) est inconnu. Pour le reste presque autant de femelles que de mâles.

Le dernier bloc renseigne sur les variables numérique chaque d'elle, on a :

- le nom de la variable numérique (skim_variable)
- le nombre de données manquantes (n_missing) et le taux de “données complètes” (complete_rate)
- la moyenne (mean) et l'écart-type (sd), qui est une nouveauté par rapport à la fonction summary()

- les valeurs minimale (p0), du premier quartile (p25), de second quartile (p50, c'est la médiane !), de troisième quartile (p75) et la valeur maximale (p100)
- un histogramme très simplifié (hist) pour un premier aperçu grossier de la forme de la distribution

Là encore, en un coup d'œil, on dispose donc de toutes les informations pertinentes pour juger de la distributi position et de la dispersion de chaque variable numéri jeu de données.

6.4.2 Tableau groupé

La fonction skim(), déjà très pratique, le devient encore plus lorsque l'on choisit de lui fournir seulement c variables, et qu'on fait certains regroupements. Par on peut sélectionner les variables relatives aux dim du bec (bill_length_mm et bill_depth_mm) avec la fonction select() que nous connaissons déjà, et demander un résumé des données pour chaque espèce grâce à la fonc group_by() que nous connaissons également :

```
penguins |>                    # Avec le tableau penguins...
  select(species,
         bill_length_mm,
         bill_depth_mm) |>     # Je sélectionne les variables d
  group_by(species) |>         # Je regroupe par espèce...
  skim()                       # Et je produis un résumé des don
```

Table 6.4: Data summary

Name	group_by(...)
Number of rows	344
Number of columns	3
_______________________	
Column type frequency:	
numeric	2
________________________	
Group variables	species

Variable type: numeric

skim_variable	species	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
bill_length_mm	Adelie	1	0.99	38.79	2.66	32.1	36.75	38.80	40.75	46.	
bill_length_mm	Chinstrap	0	1.00	48.83	3.34	40.9	46.35	49.55	51.08	5	
bill_length_mm	Gentoo	1	0.99	47.50	3.08	40.9	45.30	47.30	49.55	59.6	
bill_depth_mm	Adelie	1	0.99	18.35	1.22	15.5	17.50	18.40	19.00	21.	
bill_depth_mm	Chinstrap	0	1.00	18.42	1.14	16.4	17.50	18.45	19.40	20	
bill_depth_mm	Gentoo	1	0.99	14.98	0.98	13.1	14.20	15.00	15.70	17.3	

On constate ici que pour chaque variable numérique sél née, des statistiques descriptives détaillées nous s pour chacune des 3 espèces. Ce premier examen semble m trer que :

- L'espèce Adélie est celle qui possède le bec le plus (ses valeurs de moyennes, médianes et quartiles s plus faibles que celles des 2 autres espèces).
- L'espèce Gentoo est celle qui possède le bec le plus ou le moins épais (ses valeurs de moyennes, médian et quartiles sont plus faibles que celles des 2 aut pèces)
- Il ne semble pas y avoir de fortes diférences d'éca types (donc, une dispersion comparable des points tour de leur moyenne respective) entre les 3 espèc pour chacune des 2 variables numériques, des valeu d'écarts-types comparables sont en efet observées les 3 espèces
- La distribution des 2 variables numériques semb approximativement suivre une distribution symétr pour les 3 espèces, avec une forme de courbe en cloc Les distributions devraient donc suivre à peu un distribution normale

> **ŷ Note**
>
> Vous comprenez j'espère l'importance d'examiner c genre de résumé des données avant de vous lancer dans des tests statistiques. Ils sont un complément indis sable aux explorations graphiques que vous devez éga ment prendre l'habitude de réaliser pour mieux appr hender et comprendre la nature de vos données. Puisqu chaque jeu de données est unique, vous devrez vous adapter à la situation et aux questions scientifques

vous seront posées (ou que vous vous poserez !) : les
choix qui seront pertinents pour une situation ne le
ront pas nécessairement pour une autre. Mais dans tou
les cas, pour savoir où vous allez et pour ne pas faire
bêtise au moment des tests statistiques et de leur in
prétation, vous devrez toujours explorer vos don-
nées, avec des graphiques exploratoires et des
statistiques descriptives.

6.5 Exercice

En utilisant les fonctions de résumé abordées jusqu’
tableau weather, répondez aux questions suivante :

1. Dans quel aéroport de New York les précipitations
moyennes ont-elle été les plus fortes en 2013 ?
2. Dans quel aéroport de New York la vitesse du vent
moyenne était-elle la plus forte en 2013 ? Quelle
cette vitesse ?
3. Dans quel aéroport de New York les rafales de vent
étaient-elles les plus variables en 2013 ? Quel in
statistique vous donne cette information et quell
sa valeur ?
4. Les précipitation dans les 3 aéroports de New-York
elles une distribution symétrique ?
5. Quelle est la température médiane observée en 201
tous aéroports confondus ?
6. Tous aéroports confondus, quel est le mois de l’anné
la température a été la plus variable en 2013 ? Quel
étaient les températures minimales et maximales ob
vées ce mois-là ?

7 Dispersion et incertitude

7.1 Pré-requis

Nous avons ici besoin des packages suivants :

```
library(tidyverse)
library(palmerpenguins)
library(scales)
```

Pensez à les charger en mémoire si ce n'est pas déjà fa
si vous venez de démarrer une nouvelle session de trav

Le package scales (Wickham, Pedersen, et Seidel 2023)
contient de très nombreuses fonctions particulièr
utiles pour améliorer l'aspect des légendes d'axes
exemple, pour transformer des chiffres compris entre
en pourcentages, ou pour ajouter le symbole d'une de
quand l'axe renseigne sur des montants en euros ou do
par exemple).

7.2 La notion de dispersion

Comme expliqué à la section 6.3.1, les indices de disper-
sion nous renseignent sur la variabilité des données a
de la valeur centrale (moyenne ou médiane) d'une popul
ou d'un échantillon. L'écart-type, la variance et l'i
terquartile sont 3 exemples d'indices de dispersion.
l'exemple de l'écart-type. Un écart-type faible ind
la majorité des observations ont des valeurs proches
moyenne. À l'inverse, un écart-type important indiqu
la plupart des points sont éloignés de la moyenne. L'
type est une caractéristique de la population que l'o
grâce à un échantillon, au même titre que la moyenne.
travaillant sur un échantillon, on espère accéder au
grandeurs de la population. Même si ces vraies grand

sont à jamais inaccessibles (on ne connaîtra jamais
tement quelle est la vraie valeur de moyenne ou d'écart-
type de la population), on espère qu'avec un échantill
nage réalisé correctement, la moyenne de l'échantill
$\bar{x}$) et l'écart-type de l'échantillon (s) refètent assez fdèle-
ment les valeurs de la population générale. C'est la
d'estimateur, intimement liée à la notion d'inférence sta-
tistique : la moyenne de l'échantillon, que l'on connait av
précision, est une estimation de la moyenne de la population
qui restera à jamais inconnue. C'est la raison pour la
la moyenne de l'échantillon est parfois notée $\hat{\mu}$ (plutôt que
$\bar{x}$ ou). De même, l'écart-type et la variance d'un
échantillon sont des estimateurs de l'écart-type et de la
variance σ^2 de la population générale. C'est la raison pour
laquelle on les note parfois $\hat{\sigma}$ et $\hat{\sigma}^2$ respectivement. L'accent
circonflexe se prononce “chapeau” (sigma chapeau). On dit donc que
chapeau, l'écart-type de l'échantillon) est un estim
l´écart-type de la population générale. Comme nous l
vu, les indices de dispersion doivent accompagner les
de position lorsque l'on décrit des données, car prése
valeur de moyenne, ou de médiane seule n'a pas de sens
faut savoir à quel point les données sont proches ou él
de la tendance centrale pour savoir si, dans la popula
nérale, les indicateurs de position correspondent ou
valeurs portées par la majorité des individus.

Nous avons vu au Chapitre 6 comment calculer des indices de position et de dispersion. Tout ceci devrait donc ê
pour vous à ce stade.

7.3 La notion d'incertitude

Par ailleurs, puisqu'on ne sait jamais avec certitud
estimations (de moyennes ou d'écarts-types ou de tout
paramètre) refètent fdèlement ou non les vraies valeu
la population, nous devons quantifer à quel point nos
mations s'écartent de celles de la population généra
tout l'intérêt des statistiques et c'est ce que perme
indices d'incertitude. On ne connaîtra jamais la vraie va-
leur de moyenne ou d'écart-type de la population, mai
peut quantifer à quel point nos estimations (basées s
échantillon) sont précises ou imprécises.

Les deux indices d'incertitude les plus connus (et l
utilisés) sont :

1. l'intervalle de confiance (à 95% de la moyenne ou de tout autre estimateur ; les formules sont nombre et il n'est pas utile de les détailler ici : nous ve comment les calculer plus bas)
2. l'erreur standard de la moyenne (), dont la formule est la suivante :

$$ = \frac{}{\sqrt{}}$$

avec l'écart-type de l'échantillon et la taille de l'échantillon.

Comme pour la moyenne, on peut calculer l'erreur stan d'un écart-type, d'une médiane, d'une proportion, tout autre estimateur calculé sur un échantillon. Ce d'incertitude ne nous renseigne pas sur une grandeu la population générale qu'on chercherait à estimer bien sur l'incertitude associée à une estimation qu faisons en travaillant sur un échantillon de taille f limitée. Tout processus d'échantillonnage est for entaché d'incertitude, causée entre autre par le has l'échantillonnage (ou fuctuation d'échantillonnage nous travaillons sur des échantillons forcément imp les indices d'incertitude vont nous permettre de qua à quel point nos estimations s'écartent des vraies v de la population. Ces “vraies valeurs”, faute de po collecter tous les individus de la population, rest jamais inconnues.

> **Autrement dit…**
>
> Quand on étudie des populations naturelles grâce à de échantillons, on se trompe toujours. Les statistiques nous permettent de quantifier à quel point on se trompe grâce aux indices d'incertitude, et c'est déjà pas mal !

En examinant la formule de l'erreur standard de la moy présentée ci-dessus, on comprend intuitivement que taille de l'échantillon augmente, plus l'erreur standard

(donc l'incertitude) associée à notre estimation de mo
minue. Autrement dit, plus les données sont abondante
l'échantillon, meilleure sera notre estimation de mo
donc, moins le risque de raconter des bêtises sera gra

L'autre indice d'incertitude très fréquemment util
l'intervalle de confance à 95% (de la moyenne, de la méd
de la variance, ou de tout autre estimateur calculé da
échantillon). L'intervalle de confance nous renseig
gamme des valeurs les plus probables pour un paramèt
de la population étudiée. Par exemple, si j'observe
un échantillon, une moyenne de 10, avec un intervall
confance calculé de [7 ; 15], cela signife que, dans la
tion générale, la vraie valeur de moyenne a de bonnes ch
de se trouver dans l'intervalle [7 ; 15]. Dans la popu
générale, toutes les valeurs comprises entre 7 et 15
vraisemblables pour la moyenne alors que les valeurs
en dehors de cet intervalle sont moins probables. Une
façon de comprendre l'intervalle de confance est de
que si je récupère un grand nombre d'échantillons dan
même population, en utilisant exactement le même prot
expérimental, 95% des échantillons que je vais récu
auront une moyenne située à l'intérieur de l'interva
confance à 95%, et 5% des échantillons auront une moye
située à l'extérieur de l'intervalle de confance à 95
une notion qui n'est pas si évidente que ça à comprend
donc prenez bien le temps de relire cette section si be
de poser des questions le cas échéant.

Concrètement, plus l'intervalle de confance est larg
notre confance est grande. Si la moyenne d'un échanti
vaut $\bar{x} = 10$, et que son intervalle de confance à 95% vaut
[9,5 ; 11], la gamme des valeurs probables pour la moy
de la population est étroite. Autrement dit, la moyen
l'échantillon (10), a de bonne chances d'être très pro
vraie valeur de moyenne de la population générale (vr
blablement comprise quelque part entre 9,5 et 11). À l'
si l'intervalle de confance à 95% de la moyenne vaut [4
la gamme des valeurs possibles pour la vraie moyenne d
population est grande. La moyenne de l'échantillon au
de grandes chances d'être assez éloignée de la vraie v
la population.

La notion d'intervalle de confance à 95% est donc très p
de celle d'erreur standard. D'ailleurs, pour de nombr

mètres, l'intervalle de confance est calculé à partir
standard.

> ÿ À retenir !
>
> Les paramètres de position et de dispersion sont des caractéristiques des populations que l'on étudie. O
> père pouvoir les estimer correctement grâce aux écha
> tillons dont on dispose.
>
> Les indices d'incertitude ne sont pas des paramètres ou des caractéristiques des populations que l'on étudi
> ne cherche donc pas à les estimer. Ils nous permetter
> en revanche de quantifer à quel point on se trompe en cherchant à estimer des paramètres de la population g
> nérale à partir d'estimateurs calculés sur un échant
> Autrement dit, calculer des estimateurs de position et de dispersion permet d'apprendre des choses au sujet des populations étudiées. Calculer des indi
> d'incertitude ne permet pas d'apprendre quoi que ce soit sur les populations étudiées, mais permet
> d'apprendre des choses sur la qualité de notre trava
> d'échantillonnage et d'estimation. En général, les
> dices d'incertitudes nous fournissent une gamme de v
> leurs au sein desquelles les vraies valeurs des param
> des populations générales ont de bonnes chances de s
> trouver. Si on a bien travaillé, et qu'on dispose de b
> coup de données, ces gammes de valeurs seront peu éten
> dues. Si à l'inverse, nous ne disposons que de trop pe
> de données, ces gammes de valeurs seront très étendue

7.4 Calcul de l'erreur standard de la moyenne

Contrairement aux indices de position et de dispersi
n'existe pas de fonction prédéfinie dans R qui permette de calculer
l'erreur standard de la moyenne. Toutefois, sa formu
simple nous permet de la calculer à la main quand on en
besoin grâce à la fonction summarise()

Par exemple, reprenons les données de masse corporel
3 espèces de manchots dans le tableau penguins

que nous souhaitions étudier les diférences de masse
porelles des 3 espèces, en tenant compte du sexe des
vidus. Pour chaque espèce et chaque sexe, nous allons
culer la masse moyenne des individus en grammes (vari
body_mass_g). Nous prendrons soin au préalable d'éliminer
les lignes pour lesquelles le sexe des individus est i

```r
penguins |>
  filter(!is.na(sex)) |>
  summarise(moyenne = mean(body_mass_g, na.rm = TRUE),
            .by = c(species, sex))
```

```
# A tibble: 6 x 3
  species   sex    moyenne
  <fct>     <fct>    <dbl>
1 Adelie    male     4043.
2 Adelie    female   3369.
3 Gentoo    female   4680.
4 Gentoo    male     5485.
5 Chinstrap female   3527.
6 Chinstrap male     3939.
```

Pour pouvoir réutiliser ce tableau, je lui donne un no

```r
masses <- penguins |>
  filter(!is.na(sex)) |>
  summarise(moyenne = mean(body_mass_g, na.rm = TRUE),
            .by = c(species, sex))
```

Au fnal, on peut faire un graphique avec ces donnée
Puisqu'on dispose d'une variable numérique et 2 vari
catégorielles, je fais le cjoix de produire un diag
bâtons facettés :

```r
masses |>
  ggplot(aes(x = sex, y = moyenne, fill = species)) +
  geom_col(color = "grey20", alpha = 0.5) +
  facet_wrap(~species, ncol = 1) +
  labs(x = "", y = "Masse moyenne (g)", fill = "Espèce") +
  theme_bw() +
  scale_fill_brewer(palette = "Accent") +
  scale_y_continuous(labels = number_format())
```

```r
  scale_x_discrete(labels = c("Femelles", "Mâles"))
```

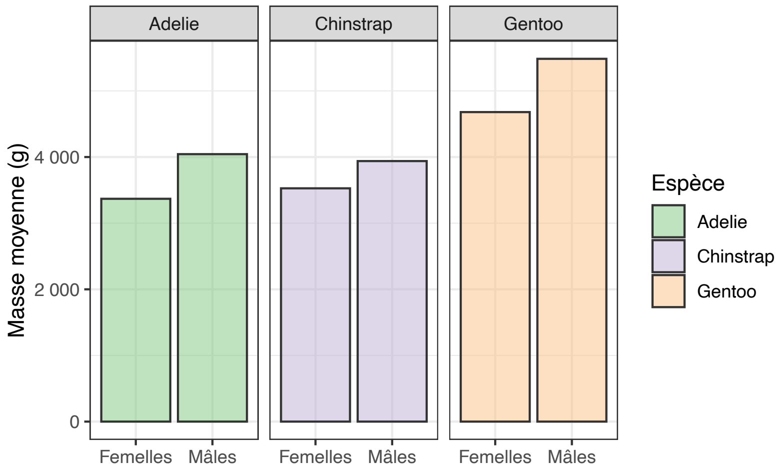


Figure 7.1: Comparaison des tailles moyennes observées chez les mâles et les femelles de 3 espèces de manchots

Vous remarquerez que :

1. J'utilise geom_col() et non geom_bar() car j'ai déjà calculé manuellement la variable que je souhaite associer à l'axe des ordonnées
2. J'associe la couleur de remplissage à l'espèce, bien que ça ne soit pas indispensable puisque je fais un sous-graphique par espèce. Cela rend toutefois le graphique plus agréable et en facilite la lecture.
3. Je modifie 3 échelles :

- avec scale_fill_brewer(), je change la palette de couleur utilisée pour le remplissage des barres
- avec scale_y_continuous(), je modifie l'échelle (continue) de l'axe des ordonnées. Je fais appel à la fonction number_format() du package scales afin d'ajouter un séparateur des milliers aux chiffres de l'axe.
- avec scale_x_discrete(), je modifie les termes qui apparaissent sur l'axe des abscisses (qui est un axe discontinu, ou catégoriel, ou facteur). Les catégories female et male de la variable sex sont transformées en Femelles et Mâles respectivement

Puisque chaque barre de la Figure 7.1 ne correspond qu'à une valeur de moyenne, il nous faut impérativement représenter également associée à ces calculs de moyenne. Pour ce faire, nous devons calculer l'erreur standard des moyennes :

```
penguins |>
  filter(!is.na(sex)) |>
  summarise(moyenne = mean(body_mass_g, na.rm = TRUE),
            N_obs = n(),
            erreur_standard = sd(body_mass_g, na.rm = TRUE) / sqrt(N_obs),
            .by = c(species, sex))
```

```
# A tibble: 6 x 5
  species   sex    moyenne N_obs erreur_standard
  <fct>     <fct>    <dbl> <int>           <dbl>
1 Adelie    male     4043.    73            40.6
2 Adelie    female   3369.    73            31.5
3 Gentoo    female   4680.    58            37.0
4 Gentoo    male     5485.    61            40.1
5 Chinstrap female   3527.    34            48.9
6 Chinstrap male     3939.    34            62.1
```

Notre tableau de statistiques descriptives possède maintenant 2 colonnes supplémentaires : le nombre d'observations (que j'ai nommé N_obs) et l'erreur standard associée à chaque moyenne, calculée grâce à la formule vue plus haut (la fonction sqrt() permet de calculer la racine carrée). On constate que l'erreur standard, qui s'exprime dans la même unité que la moyenne, varie du simple au double selon les groupes. Ainsi, pour les mâles de l'espèce Chinstrap, l'incertitude est deux fois plus importante que pour les femelles de l'espèce Adélie. Cela est probablement dû en partie à des différences de tailles d'échantillons importantes entre ces 2 catégories (73 femelles Adélie contre 34 mâles Chinstrap), mais ça n'est certainement pas la seule explication. Sinon, les femelles Chinstrap auraient elles aussi une incertitude plus grande. L'incertitude reflète aussi, de façon indirecte, la variabilité de la variable étudiée.

Une fois de plus, je donne un nom à ce tableau de données pour pouvoir le réutiliser plus tard :

```
masses_se <- penguins |>
  filter(!is.na(sex)) |>
  summarise(moyenne = mean(body_mass_g, na.rm = TRUE),
```

```
            N_obs = n(),
            erreur_standard = sd(body_mass_g, na.rm = TRUE) / sqrt(N_obs),
            .by = c(species, sex))
```

Notez que le package ggplot2 contient une fonction permettant de calculer à la fois la moyenne et erreur standa[rd de] la moyenne d'un échantillon : mean_se(). Puisque cette fonction renvoie 3 valeurs, on utilise reframe() :

```
penguins |>
  filter(!is.na(sex)) |>
  reframe(mean_se(body_mass_g),
          .by = c(species, sex))
```

```
# A tibble: 6 x 5
  species   sex        y  ymin  ymax
  <fct>     <fct>  <dbl> <dbl> <dbl>
1 Adelie    male   4043. 4003. 4084.
2 Adelie    female 3369. 3337. 3400.
3 Gentoo    female 4680. 4643. 4717.
4 Gentoo    male   5485. 5445. 5525.
5 Chinstrap female 3527. 3478. 3576.
6 Chinstrap male   3939. 3877. 4001.
```

Les résultats obtenus ne sont pas exactement au même fo[rmat] :

- la colonne y contient les valeurs de moyennes ($\bar{x}$)
- la colonne ymin contient la valeur de moyenne moins une fois l'erreur standard ($\bar{x} - se$)
- la colonne ymax contient la valeur de moyenne plus une fois l'erreur standard ($\bar{x} + se$)

Notez également que contrairement aux fonctions mean() ou sd(), la fonction mean_se() n'a pas besoin qu'on lui précise na.rm = TRUE : par défaut, elle ignore les valeurs manquantes.

Il ne nous restera plus qu'à ajouter des barres d'er[reur] sur notre graphique pour visualiser l'incertitude a[ssociée à] chaque valeur de moyenne. C'est ce que nous verrons a[u] Chapitre 8.

7.5 Calculs d'intervalles de confance à 95%

Comme pour les erreurs standard, il est possible de c
ler des intervalles de confance de n'importe quel est
calculé à partir d'un échantillon, pour déterminer la
des valeurs les plus probables pour les paramètres équ
dans la population générale. Nous nous concentrerons
le calcul des intervalles de confance à 95% de la moye
mais nous serons amenés à examiner également l'inter
de confance de la médiane, puis au Chapitre 11, l'
de confance à 95% d'une diférence de moyennes.

Contrairement à l'erreur standard, il n'y a pas qu'
bonne façon de calculer l'intervalle de confance à
d'une moyenne. Plusieurs formules existent et le choi
formule dépend en partie de la distribution des donné
distribution suit-elle une loi Normale ou non) et de l
de l'échantillon (est-il supérieur à 30
ou non ?). Dans la situation idéale d'une variable qui
distribution Normale, les bornes inférieures et supé
l'intervalle de confance à 95% sont obtenues grâce à
formule

$$\bar{x} - 1.96 \frac{s}{\sqrt{n}} < \mu < \bar{x} + 1.96 \frac{s}{\sqrt{n}}$$

Autrement dit, la vraie moyenne d'une population a de
bonnes chances de se trouver dans un intervalle de plu
moins 1.96 fois l'erreur standard de la moyenne. En pr
approximation, l'intervalle de confance est donc la m
de l'échantillon plus ou moins 2 fois l'erreur standard (que
nous avons appris à calculer à la main un peu plus tôt).
peut donc calculer à la main les bornes inférieures e
rieures de l'intervalle de confance ainsi :

```
penguins |>
  filter(!is.na(sex)) |>
  reframe(mean_se(body_mass_g, mult = 1.96),
          .by = c(species, sex))
```

```
# A tibble: 6 x 5
  species   sex   y   ymin   ymax
```

```
  <fct>     <fct>  <dbl> <dbl> <dbl>
1 Adelie    male    4043. 3964. 4123.
2 Adelie    female  3369. 3307. 3431.
3 Gentoo    female  4680. 4607. 4752.
4 Gentoo    male    5485. 5406. 5563.
5 Chinstrap female  3527. 3431. 3623.
6 Chinstrap male    3939. 3817. 4061.
```

Ici, grâce à l'argument mult = 1.96 de la fonction mean_se()

- la colonne ymin contient maintenant les valeurs de moyennes moins 1.96 fois l'erreur standard
- la colonne ymax contient maintenant les valeurs de moyennes plus 1.96 fois l'erreur standard

Dans la pratique, puisque cette méthode reste approxi
et dépend de la nature des données dont on dispose, o
utilisera plutôt des fonctions spécifques qui calc
pour nous les intervalles de confance à 95% de nos es
mateurs. C'est ce que permet en particulier la fonc
mean_cl_normal() du package ggplot2. Il est toutefois
important de bien comprendre qu'il y a un lien étro
entre l'erreur standard (l'incertitude associées à l
d'un paramètre d'une population à partir des donné
d'un échantillon), et l'intervalle de confance à 95%
paramètre.

```r
penguins |>
  filter(!is.na(sex)) |>
  reframe(mean_cl_normal(body_mass_g),
          .by = c(species, sex))
```

```
# A tibble: 6 x 5
  species   sex        y  ymin  ymax
  <fct>     <fct>  <dbl> <dbl> <dbl>
1 Adelie    male    4043. 3963. 4124.
2 Adelie    female  3369. 3306. 3432.
3 Gentoo    female  4680. 4606. 4754.
4 Gentoo    male    5485. 5405. 5565.
5 Chinstrap female  3527. 3428. 3627.
6 Chinstrap male    3939. 3813. 4065.
```

Comme dans les tableaux précédents, 3 nouvelles colo
ont été crées :

- `y` contient toujours la moyenne des températures men
suelles pour chaque aéroport
- `ymin` contient maintenant les bornes inférieures de l'intervalle à 95% des moyennes
- `ymax` contient maintenant les bornes supérieures de l'intervalle à 95% des moyennes

Pour que la suite soit plus claire, nous allons a cher
ner des noms à ces diférents tableaux en prenant soin
renommer les colonnes pour plus de clarté.

Tout d'abord, nous disposons du tableau **masses_se**, qui
contient, les masses moyennes des mâles et des femell
3 espèces, et les erreurs standard de ces moyennes :

```
masses_se
```

```
# A tibble: 6 x 5
  species   sex    moyenne N_obs erreur_standard
  <fct>     <fct>    <dbl> <int>           <dbl>
1 Adelie    male     4043.    73            40.6
2 Adelie    female   3369.    73            31.5
3 Gentoo    female   4680.    58            37.0
4 Gentoo    male     5485.    61            40.1
5 Chinstrap female   3527.    34            48.9
6 Chinstrap male     3939.    34            62.1
```

Ensuite, nous avons produit un tableau presque équiv
que nous allons nommer `masses_se_bornes` pour lequel
nous allons modifier le nom des colonnes `y`, `ymin` et `ymax` :

```
masses_se_bornes <- penguins |>
  filter(!is.na(sex)) |>
  reframe(mean_se(body_mass_g),
          .by = c(species, sex)) |>
  rename(moyenne = y,
         moyenne_moins_se = ymin,
         moyenne_plus_se = ymax)

masses_se_bornes
```

```
# A tibble: 6 x 5
  species   sex    moyenne moyenne_moins_se moyenne_p
  <fct>     <fct>    <dbl>            <dbl>      <dbl>
1 Adelie    male     4043.            4003.      4084.
2 Adelie    female   3369.            3337.      3400.
3 Gentoo    female   4680.            4643.      4717.
4 Gentoo    male     5485.            5445.      5525.
5 Chinstrap female   3527.            3478.      3576.
6 Chinstrap male     3939.            3877.      4001.
```

Nous avons ensuite calculé manuellement des interval confiance approximatifs avec la fonction mean_se() et son argument mult = 1.96. Là encore, nous allons stocker cet objet dans un tableau nommé masses_ci_approx, et nous allons modifier le nom des colonnes y, ymin, ymax :

```
masses_ci_approx <- penguins |>
  filter(!is.na(sex)) |>
  reframe(mean_se(body_mass_g, mult = 1.96),
          .by = c(species, sex)) |>
  rename(moyenne = y,
         ci_borne_inf = ymin,
         ci_borne_sup = ymax)

masses_ci_approx
```

```
# A tibble: 6 x 5
  species   sex    moyenne ci_borne_inf ci_borne_sup
  <fct>     <fct>    <dbl>        <dbl>        <dbl>
1 Adelie    male     4043.        3964.        4123.
2 Adelie    female   3369.        3307.        3431.
3 Gentoo    female   4680.        4607.        4752.
4 Gentoo    male     5485.        5406.        5563.
5 Chinstrap female   3527.        3431.        3623.
6 Chinstrap male     3939.        3817.        4061.
```

Enfin, nous avons calculé les intervalles de confiance une fonction spécialement dédiée à cette tâche : la f mean_cl_normal(). Nous allons stocker cet objet dans un tableau nommé masses_ci et nous allons modifier le nom des colonnes y, ymin, ymax :

```r
masses_ci <- penguins |>
  filter(!is.na(sex)) |>
  reframe(mean_cl_normal(body_mass_g),
          .by = c(species, sex)) |>
  rename(moyenne = y,
         ci_borne_inf = ymin,
         ci_borne_sup = ymax)

masses_ci
```

```
# A tibble: 6 x 5
  species   sex    moyenne ci_borne_inf ci_borne_sup
  <fct>     <fct>    <dbl>        <dbl>        <dbl>
1 Adelie    male     4043.        3963.        4124.
2 Adelie    female   3369.        3306.        3432.
3 Gentoo    female   4680.        4606.        4754.
4 Gentoo    male     5485.        5405.        5565.
5 Chinstrap female   3527.        3427.        3627.
6 Chinstrap male     3939.        3813.        4065.
```

Maintenant, si l'on compare les 2 tableaux contenant
culs d'intervalles de confiance de la moyenne, on const
les résultats sont très proches :

```r
masses_ci_approx
masses_ci
```

```
# A tibble: 6 x 5
  species   sex    moyenne ci_borne_inf ci_borne_sup
  <fct>     <fct>    <dbl>        <dbl>        <dbl>
1 Adelie    male     4043.        3964.        4123.
2 Adelie    female   3369.        3307.        3431.
3 Gentoo    female   4680.        4607.        4752.
4 Gentoo    male     5485.        5406.        5563.
5 Chinstrap female   3527.        3431.        3623.
6 Chinstrap male     3939.        3817.        4061.
```

```
# A tibble: 6 x 5
  species   sex    moyenne ci_borne_inf ci_borne_sup
  <fct>     <fct>    <dbl>        <dbl>        <dbl>
1 Adelie    male     4043.        3963.        4124.
2 Adelie    female   3369.        3306.        3432.
3 Gentoo    female   4680.        4606.        4754.
4 Gentoo    male     5485.        5405.        5565.
5 Chinstrap female   3527.        3427.        3627.
6 Chinstrap male     3939.        3813.        4065.
```

Les bornes inférieures et supérieures des interval
confiance à 95% des moyennes ne sont pas égales quand o
les calcule manuellement de façon approchée et quand o
calcule de façon exacte, mais les diférences sont min

8 Visualiser l'incertitude et la dispersion

8.1 Pré-requis

Nous avons ici besoin des packages suivants :

```r
library(tidyverse)
library(palmerpenguins)
library(nycflights13)
library(scales)
library(colorspace)
```

Le package colorspace (Ihaka et al. 2024) permet d'utiliser de très nombreuses palettes de couleurs. Ces palettes nombreux avantages (dont la propriété de permettre au toniens de distinguer correctement les diférentes cou palettes catégorielles). N'hésitez pas à consulter le site dédié à ce package qui est très complet et présente de nombreux exemples.

Nous aurons aussi besoin des tableaux créés au Chapit 7 (masses, masses_se, bornes_ci, masses_ci_approx et masses_ci).

Donc pensez bien à charger en mémoire les packages et à lancer les commandes de vos scripts qui vous ont permi créer ces tableaux s'ils ne sont plus en mémoire dans environnement de travail. Il existe plusieurs façons senter visuellement les positions, les dispersions et les incertitudes. Concernant les positions et les dispers d'abord, nous avons déjà vu plusieurs façons de faire a pitre 3, en particulier dans les parties consacrées aux h grammes, aux stripcharts et aux boxplots. Nous repre ici brièvement chacun de ces 3 types de graphique afn d remettre en contexte avec ce que nous avons appris ici

Dans un dernier temps, nous verrons comment visuali l'incertitude associée à des calculs de moyennes ou de variances grâce aux barres d'erreurs ou aux encoches boîtes à moustaches.

8.2 Position et dispersion : les histogrammes

Je vous renvoie à la section 3.5.1 si vous avez besoin de vous rafraîchir la mémoire. Vous pouvez aussi jeter aussi u la partie sur les histogrammes facettés, section 3.9.1

Les histogrammes permettent de déterminer à la fois o trouvent les valeurs les plus fréquemment observées (tion du pic principal correspond à la tendance central dispersion (ou variabilité) des valeurs autour de la centrale. Par exemple, la fonction facet_grid() permet de faire des histogrammes des températures pour chaque port de New York et chaque mois de l'année 2013 :

```
penguins |>
  filter(!is.na(sex)) |>
  ggplot(aes(x = body_mass_g, fill = species)) +
  geom_histogram(bins = 20, color = "grey20", alpha = 0.5, show.legend = FALSE) +
  facet_grid(species ~ sex, scales = "free_y") +
  labs(x = "Masse corporelle (g)", y = "Fréquence") +
  scale_fill_brewer(palette = "Accent") +
  scale_x_continuous(labels = number_format()) +
  theme_bw()
```

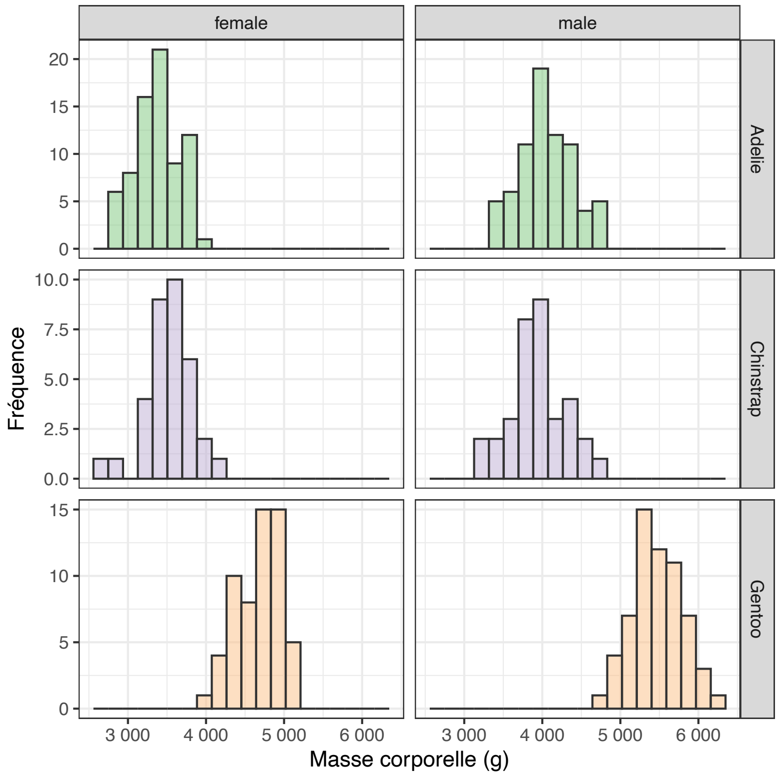


Figure 8.1: Distribution des masses corporelles chez les mâ
et les femelles de 3 espèces de manchots

Ici, 6 histogrammes sont produits. Ils permettent de c
que :

- les masses sont à peu près toutes distribuées selon
courbe en cloche
- les masses moyennes sont plus élevées chez les Gent
que chez les deux autres espèces. C'est bien la posi
des pics sur l'axe des abscisses qui nous renseign
dessus.
- pour chaque espèce, les masses moyennes sont globa
ment plus élevées chez les mâles que chez les femel
Par exemple, pour l'espèce Adélie, le pic se situe a
de 3500 grammes pour les femelles, et autour de 400
grammes pour les mâles.
- la variabilité des masses est comparable pour les 3
pèces et les 2 sexes. Cette fois, c'est l'étalemen
histogrammes qui nous renseigne sur la dispersion
l'étalement est toujours d'environ 1500 grammes,

peut-être pour les femelles Gentoo dont l'étalemen
d'environ 1000 grammes.

8.3 Position et dispersion : les stripch

Une autre façon de visualiser à la fois les tendances c
et les dispersions consiste à produire un nuage de p
"stripchart". Là encore, je vous renvoie à la partie su
charts, 3.9.2 si vous avez besoin de vous rafraîchir la mémoire.

```
penguins |>
  filter(!is.na(sex)) |>
  ggplot(aes(x = sex, y = body_mass_g, fill = sex)) +
  geom_jitter(shape = 21, color = "grey20", show.legend = FALSE,
              width = 0.15, height = 0,
              alpha = 0.5) +
  facet_wrap(~species, ncol = 1, scales = "free_y") +
  labs(x = "Sexe", y = "Masse corporelle (g)") +
  scale_fill_brewer(palette = "Accent") +
  scale_y_continuous(labels = number_format()) +
  theme_bw()
```

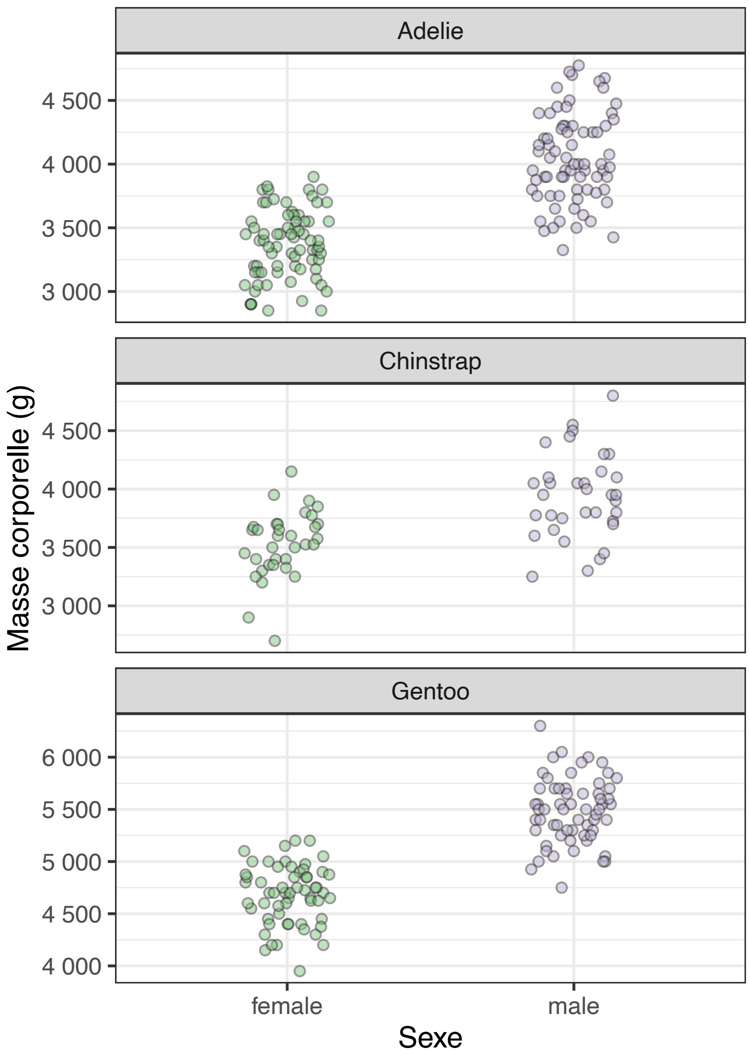


Figure 8.12: Distribution des masses corporelles chez les mâ
et les femelles de 3 espèces de manchots

Cette fois, nous visualisons la totalité des données d
et non les données regroupées dans des classes plus ou
arbitraires. Mais là encore, on peut facilement comp
position de chaque série de données : pour chaque esp
les mâles ont des masses corporelles plus importantes
femelles, et globalement, les Gentoo ont des masses c
relles plus élevées que les autres espèces. La disper
données est aussi facile à comparer entre les groupes
ici l'étendue du nuage de points sur l'axe des ordonné
nous permet de le faire.

Enfin, les stripcharts facettés sont particulièreme
lorsque le nombre de séries de données est grand. P
exemple, dans le package nycflights13 la table weather
contient des données météo enregistrées toutes les
de l'année 2013. Si l'on souhaite comparer l'évoluti
températures mensuelles dans chacun des 3 aéroports
New York, voilà ce qu'on peut faire :

```
weather |>
  mutate(temp_celsius = (temp - 32) / 1.8) |>
  ggplot(aes(x = factor(month), y = temp_celsius, fill = temp_celsius)) +
  geom_jitter(shape = 21, color = "grey20", alpha = 0.5,
              width = 0.15, height = 0, show.legend = FALSE) +
  facet_wrap(~origin, ncol = 1) +
  scale_fill_continuous_diverging(palette = "Blue-Red 3") +
  labs(x = "Mois", y = "Températures (°C)") +
  theme_bw()
```

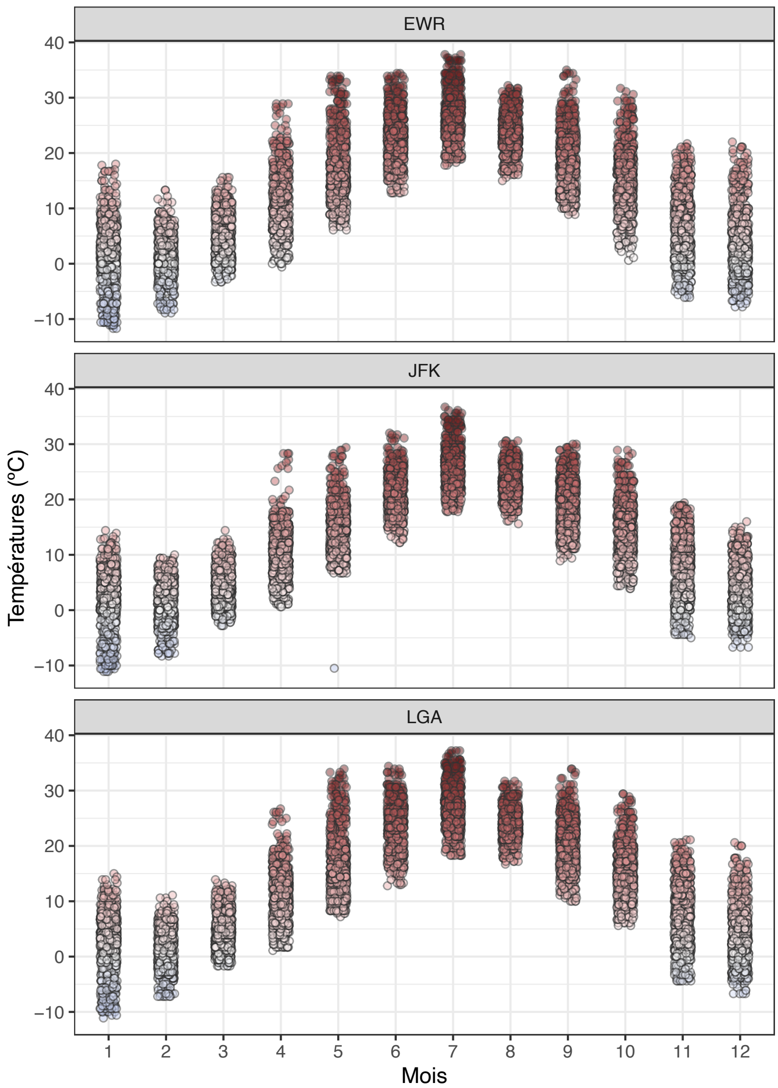


Figure 8.3: Distribution des températures mensuelles dans l
3 aéroports de New York en 2013

8.4 Position et dispersion : les boxplot

La dernière façon classique de visualiser à la fois les
centrales et les dispersions consiste à produire un g
boîtes à moustaches, ou “boxplot”. Là encore, je vous r
à la partie sur les boxplots 3.9.3 si vous avez besoin
de vous rafraîchir la mémoire. Les boîtes à moustache
également très pratiques pour comparer de nombreux gr

les uns avec les autres. Avec les données de températ
voilà à quoi ça ressemble :

```
weather |>
  mutate(temp_celsius = (temp - 32) / 1.8) |>
  ggplot(aes(x = factor(month), y = temp_celsius, fill = factor(month))) +
  geom_boxplot(show.legend = FALSE, alpha = 0.5) +
  facet_wrap(~origin, ncol = 1) +
  labs(x = "Mois", y = "Températures (°C)") +
  theme_bw()
```

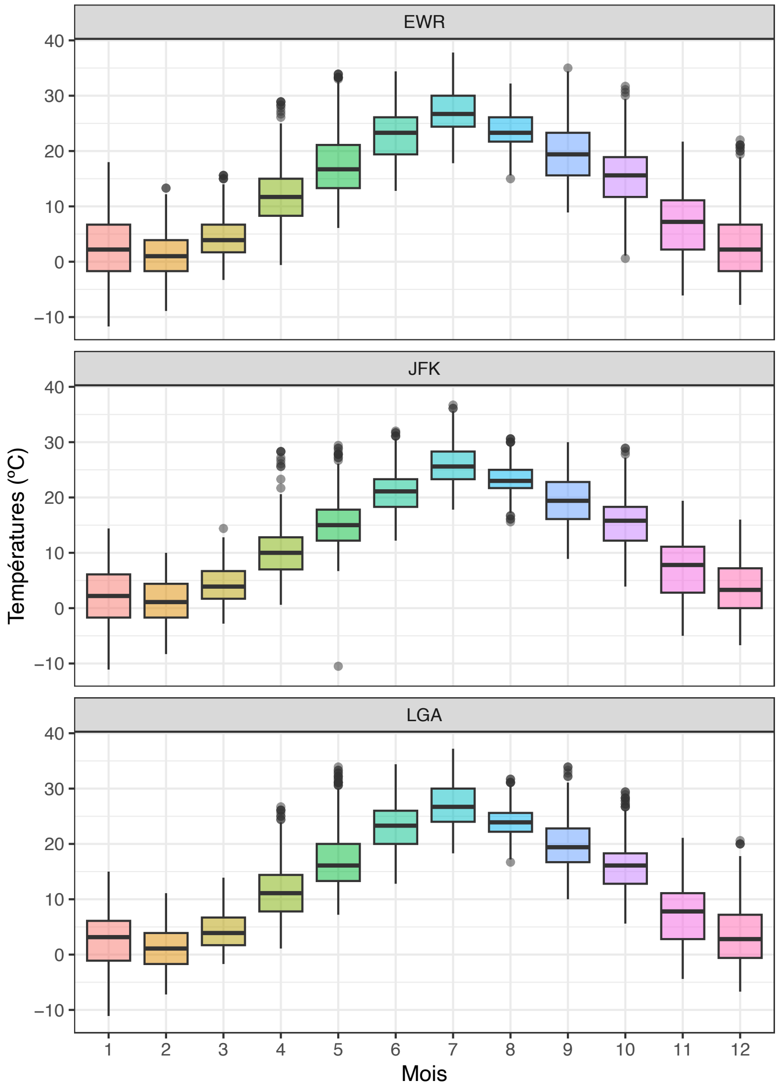


Figure 8.4: Distribution des températures mensuelles dans l
3 aéroports de New York en 2013

Vous voyez que le code est très proche pour produire un
chart ou un boxplot. Comme, il est dit au Chapitre 6, les dif-
férents éléments de chaque boîte nous renseignent sur
sition et sur la dispersion des données pour chaque m
chaque aéroport :

- La limite inférieure de la boîte correspond au pre
quartile : 25% des données de l'échantillon sont si
au-dessous de cette valeur.

- La limite supérieure de la boîte correspond au trois
quartile : 75% des données de l'échantillon sont si
au-dessous de cette valeur.
- Le segment épais à l'intérieur de la boîte correspo
second quartile : c'est la médiane de l'échantillo
nous renseigne sur la position de la distribution.
des données de l'échantillon sont situées au-dess
cette valeur, et 50% au-dessous.
- La hauteur de la boîte correspond à l'étendue (ou in
valle) interquartile ou Inter Quartile Range (IQR
anglais. On trouve dans cette boîte 50% des observ
tions de l'échantillon. C'est une mesure de la disp
des 50% des données les plus centrales. Une boîte p
allongée indique donc une plus grande dispersion.
- Les moustaches correspondent à des valeurs qui sont
dessous du premier quartile (pour la moustache du b
et au-dessus du troisième quartile (pour la mousta
du haut). La règle utilisée dans R est que les mous-
taches s'étendent jusqu'aux valeurs minimales et m
males de l'échantillon, mais elles ne peuvent en au
cas s'étendre au-delà de 1,5 fois la hauteur de la b
(1,5 fois l'IQR) vers le haut et le bas. Si des poi
apparaissent au-delà des moustaches (vers le haut
le bas), ces points sont appelés "outliers". On peu
observer ici pour plusieurs mois et pour les 3 aérop
(par exemple, en avril dans les 3 aéroports). Ce son
points qui s'éloignent du centre de la distribution
çon importante puisqu'ils sont au-delà de 1,5 fois
de part et d'autre du premier ou du troisième quarti
peut s'agir d'anomalies de mesures, d'anomalies de
sie des données, ou tout simplement, d'enregistrem
tout à fait valides mais atypiques ou extrêmes ; il
s'agit donc pas toujours de point aberrants. J'at
votre attention sur le fait que la défnition de ces
liers est relativement arbitraire. Nous pourrions
choix d'étendre les moustaches jusqu'à 1,8 fois l
(ou 2, ou 2,5). Nous observerions alors beaucoup mo
d'outliers. D'une façons générale, la longueur des
taches renseigne sur la variabilité des données en d
de la zone centrale. Plus elles sont longues, plus
riabilité est importante. Très souvent, l'examen a
des outliers est utile car il nous permet d'en appre
plus sur le comportement extrême de certaines obse

tions.

Lorsque les boîtes ont une forme à peu près symétriq de part et d'autre de la médiane (c'est le cas pour c exemple dans la plupart des catégories), cela sig qu'un histogramme des mêmes données serait symétriq également.

Les stripcharts et les boxplots sont donc un bon moye comparer rapidement la position et la dispersion d'un nombre de séries de données : ici, en quelques lignes d nous en comparons 12 pour chacun des 3 aéroports de Ne York.

Les histogrammes sont plus utiles lorsqu'il y a moins gories à comparer, comme le montre la Figure 8.11. Ils permettent en outre de mieux visualiser les distributions non sym ou qui présentent plusieurs pics (distribution bi- modales).

8.5 Visualiser l'incertitude : les barre d'erreur

Comme évoqué plus haut, il est important de ne pas confondre dispersion et incertitude. Lorsque l'on visualise des moyennes calculées à partir des données d'un échan il est important de faire apparaître des barres d'err correspondent en général :

- soit à l'erreur standard de la moyenne
- soit à l'intervalle de confance à 95% de la moyenne

Puisque deux choix sont possibles, il sera important ciser systématiquement dans la légende du graphique, ture des barres représentées. Revenons aux données de corporelles des manchots, et commençons par visualis masses moyennes avec les erreurs standards. Pour cela prends le tableau masses_se créé précédemment :

```
masses_se |>
  ggplot(aes(x = sex, y = moyenne, fill = species)) +
  geom_col(color = "grey20", alpha = 0.5) +
  geom_errorbar(aes(ymin = moyenne - erreur_standard,
```

```
               ymax = moyenne + erreur_standard),
           width = 0.15) +
  facet_wrap(~species, nrow = 1) +
  labs(x = "", y = "Masse moyenne (g)", fill = "Espèce") +
  theme_bw() +
  scale_fill_brewer(palette = "Accent") +
  scale_y_continuous(labels = number_format()) +
  scale_x_discrete(labels = c("Femelles", "Mâles"))
```

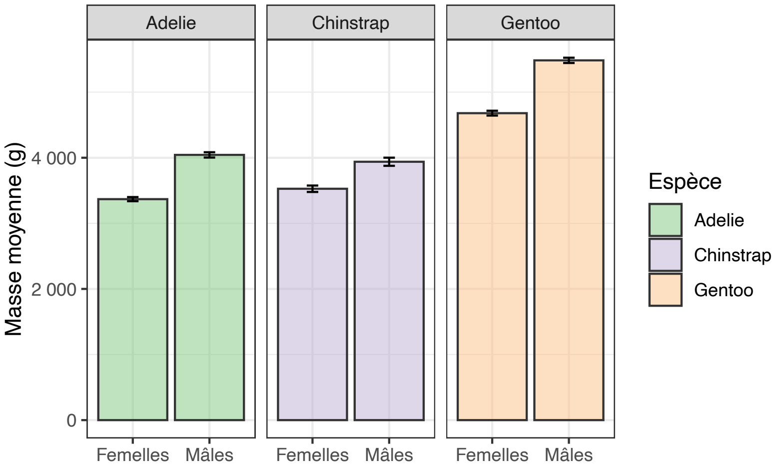


Figure 8.5 : Comparaison des masses moyennes observées chez les mâles et les femelles de 3 espèces de manchots. Les barres d'erreur sont les erreurs standard

Vous remarquerez que :

1. la fonction geom_errorbar() nécessite de nouvelles caractéristiques esthétiques qu'il nous faut ob toirement renseigner : les extrémités inférieur supérieures des barres d'erreur. Il nous faut do associer 2 variables à ces caractéristiques esthé Ici, nous utilisons moyenne - erreur_standard la borne inférieure des barres d'erreur, et moyenne + erreur_standard pour la borne supérieure. Les variables moyenne et erreur_standard faisant partie du tableau masses_se, geom_errorbar() les trouve sans difficulté.
2. l'argument width de la fonction geom_errorbar() met d'indiquer la longueur des segments horizonta qui apparaissent à chaque extrémité des barres d'er

Nous pouvons arriver au même résultats en utilisant le tableau masses_se_bornes, qui contient des variables différentes :

```
masses_se_bornes |>
  ggplot(aes(x = sex, y = moyenne, fill = species)) +
  geom_col(color = "grey20", alpha = 0.5) +
  geom_errorbar(aes(ymin = moyenne_moins_se,
                    ymax = moyenne_plus_se),
                width = 0.15) +
  facet_wrap(~species, nrow = 1) +
  labs(x = "", y = "Masse moyenne (g)", fill = "Espèce") +
  theme_bw() +
  scale_fill_brewer(palette = "Accent") +
  scale_y_continuous(labels = number_format()) +
  scale_x_discrete(labels = c("Femelles", "Mâles"))
```

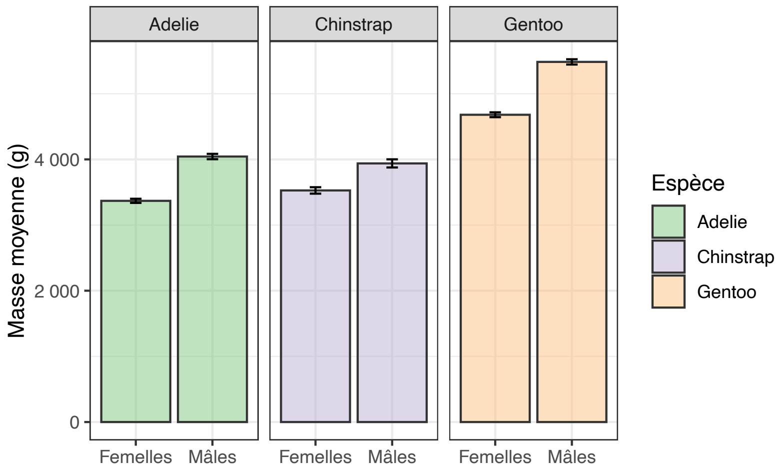


Figure 8.6: Comparaison des masses moyennes observées chez les mâles et les femelles de 3 espèces de manchots. Les barres d'erreur sont les erreurs standard

Seule la syntaxe de geom_errorbar() a changé puisque le tableau masses_se_bornes contient des variables différentes du tableau masses_se.

De la même façon, nous pouvons parfaitement faire apparaître, au lieu des erreurs standards, les intervalles de confiance à 95% de chaque masse moyenne. Il nous suffit pour cela d'utiliser le tableau masses_ic qui contient les

valeurs de moyennes et des bornes supérieures et infé
de ces intervalles :

```
masses_ci |>
  ggplot(aes(x = sex, y = moyenne, fill = species)) +
  geom_col(color = "grey20", alpha = 0.5) +
  geom_errorbar(aes(ymin = ci_borne_inf,
                    ymax = ci_borne_sup),
                width = 0.15) +
  facet_wrap(~species, nrow = 1) +
  labs(x = "", y = "Masse moyenne (g)", fill = "Espèce") +
  theme_bw() +
  scale_fill_brewer(palette = "Accent") +
  scale_y_continuous(labels = number_format()) +
  scale_x_discrete(labels = c("Femelles", "Mâles"))
```

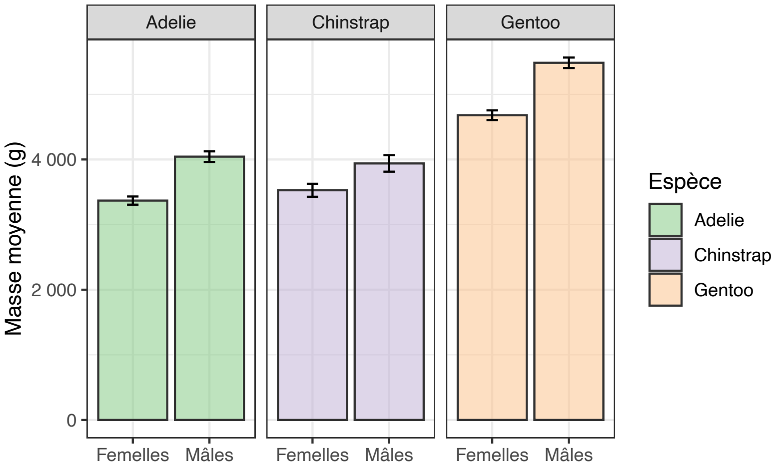


Figure 8.7: Comparaison des masses moyennes observées chez les mâles et les femelles de 3 espèces de manchots. Les barres d'erreur sont les intervalles de confia
à 95% des masses moyennes.

Comme vous voyez, les barres d'erreurs sont maintenan
longues que sur la Figure 8.5. C'est normal car rappelez-vous
que les intervalles de confiance sont à peu près équiv
à 2 fois les erreurs standards. L'intérêt de représe
intervalles de confiance est qu'ils sont directement
tests statistiques que nous aborderons dans les chapi
viennent. Globalement, quand 2 séries de données ont d
tervalles de confiance qui se chevauchent largement (

par exemple pour les mâles Adélie et Chinstrap), alor
test d'hypothèses conclurait presque toujours à l'ab
différence significative entre les 2 groupes. À l'inver
2 séries de données ont des intervalles de confiance q
se chevauchent pas du tout (comme les mâles et les feme
Adélie par exemple), alors, un test d'hypothèses con
presque toujours à l'existence d'une différence sign
entre les 2 groupes. Lorsque les intervalles de confia
2 catégories se chevauchent faiblement ou partiell
(comme entre les femelles Adélie et Chinstrap), la si
est moins tranchée, et nous devrons nous en remettre
résultats du test pour savoir si la différence observé
être considérée comme significative ou non.

8.6 Visualiser l'incertitude : les boîte moustaches

Outre les informations de position et de dispersion, l
à moustaches permettent également de visualiser l'inc
associée aux médianes. Il suffit pour cela d'ajouter l'a
notch = TRUE à la fonction geom_boxplot() :

```r
penguins |>
  filter(!is.na(sex)) |>
  ggplot(aes(x = sex, y = body_mass_g, fill = species)) +
  geom_boxplot(color = "grey20", alpha = 0.5, notch = TRUE) +
  facet_wrap(~species, nrow = 1) +
  labs(x = "Sexe", y = "Masse moyenne (g)", fill = "Espèce") +
  theme_bw() +
  scale_fill_brewer(palette = "Accent") +
  scale_y_continuous(labels = number_format()) +
  scale_x_discrete(labels = c("Femelles", "Mâles"))
```

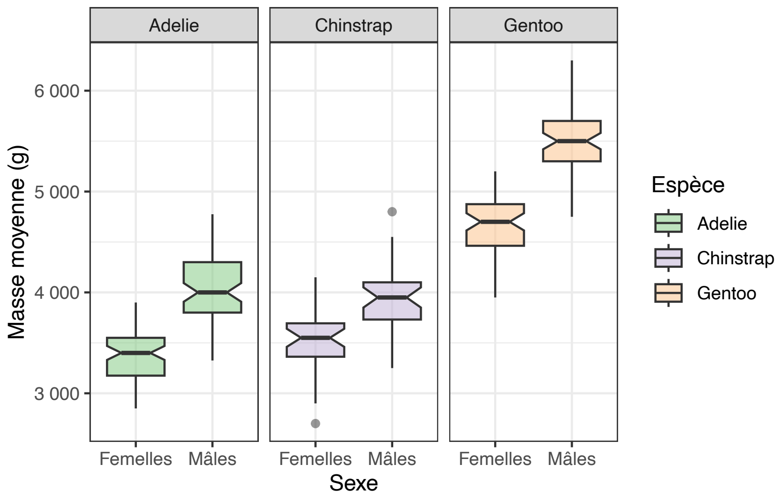


Figure 8.8: Comparaison des masses corporelles des mâles et femelles de 3 espèces de manchots.

Des encoches ont été ajoutées autour de la médiane d chaque boîte à moustache. Ces encoches sont des enco d'incertitudes. Les limites inférieures et supérieur coches correspondent aux bornes inférieures et supér l'intervalle de confance à 95% des médianes. Comme pou moyennes, le chevauchement ou l'absence de chevauche entre les encoches de 2 séries de données nous renseig l'issue probable des futurs tests statistiques que no amenés à réaliser. Notez que tout ce que nous avons dit haut sur le chevauchement des intervalles de confianc moyennes se retrouve ici pour les intervalles de con des médianes (e.g. large chevauchement entre les encoches des mâles Adélie et Chinstrap, absence de chevauchem entre femelles et mâles Adélie, faible chevauchemen femelles Adélie et Chinstrap). Il sera donc important examiner ces encoches en amont des tests statistique éviter de faire/dire des bêtises…

8.7 Exercice

1. Avec le tableau penguins, calculez les grandeurs suivantes pour chaque espèce de manchot et chaque sexe :

- la moyenne de la longueur des nageoires

- la variance de la longueur des nageoires
- l'écart-type de la longueur des nageoires
- l'erreur standard de la longueur moyenne des nageo
- la moyenne de l'épaisseur du bec
- la variance de l'épaisseur du bec
- l'écart-type de l'épaisseur du bec
- l'erreur standard de l'épaisseur du bec

Attention : pensez à retirer les individus dont le sex
connu.

2. Vérifez avec la fonction `skim()` que les moyennes et écart-types calculés ci-dessus sont corrects.

3. Avec ces données synthétiques faites le graphique
vant :