TP OUMOBIO 6: Représenter graphiquement les relations entre deux variables

Mathieu Brevet

2025-01-14

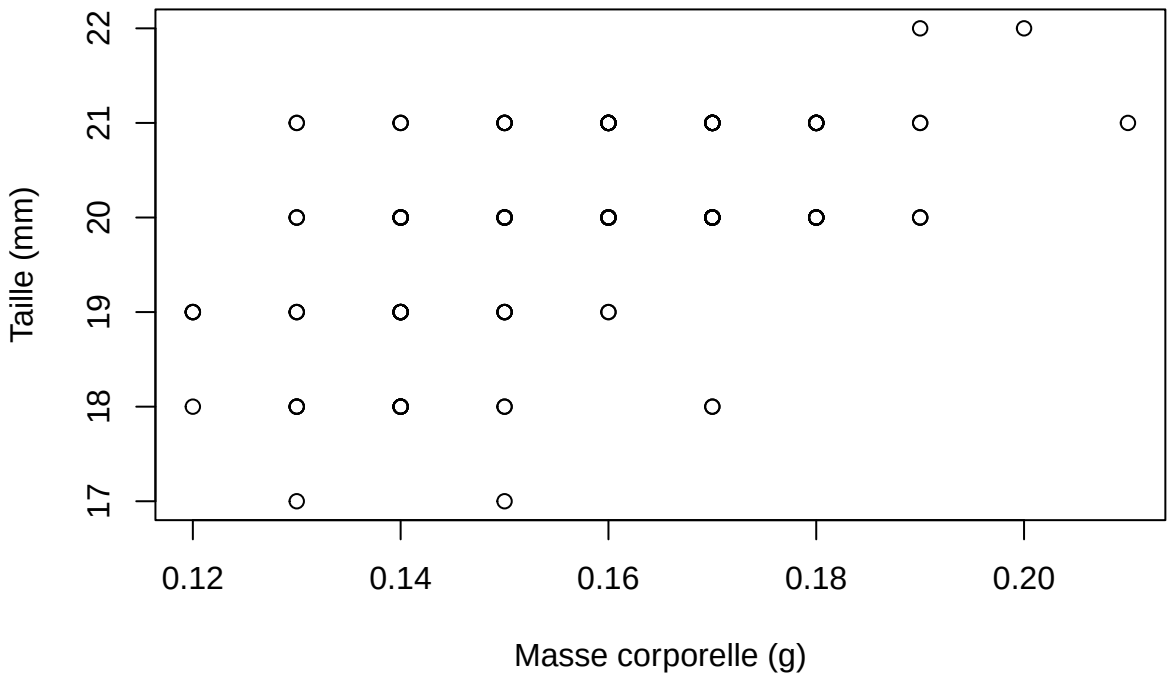
Bienvenue dans ce sixième TP sur R. Après avoir vu comment manipuler des données multivariées sous R, nous allons maintenant nous concentrer sur l'étude des relations statistiques entre deux variables (i.e. des analyses bivariées). Nous allons en particulier commencer par apprendre à explorer la nature de ces relations en les représentant graphiquement.

```
setwd("~/ATER PAU 2024/Cours modifiés/OUMOBIOS")


data_lezard = read.table("Suivi_lezard_vivipare.csv", header = T, sep = "\t", dec = ",")
```

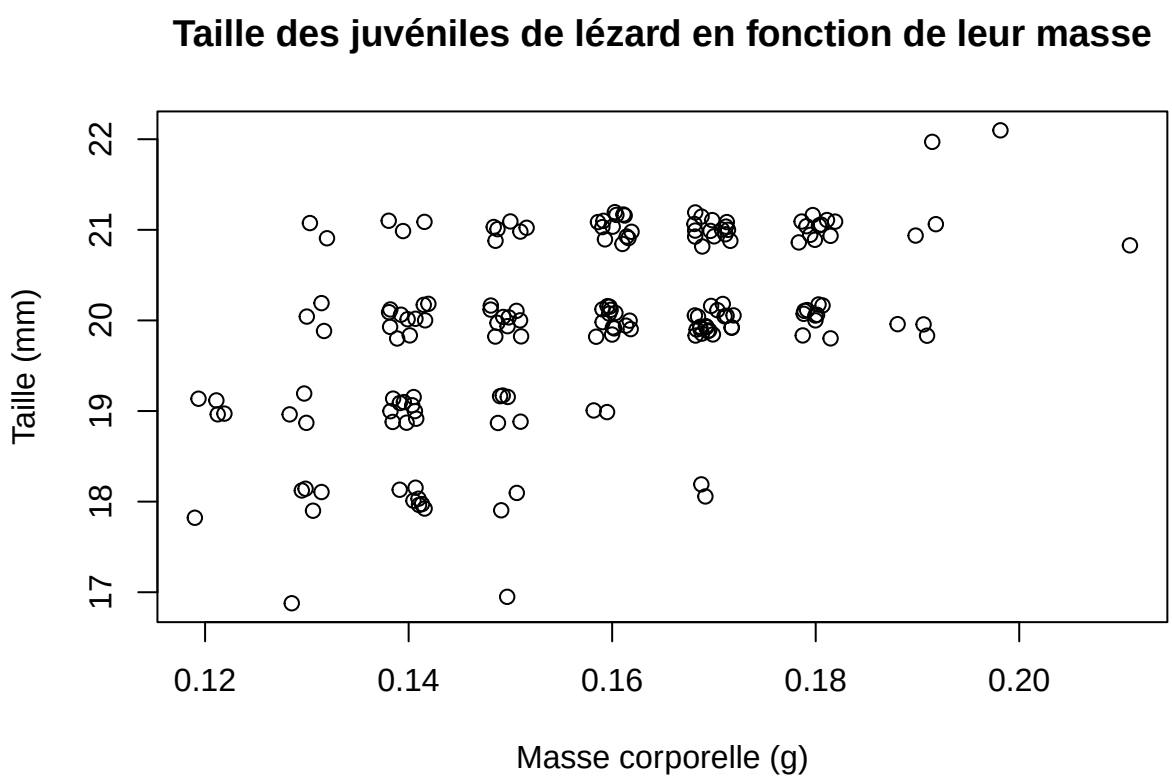
Décrire la relation entre deux variables quantitatives

Nous allons tenter de mettre en relation les différentes variables quantitatives du jeu de données entre elles, en produisant des **graphiques** mettant en relation des **variables quantitatives deux à deux**:

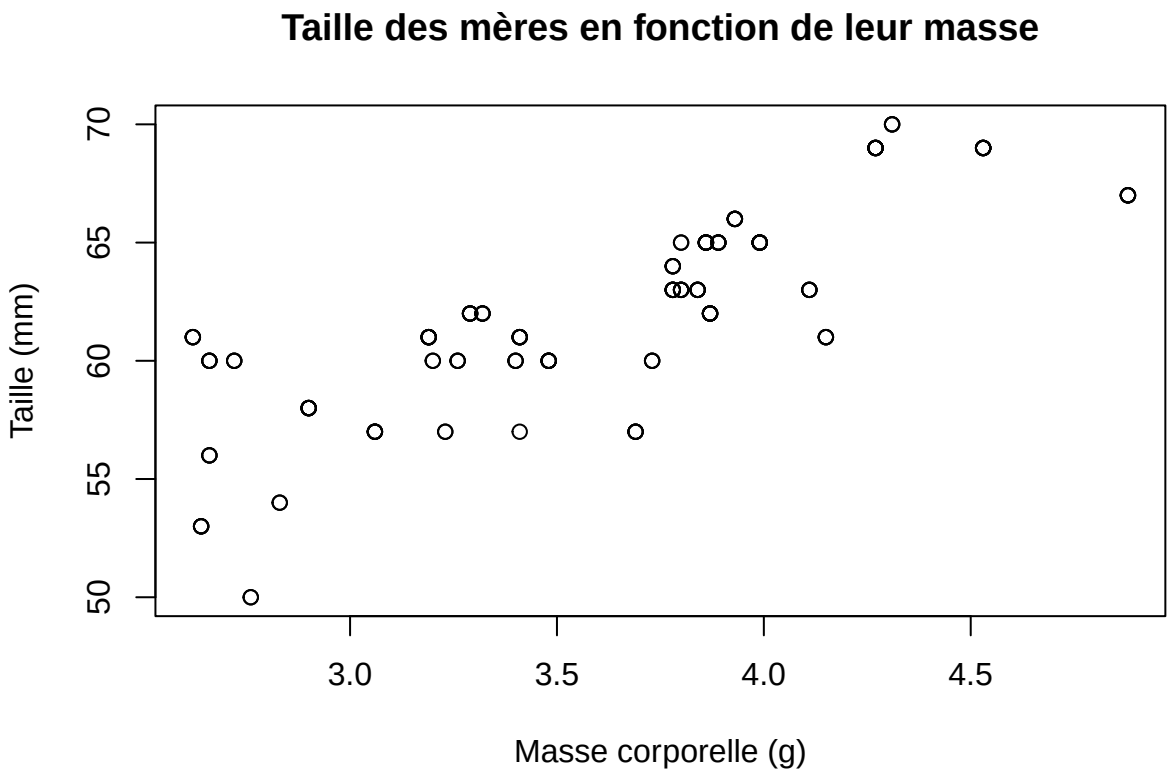
```
# essayons de mettre en relation la taille et le poids chez les juvéniles ou chez les mères:




plot(data_lezard$SVL_IND ~ data_lezard$M_IND,
     main = "Taille des juvéniles de lézard en fonction de leur masse",
     xlab = "Masse corporelle (g)",
     ylab = "Taille (mm)")
```



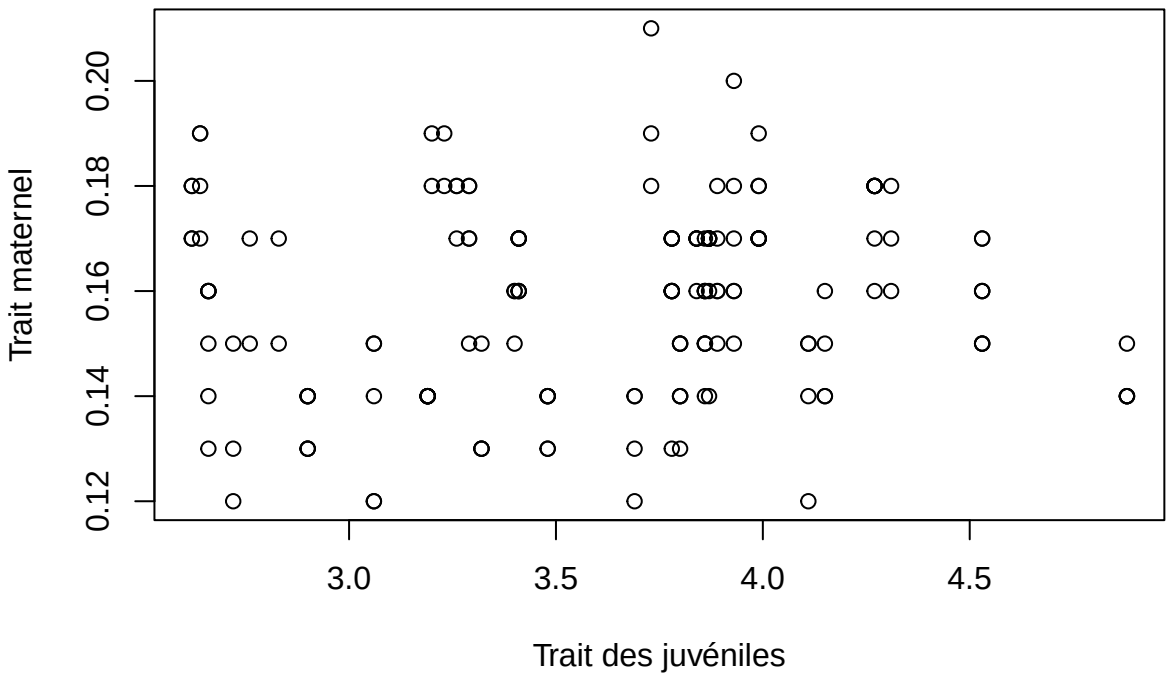

```
# graphique représentant la taille en fonction de la masse, beaucoup d'individus ont la
# même masse et taille, rendant le graphique peu lisible

# NB: dans R le signe "~" signifie "en fonction de", ainsi "plot(y~x)" signifie qu'on va
# tracer le graphique de y (en ordonnée) en fonction de x (en abscisse)


plot(jitter(data_lezard$SVL_IND, 1) ~ jitter(data_lezard$M_IND, 1),
     main = "Taille des juvéniles de lézard en fonction de leur masse",
     xlab = "Masse corporelle (g)",
     ylab = "Taille (mm)")
```

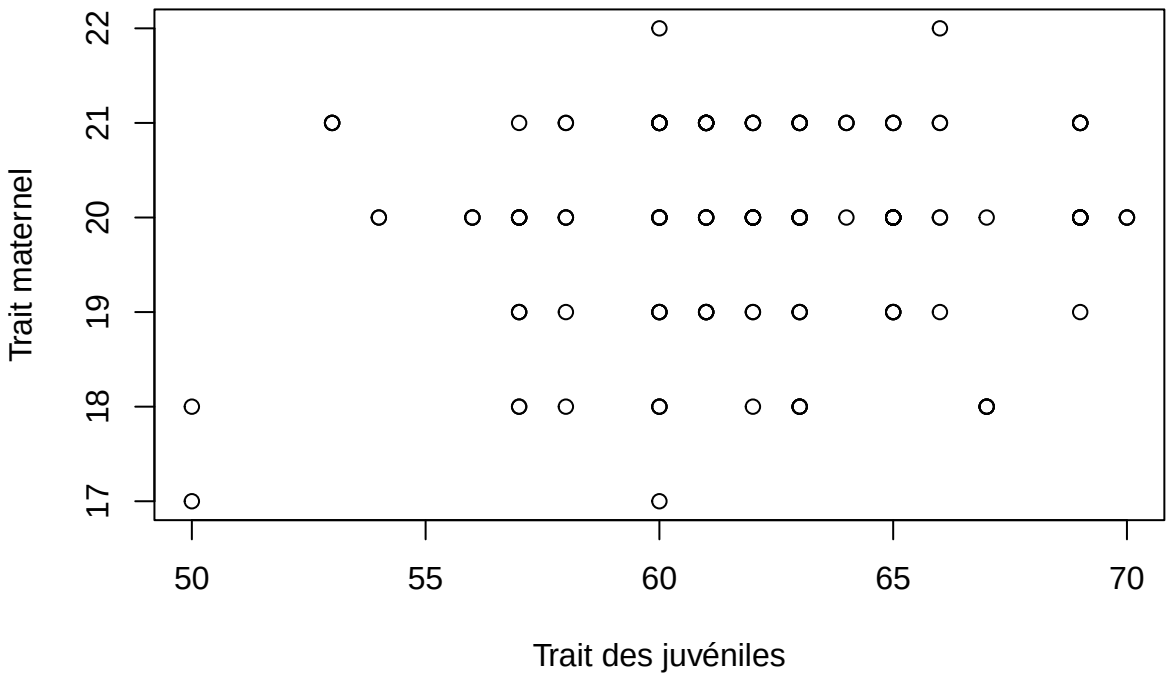



```
# représentation des points en ajoutant de l'instabilité aléatoire (fonction "jitter()")
# pour améliorer la lisibilité




plot(data_lezard$SVL_MOTHERS ~ data_lezard$M_MOTHERS,
     main = "Taille des mères en fonction de leur masse",
     xlab = "Masse corporelle (g)",
     ylab = "Taille (mm)")
```




```
# Essayons maintenant d'automatiser ces comparaisons deux à deux à l'aide d'une fonction
# de la famille "apply": "mapply()". Cette fonction permet d'appliquer une fonction à
# tous les éléments de plusieurs vecteurs/listes/tableaux, position par position.
# Elle s'adresse donc à des fonctions avec plusieurs arguments (de données) en entrée
# (autant qu'il y a de vecteurs/listes/tableaux utilisés)


# Nous allons comparer chacune des métriques (taille, poids) entre les
# juvéniles et leurs mères:


mapply(FUN = function(x,y) plot(x ~ y,
                          xlab = "Trait des juvéniles",
                          ylab = "Trait maternel"),
       data_lezard[, c("M_IND", "SVL_IND")],
       data_lezard[, c("M_MOTHERS", "SVL_MOTHERS")])
```


Trait maternel
0.12
0.14
0.16
0.18
0.20
3.0
3.5
4.0
4.5
Trait des juvéniles




```
## $M_IND
## NULL
##
## $SVL_IND
## NULL
```

```
# NB: comme montré ici les fonctions de la famille "apply" peuvent s'utiliser en
# définissant soit même sa fonction plutôt qu'en utilisant une fonction pré-définie.
# Dans ce cas il suffit d'écrire la fonction après avoir indiqué function("arguments")
# au préalable (ex: function(x) mean(x)), lorsque plusieurs commandes sont combinés
# dans la fonction ont peut les mettre entre accolades et les séparer par des
# sauts de ligne ou des points-virgules
```

```
# pour information, voici comment réaliser la même d'opération  avec des boucles "for" (combinées):


for (i in c("SVL_IND", "M_IND")) {
  for (j in c("SVL_MOTHERS", "M_MOTHERS")) {
    plot(data_lezard[, i] ~ data_lezard[, j],
         xlab = "Trait des juvéniles",
         ylab = "Trait maternel")
  }
}
```

Ces graphiques nous permettent d'estimer visuellement le niveau de **corrélation** entre deux variables quantitative, une notion que nous aborderons plus en détails (sur le plan statistique) au cours du prochain TP.

Ici on observe une corrélation positive entre la taille et le poids chez les juvéniles et chez leur mère (lorsque la taille augmente, le poids a lui aussi tendance à augmenter; on peut aussi parler de relation croissante). A l'inverse, il ne semble pas y avoir de corrélations entre mère et juvénile pour un même trait mesuré. On peut aussi estimer graphiquement la **linéarité** de la relation (pourrait être représentée par une droite), ainsi que sa **monotonie** (uniquement croissante ou décroissante).

EXERCICE

- Produisez le graphique permettant de comparer la masse des mères à leur corpulence (taille divisée par la masse). Que pouvez-vous dire sur la relation entre ces deux variables ?

```
plot(data_lezard$SVL_MOTHERS / data_lezard$M_MOTHERS ~ data_lezard$M_MOTHERS,
     main = "Corpulence des mères en fonction de leur masse",
     xlab = "Masse corporelle (g)",
     ylab = "Corpulence (mm/g)")
```

```
# une corrélation négative est observée: lorsque la masse corporelle augmente,
# la corpulence diminue (relation décroissante). La relation semble linéaire.
```

Décrire la relation entre une variable quantitative et une qualitative

Nous allons maintenant décrire une **variable quantitative en fonction d'une variable qualitative**. Par exemple nous allons décrire la masse et la taille des juvéniles en fonction de leur sexe.

```
# nous allons commencer par comparer la distribution de taille des juvéniles en
# fonction de leur sexe:

tapply(data_lezard$SVL_IND, data_lezard$SEX, summary)
```

```
## $f
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   20.00   20.00   20.23   21.00   22.00
##
## $m
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   17.00   19.00   20.00   19.74   20.00   22.00
```

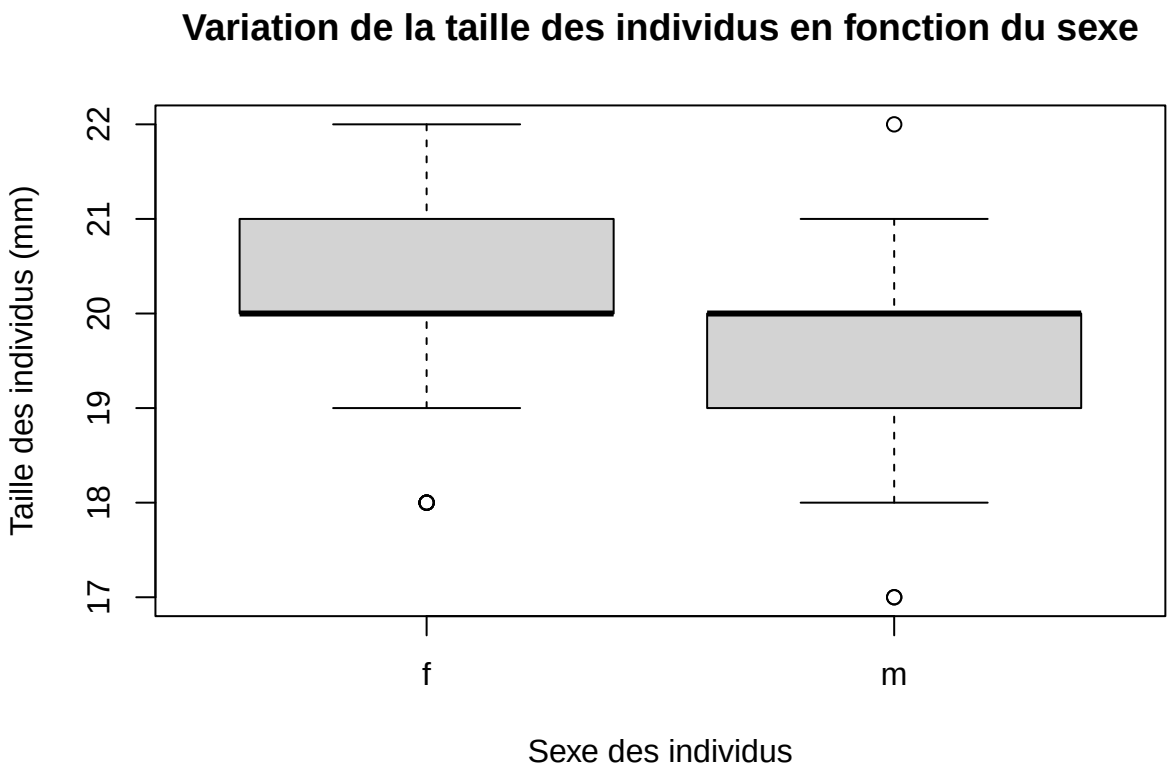
```
# résumés statistiques des distributions de taille pour les différents sexes
# (femelles et mâles)
```

POUR ALLER PLUS LOIN

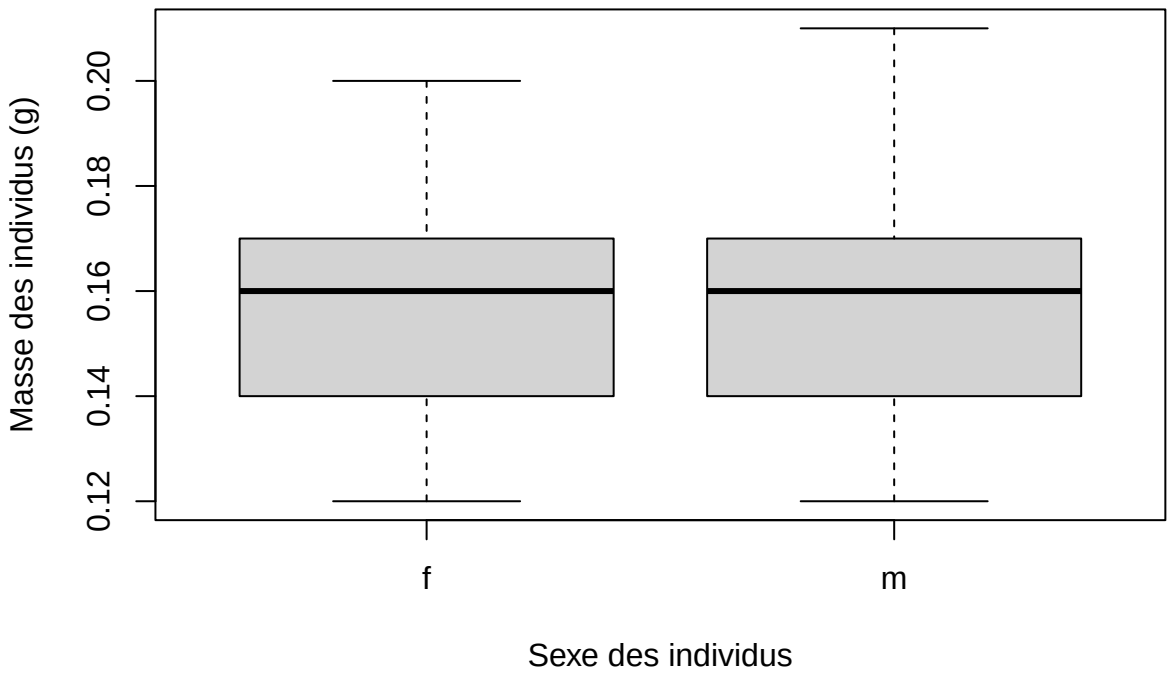
Il existe dans R d'autres fonctions de la famille apply, permettant notamment d'effectuer des opérations similaires à "tapply" mais simultanément sur plusieurs colonnes et/ou suivant plusieurs vecteurs de regroupement (fonction "aggregate"), ou encore directement sur plusieurs colonnes en entrée (fonction "by") ce qui permet alors d'utiliser des fonctions présentant plus d'un paramètre en entrée.

Après avoir décrit statistiquement notre variable quantitative en fonction des valeurs de notre variable qualitative, nous pouvons également produire des sorties graphiques permettant de telles comparaisons. Pour cela il convient d'utiliser des **boxplots** qui vont nous permettre de comparer les distributions entre les différentes classes de comparaisons:

```
boxplot(data_lezard$SVL_IND ~ data_lezard$SEX,
        xlab = "Sexe des individus",
        ylab = "Taille des individus (mm)",
        main = "Variation de la taille des individus en fonction du sexe")
```




```
boxplot(data_lezard$M_IND ~ data_lezard$SEX,
        xlab = "Sexe des individus",
        ylab = "Masse des individus (g)",
        main = "Variation de la masse des individus en fonction du sexe")
```




```
# il n'y a pas de différences majeures de taille ou de poids entre sexe
```

EXERCICE

- En utilisant des statistiques descriptives et des outils graphiques adaptés, déterminez s'il existe des différences de taille ou de masse corporelle en fonction de la date de naissance chez les juvéniles ou de la date de capture chez les mères.

```
tapply(data_lezard[, "M_IND"],
       data_lezard$BIRTH_DATE,
       summary)

tapply(data_lezard[, "SVL_IND"],
       data_lezard$BIRTH_DATE,
       summary)



boxplot(data_lezard$M_IND ~ data_lezard$BIRTH_DATE,
        ylab = "Masse (g)",
        xlab = "Date")
```

```r
boxplot(data_lezard$SVL_IND ~ data_lezard$BIRTH_DATE,
        ylab = "Taille (mm)",
        xlab = "Date")
```

```r
# Pas de relation visible entre taille/poids des juvéniles et leur date de naissance.
# On peut toutefois noter que les individus sont globalement de petite taille le 21/07
# et de masse importante le 22/07
```

```r
tapply(data_lezard[, "M_MOTHERS"],
       data_lezard$CAPT_DATE,
       summary)

tapply(data_lezard[, "SVL_MOTHERS"],
       data_lezard$CAPT_DATE,
       summary)
```

```r
boxplot(data_lezard$M_MOTHERS ~ data_lezard$CAPT_DATE,
        ylab = "Masse (g)",
        xlab = "Date")
```

```r
boxplot(data_lezard$SVL_MOTHERS ~ data_lezard$CAPT_DATE,
        ylab = "Taille (mm)",
        xlab = "Date")
```

```r
# Il n'y a pas de relation visible entre taille/poids des mères et date de capture.
# On peut toutefois observer que les tailles/poids sont plus élevés après le 18/06,
# à l'exception du 22/06
```

Décrire la relation entre deux variables qualitatives

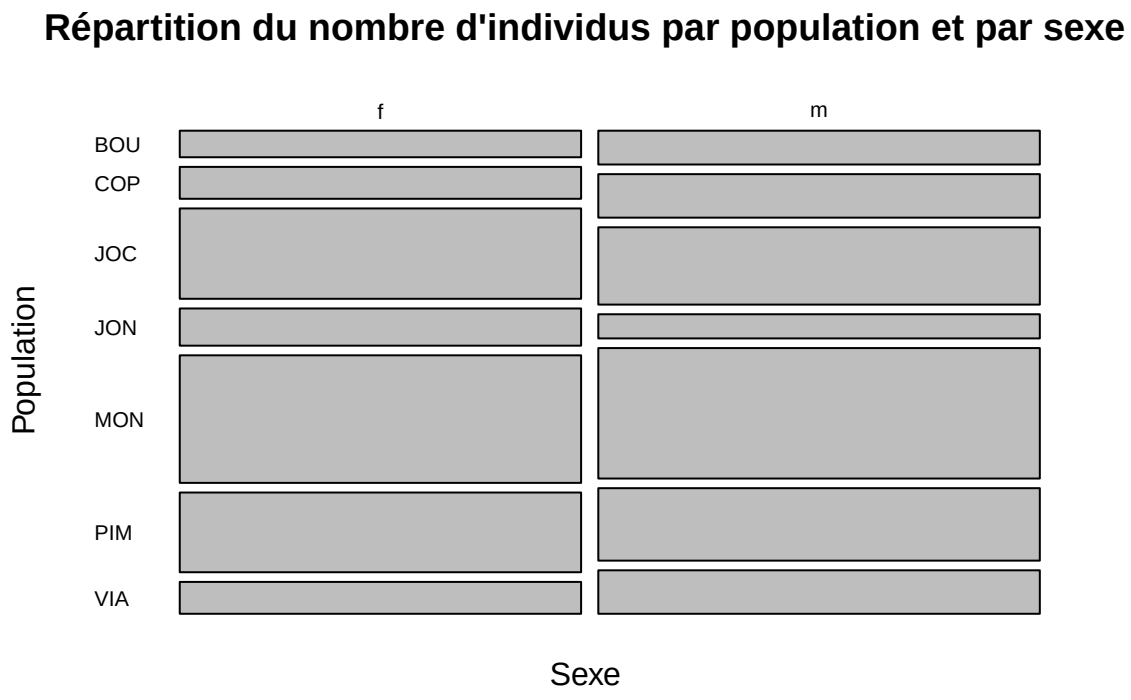
Pour finir, nous allons étudier comment réaliser une description graphique de la relation entre **deux variables qualitatives**. Cette représentation est basée sur la **table de contingence** issue du croisement des deux variables (c'est-à dire la table des effectifs de chaque combinaison de classes entre les deux variables). Le principal outil de visualisation est ensuite le **diagramme en barres** (comme pour une variable quantitative seule).

Voici un exemple décrivant la relation entre population d'origine et sexe des individus suivis:

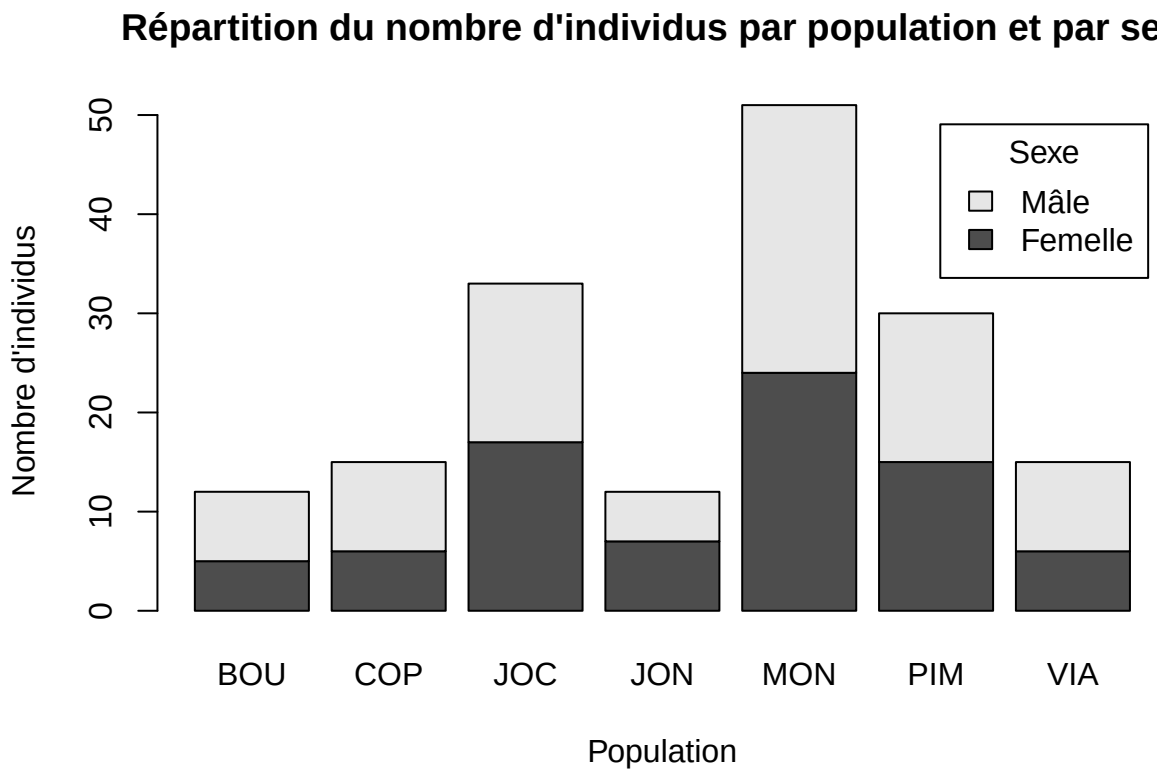
```
table(data_lezard$SEX, data_lezard$POP)
```

```
##
##     BOU COP JOC JON MON PIM VIA
##   f   5   6  17   7  24  15   6
##   m   7   9  16   5  27  15   9
```

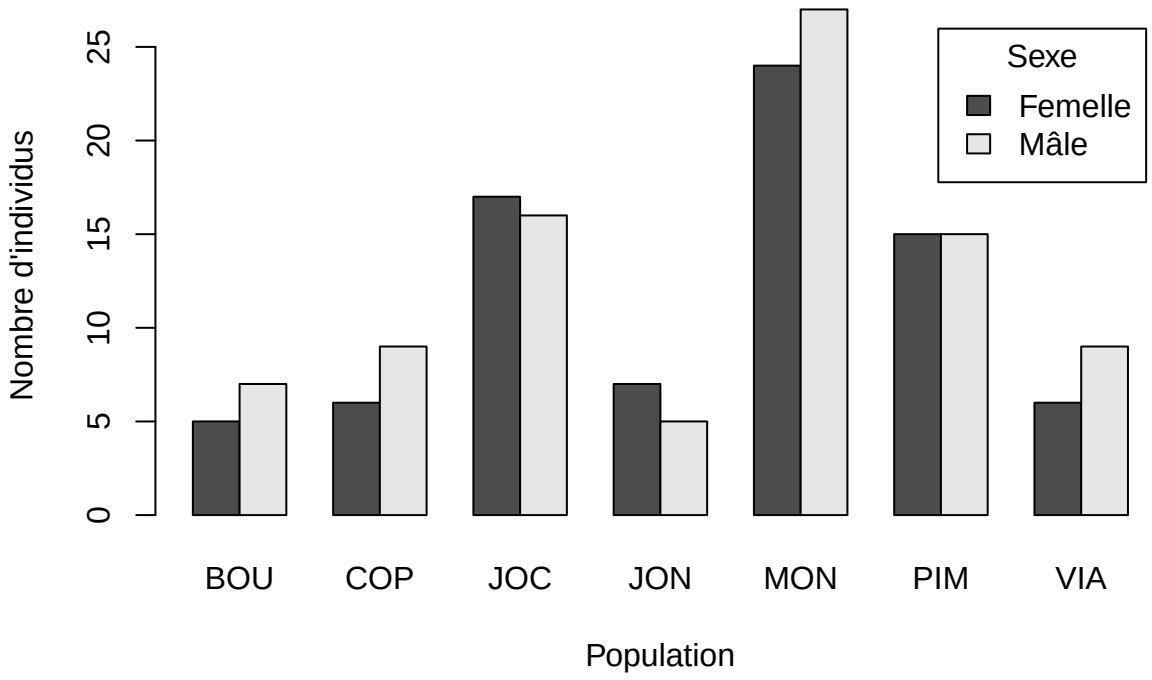
```
# table de contingence


plot(table(data_lezard$SEX, data_lezard$POP),
     main = "Répartition du nombre d'individus par population et par sexe",
     xlab = "Sexe",
     ylab = "Population",
     las = 1)
```



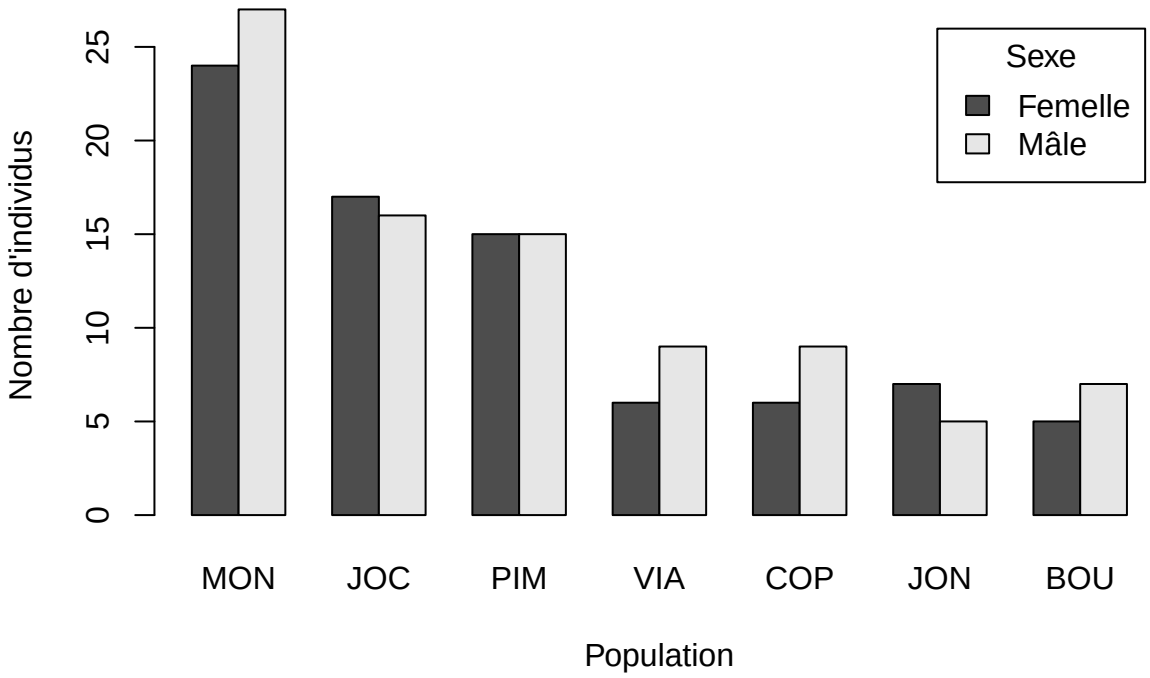
```r
# diagramme en boîte représentant le nombre d'individus par combinaison de classe
# (proportionnel à l'aire de chaque boîte)


barplot(table(data_lezard$SEX, data_lezard$POP),
        legend.text = c("Femelle", "Mâle"), # légende des couleurs de barres
        args.legend = list(title = "Sexe"), # titre des légendes de couleurs
        main = "Répartition du nombre d'individus par population et par sexe",
        xlab = "Population",
        ylab = "Nombre d'individus")
```



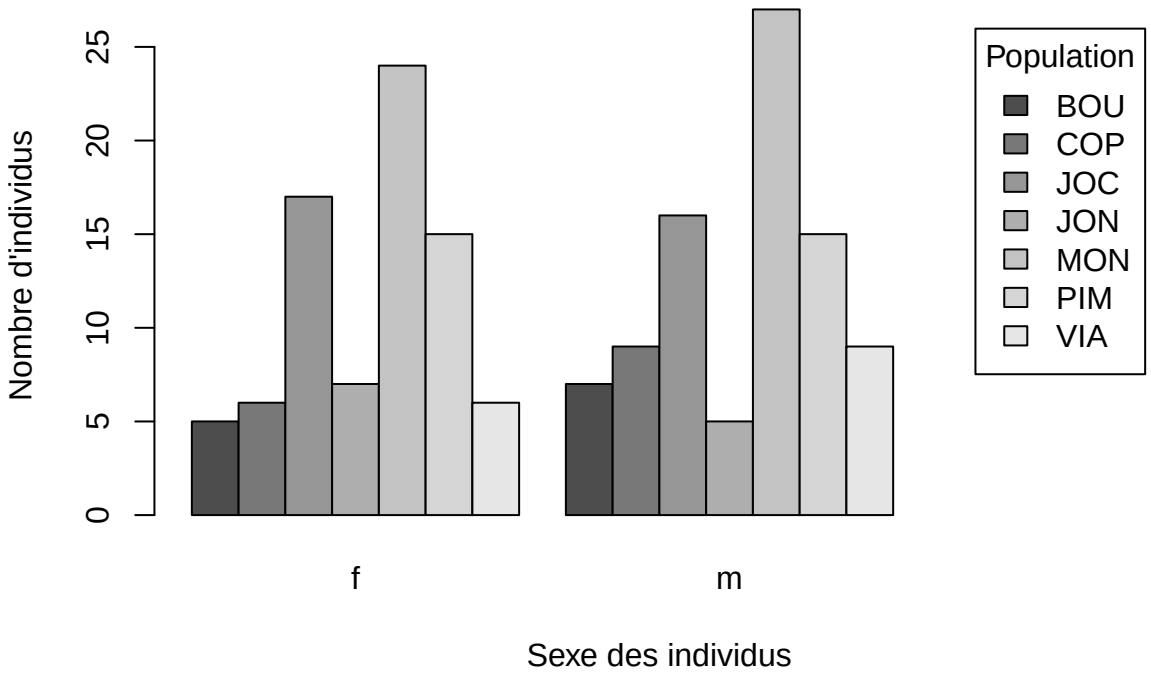

```r
# diagramme en barres représentant le nombre d'individus par combinaison de classe


barplot(table(data_lezard$SEX, data_lezard$POP),
        legend.text = c("Femelle", "Mâle"),
        args.legend = list(title = "Sexe"),
        main = "Répartition du nombre d'individus par population et par sexe",
        xlab = "Population",
        ylab = "Nombre d'individus",
        beside = T)
```




```r
# diagramme identique au précédent mais avec les catégories de sexe représentées côte à côte



# NB: on peut réordonner l'axe des abscisses (catégories) pour faire apparaître les
# classes dans l'ordre souhaité (exemple ci-dessous).

barplot(table(data_lezard$SEX,
              factor(data_lezard$POP,
                     levels = c("MON", "JOC", "PIM", "VIA", "COP", "JON", "BOU"))),
        legend.text = c("Femelle", "Mâle"),
        args.legend = list(title = "Sexe"),
        main = "Répartition du nombre d'individus par population et par sexe",
        xlab = "Population",
        ylab = "Nombre d'individus",
        beside = T)
```




```
# transformation de la variable "POP" en facteur et ordination de ces niveaux (=catégories)
# dans l'ordre souhaité, à l'aide de l'argument "levels"




# Rq: l'ordre des variables dans la table de contingence importe !


barplot(table(data_lezard$POP, data_lezard$SEX),
        legend.text = c("BOU", "COP", "JOC", "JON", "MON", "PIM", "VIA"),
        args.legend = list(title = "Population"),
        main = "Répartition du nombre d'individus par population et par sexe",
        xlab = "Sexe des individus",
        ylab = "Nombre d'individus",
        xlim = c(1, 21), # élargissement abscisse pour affichage légende
        beside = T)
```



EXERCICE

- Décrivez la relation entre population d'origine et date de naissance. Que pouvez-vous en dire ?

```
table(data_lezard$BIRTH_DATE, data_lezard$POP)

barplot(
    table(data_lezard$POP, data_lezard$BIRTH_DATE),
    legend.text = c("BOU", "COP", "JOC", "JON", "MON", "PIM", "VIA"),
    args.legend = list(title = "Population"),
    main = "Répartition du nombre d'individus par population et par date de naissance",
    xlab = "Date de naissance", ylab = "Nombre d'individus", beside = T
)
```

```
# il y a une ségrégation des populations par date de naissance: chaque
# population est associée à une période de naissance spécifique
```

Bilan

Nous avons appris au cours de cette séance à **comparer** graphiquement **des variables deux à deux**, à l'aide de graphiques **"nuage de points" ("plot()")** pour deux variables **quantitatives**, de graphiques **boîtes à moustaches ("boxplot()"**) pour représenter une variable **quantitative en fonction d'une variable qualitative**, et des **diagrammes en barres ("barplot()")** pour comparer deux variables **qualitatives** (en utilisant leur **table de contingence: "table()"**).
A l'intérieur des commandes graphiques, les relations entre variables (toutes deux quantitatives ou quantitative et qualitative) peuvent être définies à l'aide de l'opérateur **"~"** qui signifie **"en fonction de"**. Ainsi "plot(y ~ x)" peut être lu "graphique représentant y (= axe des ordonnées) en fonction de x (= axe des abscisses)".
Ces opérations permettent d'explorer le type de relation existant entre deux variables, et notamment pour deux variables quantitatives de faire une première estimation de la **linéarité** (le nuage de points pourrait être représenté par une ligne droite), la **monotonie** (la relation semble uniquement croissante ou décroissante), et la **croissance/décroissance**. Pour la comparaison des valeurs d'une variable quantitative par catégorie d'une variable qualitative, on se focalisera sur le niveau de **chevauchement des distributions** (représentés par les boîtes à moustaches), les distributions entièrement disjointe représentant une différence de valeurs marquée entre catégories.