TP OUMOBIO 3: Fonctions statistiques de base et visualisation des données sous R

Mathieu Brevet

2024-09-11

Bienvenue dans ce second volet de TP sur R ! Nous allons maintenant étudier comment décrire des variables qualitatives et quantitatives sur les plans statistique et graphique.

Décrire une variable qualitative

Dans un premier temps nous allons essayer de comprendre comment **décrire une variable qualitative** sur les plans statistique et graphique.

Reprenons notre jeu de données d'études là où nous l'avions laissé:

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


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

Descriptions statistiques

Nous allons dans un premier temps décrire la variable "Population d'origine" (colonne POP) de notre jeu de données. Nous allons tout d'abord en faire une **description statistique** en essayant de comprendre la répartition des individus dans les différentes populations:

```
unique(data_lezard$POP)
```

```
## [1] "MON" "JOC" "JON" "COP" "VIA" "PIM" "BOU"
```

```
# classes (=valeurs) de cette variable catégorielle

table(data_lezard$POP)
```

```
##
## BOU COP JOC JON MON PIM VIA
##  12  15  33  12  51  30  15
```

```
# effectifs de chaque classe de la variable, aussi appelé table de contingence

table(data_lezard$POP)/length(data_lezard$POP)
```

```
##
##         BOU         COP         JOC         JON         MON         PIM         VIA
## 0.07142857 0.08928571 0.19642857 0.07142857 0.30357143 0.17857143 0.08928571
```

```
# fréquence de chaque classe de la variable


# essayons également de trouver le mode de la variable (valeur pour laquelle on
# a l'effectif maximal):

max(table(data_lezard$POP))
```

```
## [1] 51
```

```
# effectif maximal

names(which.max(table(data_lezard$POP)))
```

```
## [1] "MON"
```

```
# la fonction which.max() permet de déterminer la position de la valeur
# maximale, puis names() permet de récupérer le nom associé à cette position
```

Prenons un autre exemple avec les dates de naissances des individus, nous voulons comprendre comment le nombre de naissance évolue dans le temps, nous allons donc utilisé des fréquences cumulées:

```
cumsum(table(data_lezard$BIRTH_DATE)/length(data_lezard$BIRTH_DATE))
```

```
##  02/07/19  04/07/19  05/07/19  10/07/19  11/07/19  12/07/19  13/07/19  16/07/19
## 0.1607143 0.1845238 0.2678571 0.2916667 0.3511905 0.3928571 0.5714286 0.6607143
##  17/07/19  18/07/19  20/07/19  21/07/19  22/07/19  23/07/19  24/07/19
## 0.7500000 0.8035714 0.8750000 0.8869048 0.9107143 0.9345238 1.0000000
```

```
# cumsum() permet de réaliser la somme cumulée d'un vecteur
```

Lorsque nous avons cherché à décrire les fréquences de classes et le mode de la variable nous avons utilisé une suite de commande. Au lieu d'utiliser cette suite de commande de manière répétée à chaque fois que nous en aurons besoin, nous pouvons construire **une fonction** nous permettant de formaliser l'opération et de pouvoir l'appeler dès que nécessaire (et ainsi gagner du temps dans l'écriture des scripts).
Une fonction s'écrit sur R sous la forme: **Nom <- function(arg)** {...} ("Nom": nom de la fonction; "arg": argument de la fonction, il peut y en avoir plusieurs séparés par des virgules; "...": commandes réalisées par la fonction, chaque nouvelle commande est séparé par un retour à la ligne ou un point-virgule). Voici deux exemples de construction de fonction:

```
freq <- function (vect) { # argument de la fonction: vecteur de chaînes de caractères
  table(vect) /       # table des effectifs du vecteur
    length(vect)      # taille du vecteur
}
# fonction calculant la fréquence de toutes les classes d'une variable catégorielle
```

```r
mode <- function (vect) { # argument de la fonction: vecteur
  names(                          # récupère le nom sur une variable avec identifiant
    which.max(                    # récupère la (ou les) positions de la valeur maximale
      table(                      # table des effectifs du vecteur
        vect)
      )
    )
}
# fonction calculant le mode d'une variable
```

BONNES PRATIQUES

Il est utile de regrouper les fonctions au même endroit dans votre script dans le cas de fonctions qui sont utilisées de manière récurrentes au cours de votre script, pour qu'un lecteur (ou vous-mêmes) puisse rapidement y accéder pour les consulter ou les modifier. Une bonne pratique consiste à réaliser un **préambule** au tout début de votre script pour regrouper vos fonctions (et décrire à l'aide de commentaires leur fonctionnement et utilité).

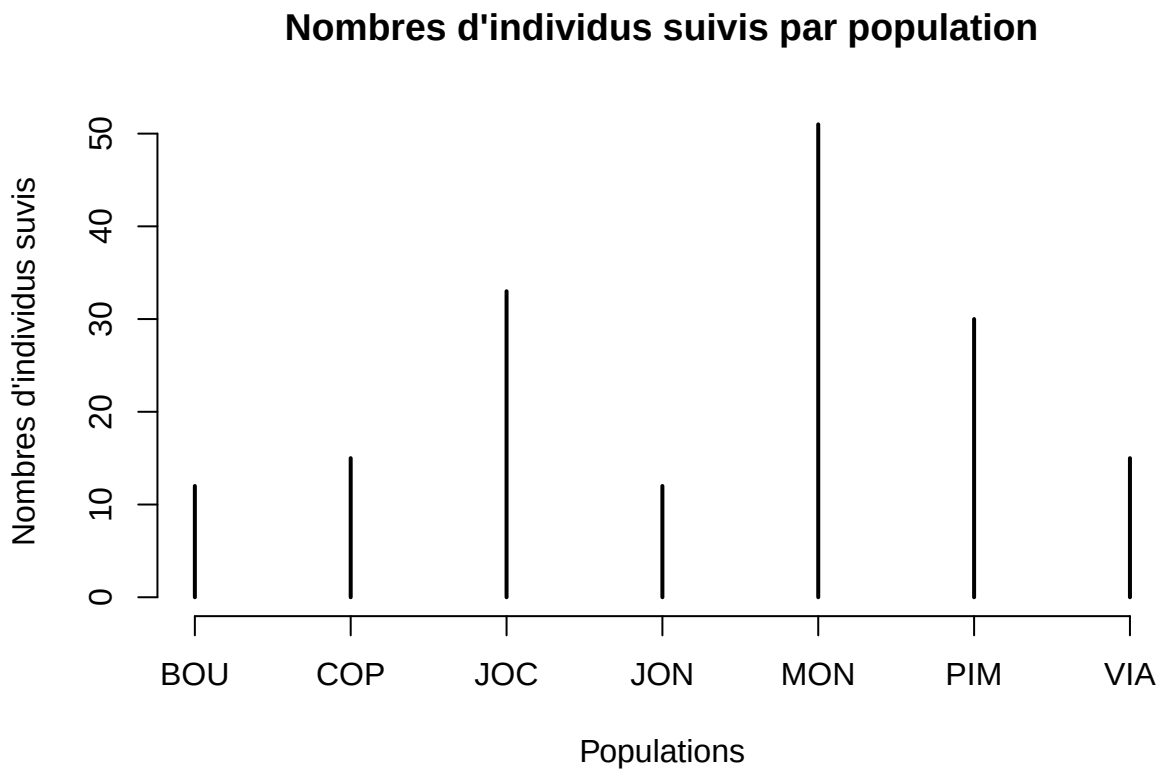
POUR ALLER PLUS LOIN

Comme vous avez déjà pu le constater il existe souvent plusieurs manières de réaliser une opération complexe sur R et donc d'écrire une fonction. Il peut parfois être intéressant d'identifier la solution la plus efficiente parmi les différentes possibilités. Pour cela il existe des outils dans R qui permettent de calculer le temps écoulé au cours d'une opération (system.time() ou microbenchmark() du paquet "microbenchmark" qui permet de comparer le temps de plusieurs opérations en les répétant de nombreuses fois), plus ce temps est court plus votre opération/fonction est performante. Cela peut être très utile si vous répétez de très nombreuses fois une opération/fonction ou si vous l'utilisez sur des volumes de données extrêmement important. Cela peut également vous permettre d'identifier quelles parties de votre fonction/opération ralentissent votre processus.

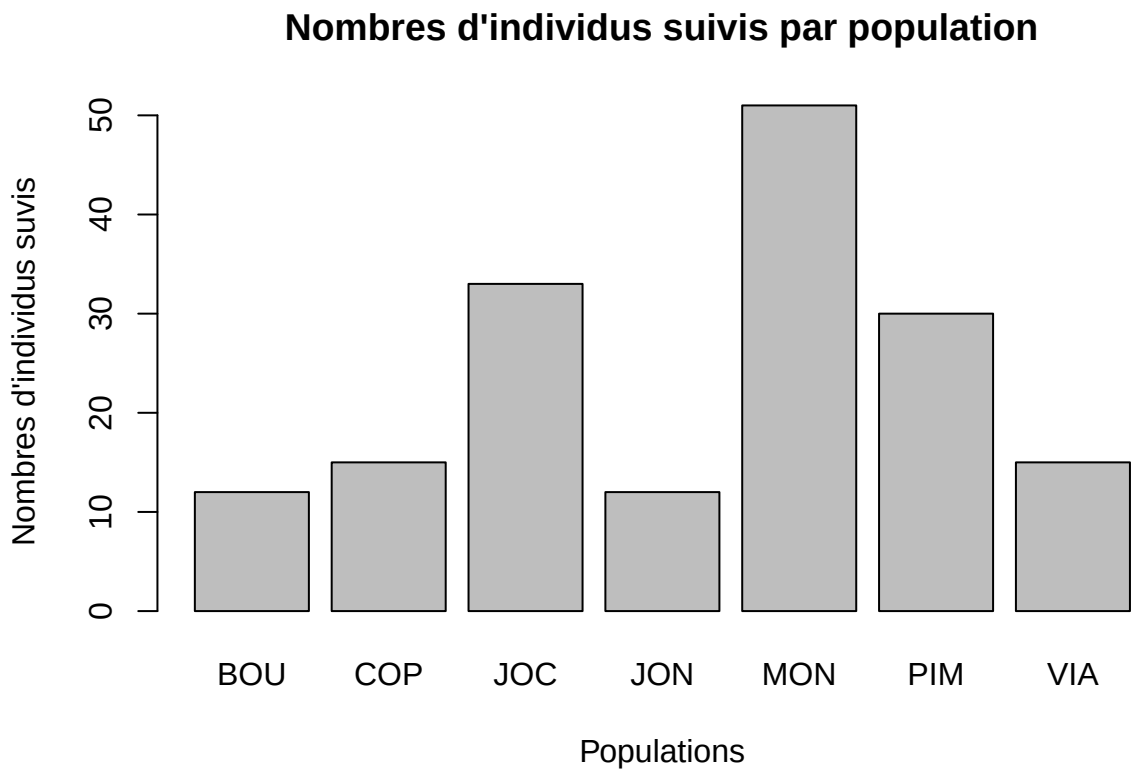
Descriptions graphiques

Après avoir décrit la variable sur le plan statistique nous allons maintenant illustrer nos résultats graphiquement. N'oubliez pas certaines bonnes pratiques lorsque vous créez un graphique (que vous utiliserez pour un rendu): le graphique doit toujours avoir un **titre**, des **légendes** décrivant les axes et une **échelle** facilement lisible. Voici les principaux **outils graphiques** que vous pouvez utiliser pour une variable qualitative:

```r
plot(
  table(data_lezard$POP),
  main = "Nombres d'individus suivis par population",
  xlab = "Populations",
  ylab = "Nombres d'individus suvis"
  )
```

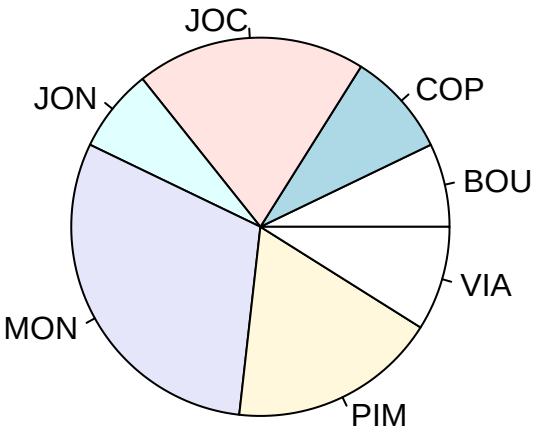



```r
# autre visualisation, la plus employée et commune pour une variable qualitative (conseillée):
barplot(
  table(data_lezard$POP),
  main = "Nombres d'individus suivis par population",
  xlab = "Populations",
  ylab = "Nombres d'individus suvis"
  )
```

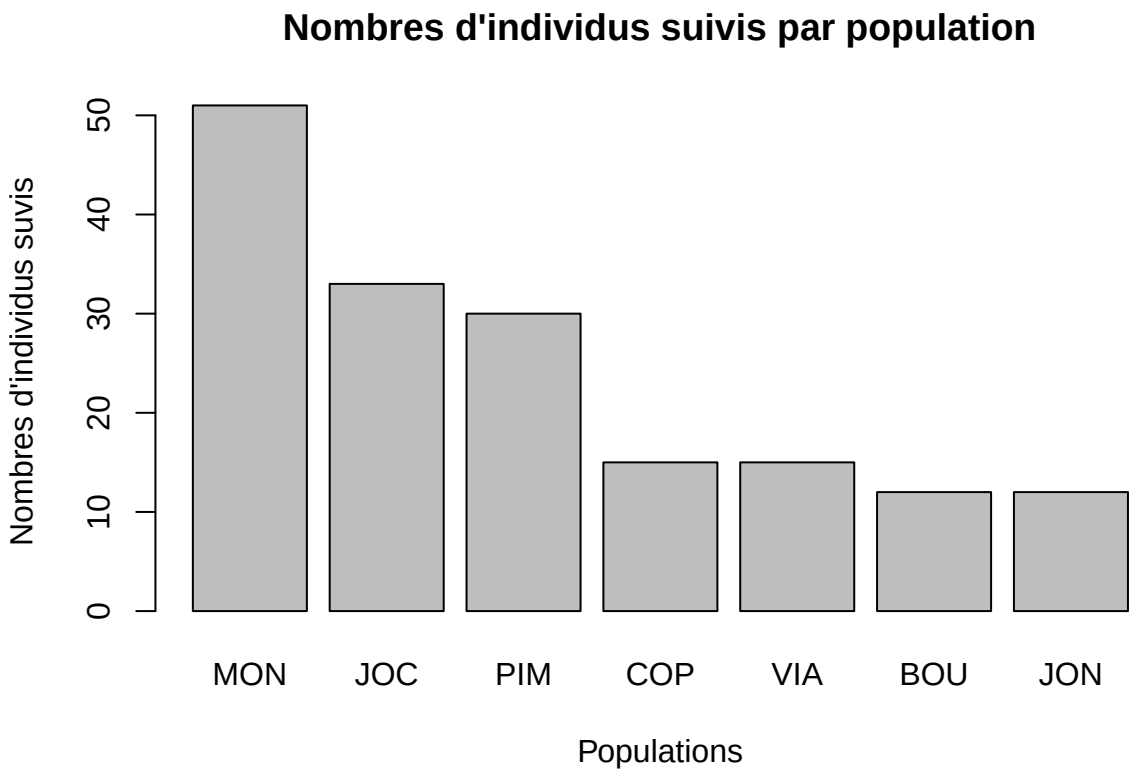



```
pie(
  table(data_lezard$POP),
  main = "Proportion d'individus suivis par population"
  )
```

Proportion d'individus suivis par population




```r
# diagramme criculaire (peu utilisé)



# visualiser après un tri des valeurs:

table = table(data_lezard$POP) [order(table(data_lezard$POP), decreasing = T)]

barplot(
  table,
  main = "Nombres d'individus suivis par population",
  xlab = "Populations",
  ylab = "Nombres d'individus suvis"
  )
```




```
 # enregistrement d'un graphique dans votre dossier de travail:

png("Distribution_population.png", res=100)
# ouvre une session graphique, enregistrement en png (aussi possible en jpeg, pdf), avec
# la résolution en ppi (pixels per inch, pour les formats hors pdf), et les dimensions
# gérés par les paramètres (height, width)
barplot(
  table,
  main = "Nombres d'individus suivis par population",
  xlab = "Populations",
  ylab = "Nombres d'individus suvis"
  )
# création figure
dev.off()
```

```
## pdf
##   2
```

```
graphics.off()
# fermeture de la session graphique


rm(table)
```

NOTE IMPORTANTE

Les figures que vous créez sous R peuvent être enregistrées sur votre dossier de travail, soit directement en utilisant R (voir lignes de code ci-dessus), soit en utilisant l'interface Rstudio, dans le panel en bas à droite, sur l'onglet "Plots" (voir image ci-dessous). L'avantage de la première méthode est qu'elle permet de gérer la résolution de l'image, chose très utile dans le cas où le rendu doit être propre. L'export en pdf assure une qualité optimale dans tous les cas mais peut être plus dur à insérer dans certains documents.

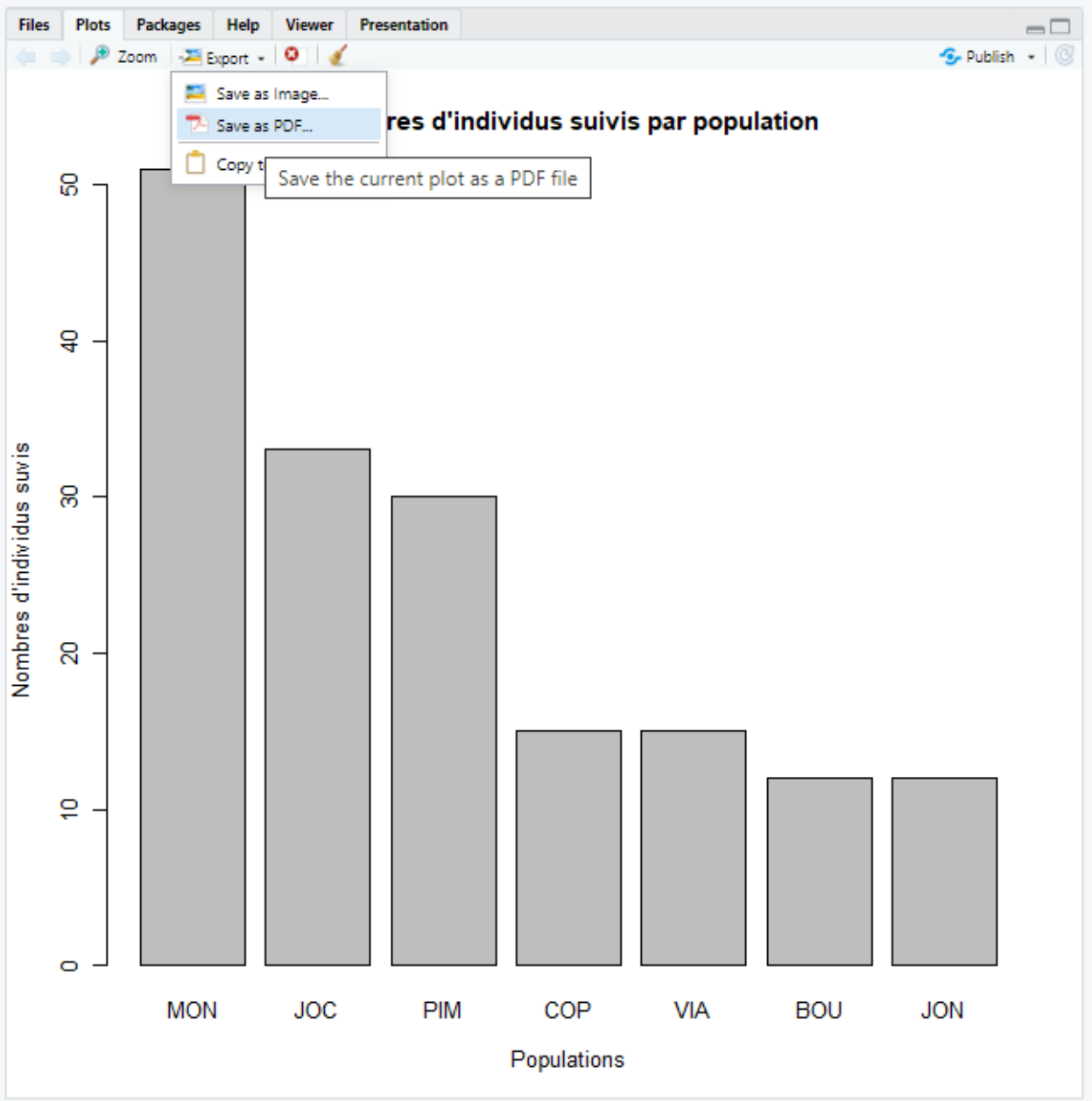


Figure 1: Enregistrement d'une image

EXERCICE

- Créez une fonction permettant d'obtenir des fréquences cumulées
- Convertissez la taille des individus en variable catégorielle et décrivez-la statistiquement et graphiquement

```
cumfreq <- function (vect) { # argument de la fonction: vecteur de chaînes de caractères
  cumsum(               # somme cumulée des éléments d'un vecteur
    table(vect) /       # table des effectifs du vecteur
      length(vect)      # taille du vecteur
    )
}
# fonction calculant la fréquence cumulée de toutes les classes d'une variable catégorielle




cat_taille = as.character(data_lezard$SVL_IND)

table(cat_taille)
mode(cat_taille)
cumfreq(cat_taille)

barplot(table(cat_taille),
        main = "Nombres d'individus par valeurs de taille",
        xlab = "Tailles mesurées",
        ylab = "Nombres d'individus"
  )
```

```
rm(cat_taille)
```

Décrire une variable quantitative

Descriptions statistiques

Nous allons maintenant décrire une **variable quantitative**, en prenant ici la masse des juvéniles suivis comme exemple. Dans un premier temps nous allons décrire les **caractéristiques de position et de dispersion** de ce vecteur de valeurs:

```
# paramètres de position de la distribution:

mean(data_lezard$M_IND)
```

```
## [1] 0.1576786
```

```
# masse moyenne des juvéniles

median(data_lezard$M_IND)
```

```
## [1] 0.16
```

```r
# masse médiane des juvéniles

quantile(data_lezard$M_IND)
```

```
##   0%  25%  50%  75% 100%
## 0.12 0.14 0.16 0.17 0.21
```

```r
# quantile (par défaut quartile) de la distribution

quantile(data_lezard$M_IND, seq(0, 1, 0.1))
```

```
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
## 0.12 0.13 0.14 0.15 0.15 0.16 0.16 0.17 0.17 0.18 0.21
```

```r
# décile de la distribution

mode(data_lezard$M_IND)
```

```
## [1] "0.17"
```

```r
# mode (valeur la plus représentée) de la distribution de masse



# exemples de distribution ne pouvant pas être différenciées sur la base de
# leurs paramètres de position:

mean(c(10, 10, 11, 12, 12)) ==
    mean(c(0, 0, 0, 0, 55))
```

```
## [1] TRUE
```

```r
mean(c(0, 0, 50, 100, 100)) ==
    mean(c(40, 40, 50, 51, 51))
```

```
## [1] FALSE
```

```r
mean(c(40, 50, 50, 60, 70)) ==
    mean(c(0, 0, 50, 100, 120))
```

```
## [1] TRUE
```

```r
median(c(40, 50, 50, 60, 70)) ==
    median(c(0, 0, 50, 100, 120))
```

```
## [1] TRUE
```

```r
# paramètres de dispersion de la distribution:


min(data_lezard$M_IND)
```

```
## [1] 0.12
```

```r
# valeur minimale de la distribution

max(data_lezard$M_IND)
```

```
## [1] 0.21
```

```r
# valeur maximale de la distribution

range(data_lezard$M_IND)
```

```
## [1] 0.12 0.21
```

```r
# min et max de la distribution

max(data_lezard$M_IND) -
    min(data_lezard$M_IND)
```

```
## [1] 0.09
```

```r
# étendue de la distribution

IQR(data_lezard$M_IND)
```

```
## [1] 0.03
```

```r
# écart inter-quartiles de la distribution




var(data_lezard$M_IND)
```

```
## [1] 0.0003305068
```

```r
# variance de la distribution

# NB: la variance d'une distribution correspond à son moment d'ordre 2


sd(data_lezard$M_IND)
```

```
## [1] 0.01817985
```

```r
# écart-type (standard deviation) de la distribution

mad(data_lezard$M_IND)
```

```
## [1] 0.014826
```

```r
# median absolute deviation (valeur médiane des écarts absolus à la médiane),
# cette métrique de dispersion est très peu sensible aux outliers (points
# extrêmes) contrairement à la variance ou à l'écart-type (analogue à la
# comparaison moyenne-médiane)




summary(data_lezard$M_IND)
```

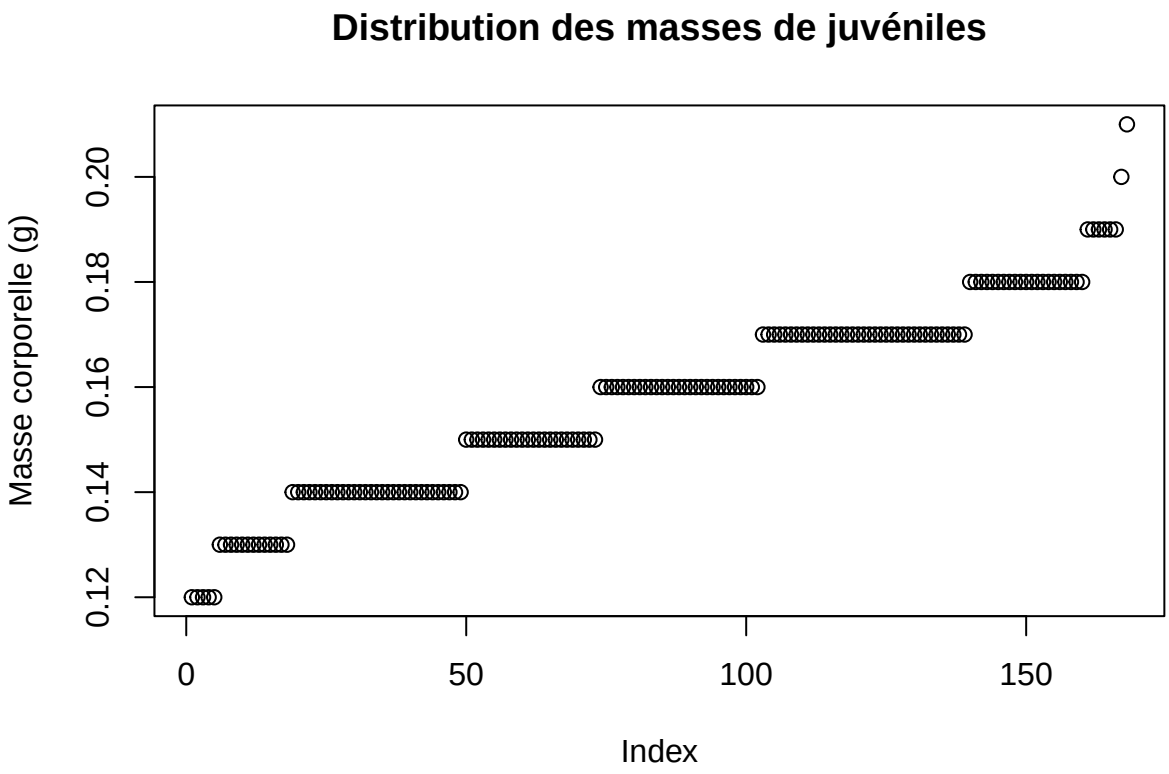
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1200  0.1400  0.1600  0.1577  0.1700  0.2100
```

```r
# résumé des principaux paramètres de position de la distribution (en y
# ajoutant l'écart-type on obtient les principaux descripteurs usuels d'une
# distribution quantitative)
```

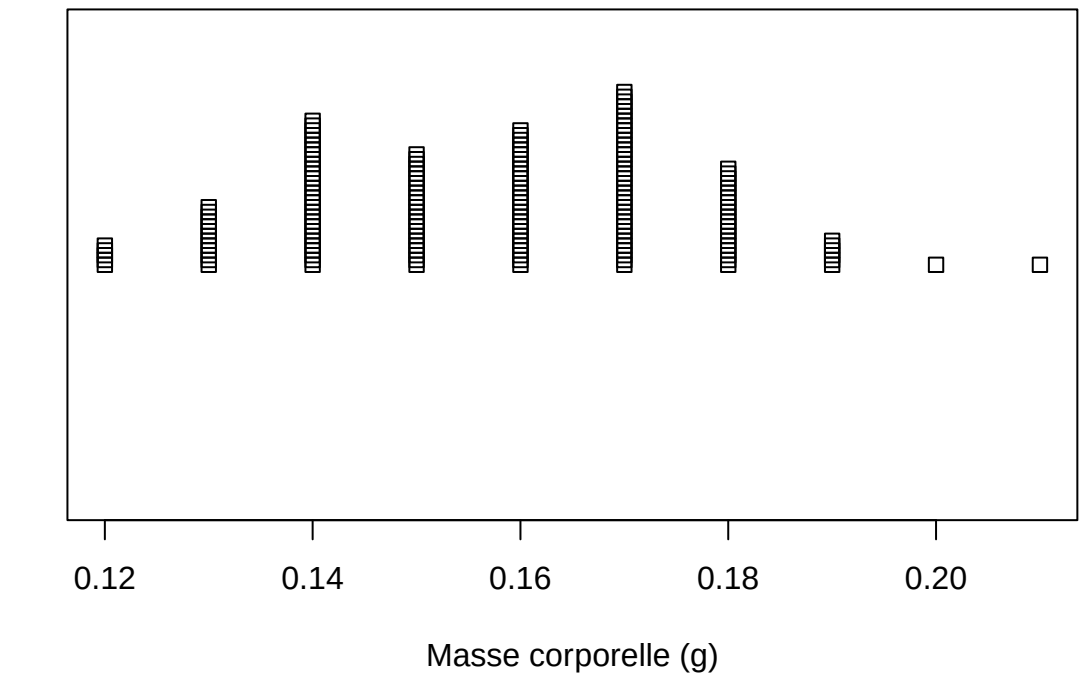
Descriptions graphiques

Nous allons maintenant décrire graphiquement la distribution que nous venons d'étudier. Les deux modes graphiques les plus utilisés pour la visualisation de variables quantitatives sont les **histogrammes** (permettant d'avoir une idée du type de distribution de probabilité que peut suivre la variable et sa forme globale) et les **boxplots** qui permettent d'avoir une visualisation rapide des paramètres de position et de dispersion de la variable. D'autres visualisation graphiques alternatives (par nuage de points) permettent également de décrire la distribution. Voici les principaux exemples de sorties graphiques ci-dessous:

```r
plot(
  sort(data_lezard$M_IND),
  main = "Distribution des masses de juvéniles",
  ylab = "Masse corporelle (g)"
  )
```

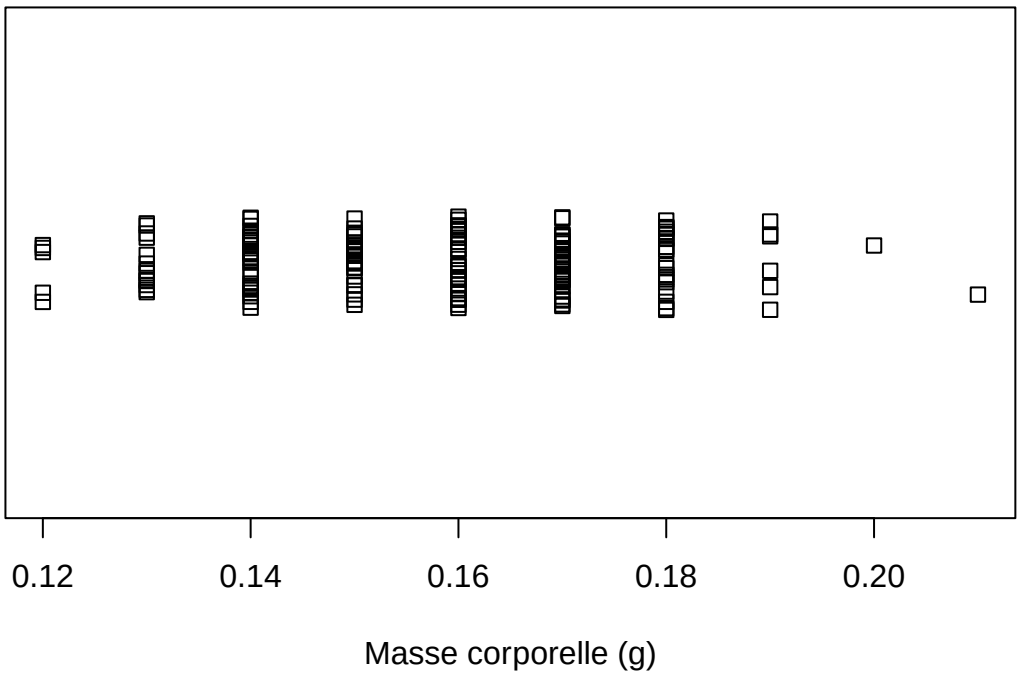



```r
# graphe des valeurs ordonnées de manière croissantes
# (Rq: ne pas oublier l'unité en légende d'axe !)



# autre mode visualisation par "dot plot" (ou "scatter plot"):

stripchart(
  data_lezard$M_IND,
  method = "stack",
  main = "Distribution des masses de juvéniles (empilements des points)",
  xlab = "Masse corporelle (g)",
  offset = 1/8  # zoom in plot
  )
```

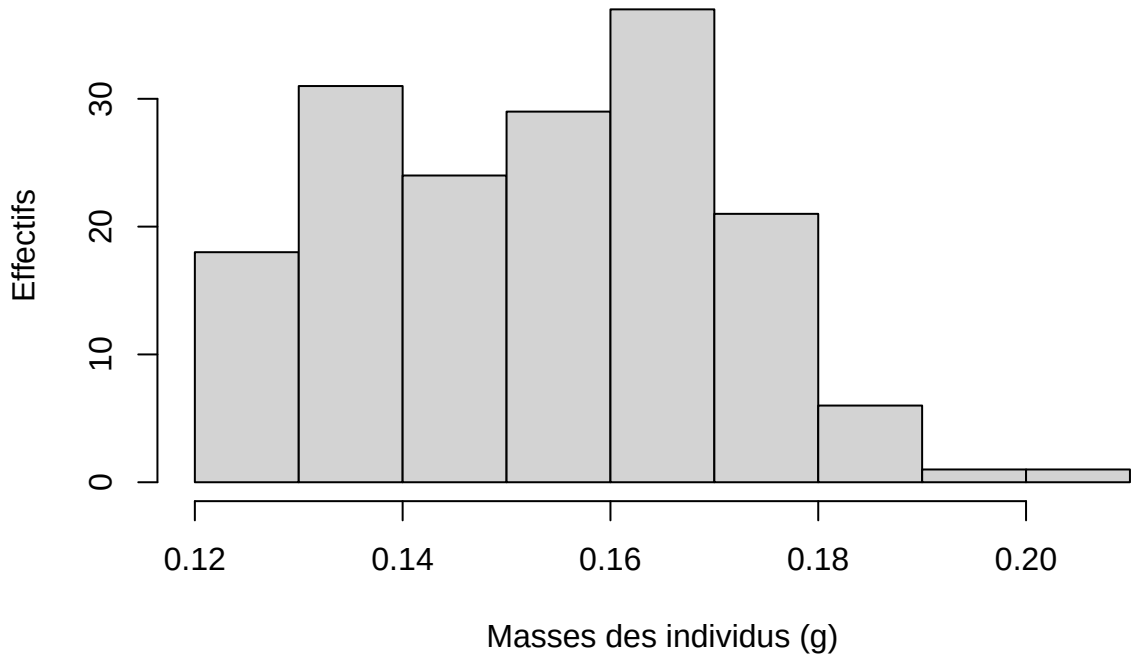



```r
stripchart(
  data_lezard$M_IND,
  method = "jitter",
  main = "Distribution des masses de juvéniles (instabilités des points)",
  xlab = "Masse corporelle (g)"
  )
```

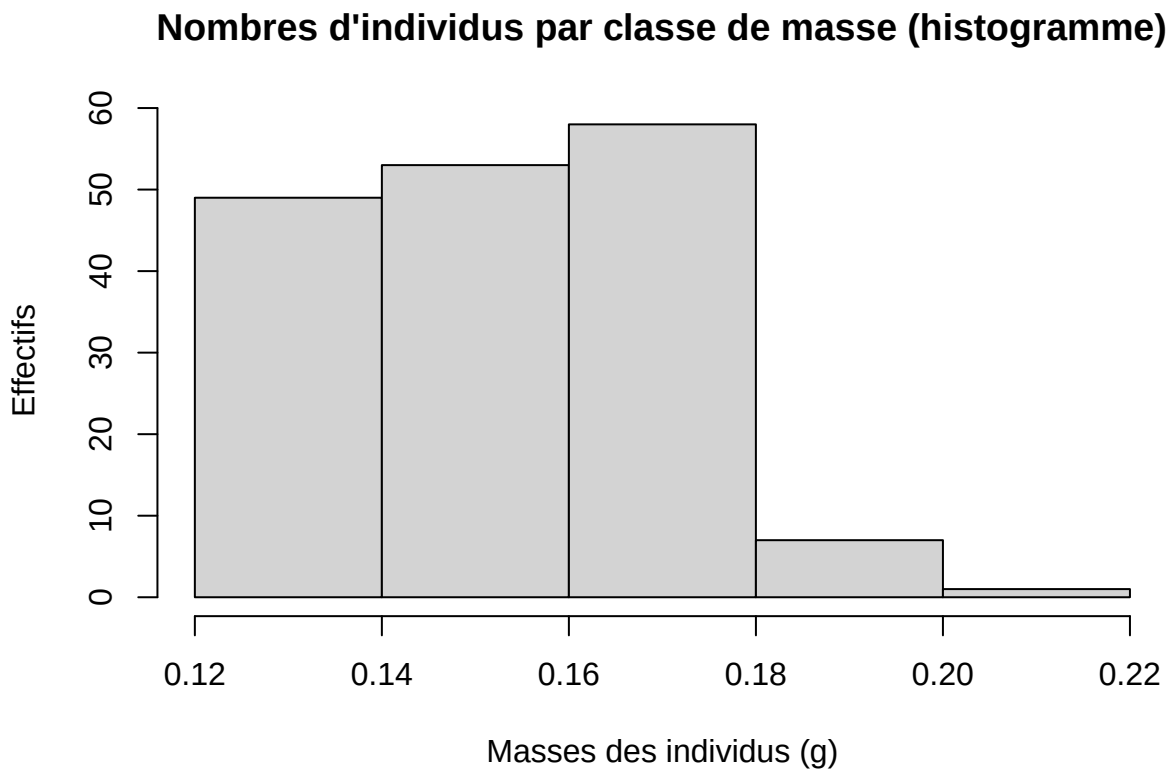
Distribution des masses de juvéniles (instabilités des points)



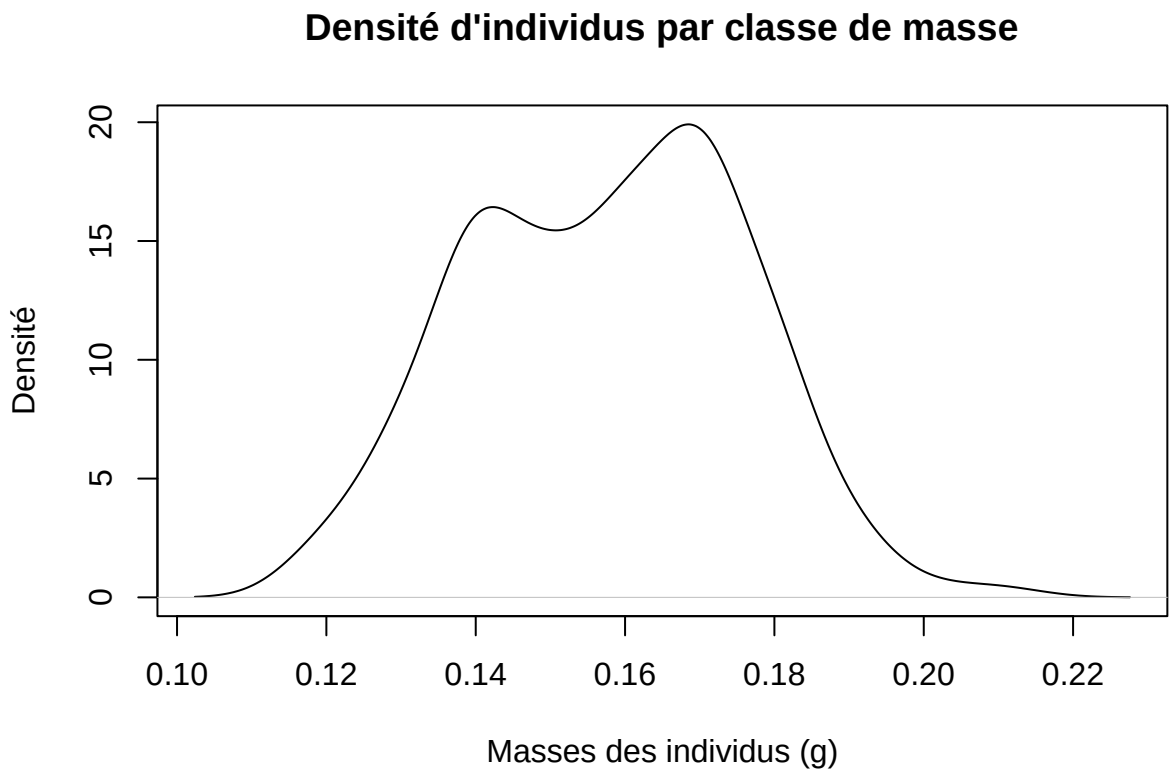

```r
# histogramme et kernels de densité de la distribution:

hist(
  data_lezard$M_IND,
  main = "Nombres d'individus par classe de masse (histogramme)",
  xlab = "Masses des individus (g)",
  ylab = "Effectifs"
  )
```



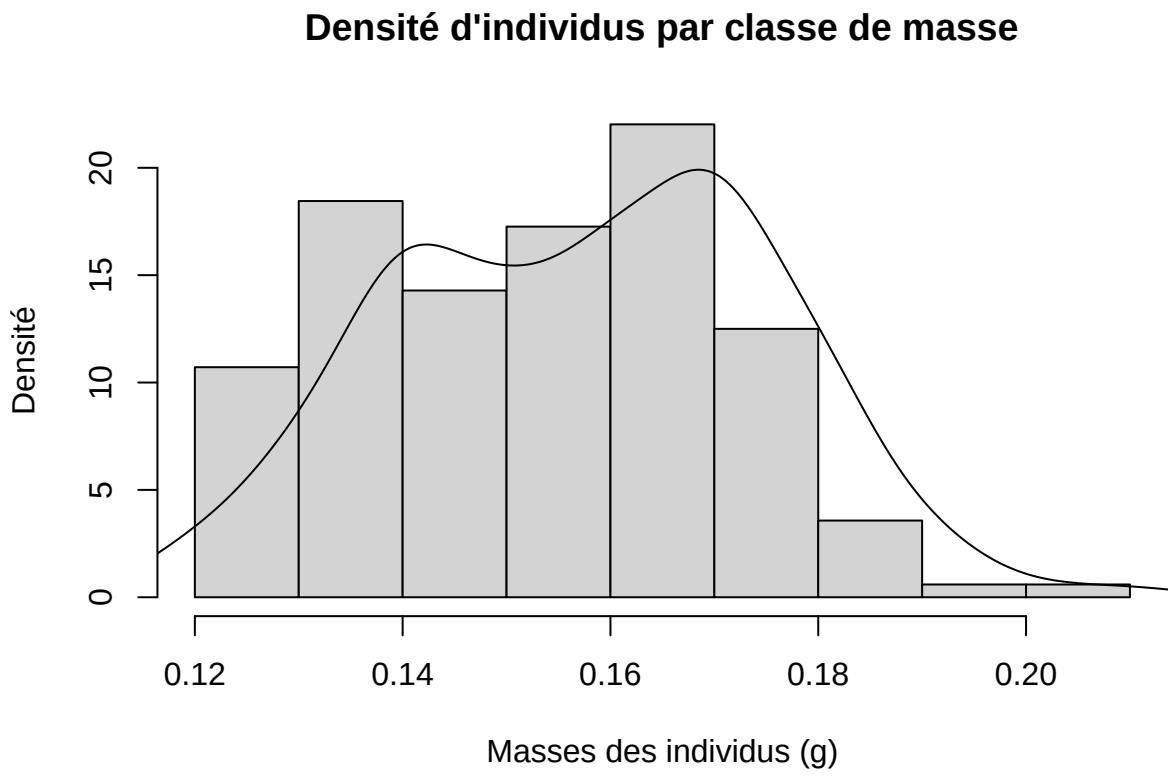

```r
# par défaut visualisation cherchant à optimiser la lisibilité

hist(
  data_lezard$M_IND,
  breaks = seq(0.12, 0.22, 0.02),
  main = "Nombres d'individus par classe de masse (histogramme)",
  xlab = "Masses des individus (g)",
  ylab = "Effectifs"
  )
```



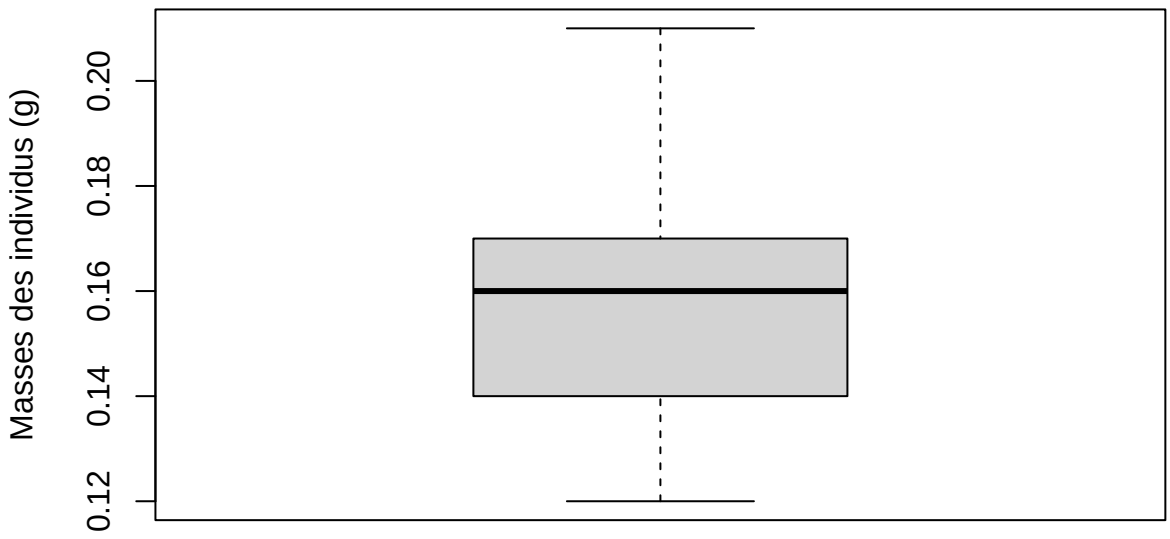

```r
# possibilité de changer manuellement les classes de regroupement pour la visualisation
# (on peut également renseigner le nombre de classe souhaité)

plot(
  density(data_lezard$M_IND),
  main = "Densité d'individus par classe de masse",
  xlab = "Masses des individus (g)",
  ylab = "Densité"
  )
```




```r
# graphe du kernel de densité
# (estimation de l'aire sous la courbe de distribution, avec une aire égale à 1)


# il est possible de combiner des graphes sur une même sortie:

hist(
  data_lezard$M_IND,
  prob = T,
  main = "Densité d'individus par classe de masse",
  xlab = "Masses des individus (g)",
  ylab = "Densité"
  )
# histogramme, avec une modification des valeurs d'effectifs en ordonnée
# tel que l'aire sous la courbe soit égale à 1 (argument "prob = T")
lines(density(data_lezard$M_IND))
```




```
# ajout d'une ligne correspondant à la courbe de densité
# (il est également possible d'ajouter des points avec la fonction "points()"
# ou des graphes entiers en utilisant l'argument "add = T" dans la fonction graphique)




# boîte à moustaches (boxplots):


boxplot(
  data_lezard$M_IND,
  main = "Distribution des masses corporelles des juvéniles (boxplot)",
  ylab = "Masses des individus (g)"
  )
```




```
# pour rappel le boxplot donne les positions (de haut en bas):
# du maximum, du troisième quartile, de la médiane, du premier quartile, du minimum.
# Lorsque les valeurs les plus extrêmes dépassent 1.5 fois la distance inter-quartile
# (en partant du premier ou troisième quartile) on considère les points comme des "outliers"
# et ils sont affichés à part, les barres horizontales les plus externes indiquent alors
# cette distance d'1.5 fois l'écart interquartile (au lieu du minimum et du maximum)
```

Les distributions quantitatives peuvent également être décrite à l'aide de **paramètres de forme**, ils visent à savoir si la distribution est **symétrique** ("skewness" en anglais) et à connaître son degré d'**aplatissement** ("kurtosis" en anglais). Ces caractéristiques de formes sont visibles graphiquement avec l'histogramme (voir paragraphe précédent) mais on peut aussi les formaliser de manière numérique à l'aide de coefficient tel que le coefficient de Fisher (voir exercice ci-dessous).

EXERCICE

- Re-créez les fonctions mean() et sd() par vous mêmes
  Rappels:

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2}$$

- Faites une description statistique et graphique de la masse des mères suivies dans l'étude (attention: une même mère peut apparaître plusieurs fois dans le tableau car certaines ont donné naissance à plusieurs nouveau-nés, veillez à bien éliminer les duplicats !)
- Comparez les deux valeurs suivantes: mean(c(1,2,3,4,50)) et median(c(1,2,3,4,50)), que pouvez-vous en conclure ? Comment décrire (à l'aide d'outils statistiques) la différence entre ces deux distributions ?
- Créez une fonction permettant de calculer les moments d'ordre n
  Rappel:

$$\mu_n = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^n$$

- Créez une fonction permettant de calculez les coefficients de Fisher de symétrie (skewness) et d'aplatissement (kurtosis), sachant qu'ils sont définis de la manière suivante:

$$\gamma_{skewness} = \frac{\mu_3}{s^3}$$

$$\gamma_{kurtosis} = \frac{\mu_4}{s^4} - 3$$

- Si $\gamma_{skewness} = 0$ la distribution est symétrique, si $\gamma_{skewness} > 0$ la courbe est étalée à droite, si $\gamma_{skewness} < 0$ la courbe est étalée à gauche. Si $\gamma_{kurtosis} = 0$ la distribution est mésokurtique (typique d'une loi gaussiennne), si $\gamma_{kurtosis} > 0$ la courbe est leptokurtique (piquée), si $\gamma_{kurtosis} < 0$ la courbe est platykurtique (aplatie). Que pouvez-vous dire de la symétrie et de l'aplatissement de la distribution des masses chez les mères suivies ?

```r
mean_bis <- function(x) {
  sum(x) /
    length(x)
}

sd_bis <- function(x) {
  sqrt(
    sum((x - mean(x))^2) /
      (length(x) - 1)
  )
}



summary(data_lezard[!duplicated(data_lezard$ID_MOTHERS),]$M_MOTHERS)

sd(data_lezard[!duplicated(data_lezard$ID_MOTHERS),]$M_MOTHERS)
```

```r
hist(
  data_lezard[!duplicated(data_lezard$ID_MOTHERS),]$M_MOTHERS,
  main = "Nombres de mères par classe de masse (histogramme)",
  xlab = "Masse corporelle (g)",
  ylab = "Effectifs"
  )
```

```r
boxplot(
  data_lezard[!duplicated(data_lezard$ID_MOTHERS),]$M_MOTHERS,
  main = "Distribution des masses corporelles des mères (boxplot)",
  ylab = "Masse corporelle (g)"
  )
```

```r
mean(c(1,2,3,4,50))
median(c(1,2,3,4,50))
# sensibilité de la moyenne aux point extrêmes par rapport à la médiane



moment_n <- function(x, n) {
    sum((x - mean(x))^n) /
      (length(x) - 1)
}



gamma_skewness <- function(x) {
  moment_n(x, 3) /
    sd(x)^3
}


gamma_kurtosis <- function(x) {
  moment_n(x, 4) /
    sd(x)^4 - 3
}


gamma_skewness(data_lezard[!duplicated(data_lezard$ID_MOTHERS),]$M_MOTHERS)
gamma_kurtosis(data_lezard[!duplicated(data_lezard$ID_MOTHERS),]$M_MOTHERS)
# la courbe de distribution est asymétrique (étalée vers la droite) et platykurtique (aplatie)

hist(
  breaks=seq(2.5, 5, 0.25),
  data_lezard[!duplicated(data_lezard$ID_MOTHERS),]$M_MOTHERS,
  main = "Nombres de mères par classe de masse (histogramme)",
  xlab = "Masse corporelle (g)",
  ylab = "Effectifs"
  )
```

```
# plus visible sur cette figure
```

Bilan

Nous avons étudié comment décrire statistiquement et graphiquement des variables qualitatives et quantitatives sous R.
Les fonctions clés pour décrire une variable **qualitative** sont les suivantes:

- **table()**: table d'effectifs
- **barplot()**: graphe en barres

Les fonctions clés pour décrire une variable **quantitative** sont les suivantes:

- **summary()**: résumé statistique de la distribution avec moyenne (fonction **mean**), médiane (fonction **median**), quartile (fonction **quantile**) et min/max
- **sd()**: écart-type de la distribution
- **hist()**: histogramme de la distribution
- **boxplot()**: graphe "boîte à moustache"

Ces différentes applications nous ont permis d'aborder la **création de fonction** et les **sorties graphiques** sous R. Les principaux arguments (options) graphiques à retenir sont les suivants:

- **main**: titre
- **xlab**: nom de l'axe des abscisses
- **ylab**: nom de l'axe des ordonnées

Une fonction se définit de la manière suivante: **Nom <- function(arg) {commandes(arg)}**.

MISE EN APPLICATION

Dans l'espace E-learn vous trouverez un autre jeu de données appelé "Interactions_dauphins_bateaux.txt". Cette table de données décrit le comportement de dauphins à proximité de bateaux (colonne boat.dist: "no"= pas de réponse, "approach"= s'approche du bateau, "avoidance"= s'éloigne du bateau, "response"= interagit avec le bateau) et peut être utile à des gestionnaires pour comprendre le potentiel dérangement généré par ces interactions. L'objectif est pour vous d'importer ce jeu de données dans R et d'utiliser les outils qui vous ont été présentés au cours de cette séance pour explorer ces données et vous les approprier. Voici quelques exemples ci-dessous d'objectifs que vous pouvez chercher à réaliser lors de votre exploration.

- Comment sont distribués les comportements et les réponses au cours de l'étude ?
- Quel est la distribution des tailles de groupes ? Comment peut-elle être décrite ?

Pensez bien à respectez les bonnes pratiques lorsque vous écrivez votre script pour explorer ce jeu de données, en gardant un espace de travail réduit au nécessaire et propre, en nommant correctement vos variables et objets R, en commentant bien votre code et en le structurant clairement.

POUR ALLER PLUS LOIN

Il existe une très grande communauté internationale travaillant sur le logiciel R, il est donc assez facile d'obtenir de l'aide en cas de blocage sur R. Une première source d'entraide sont les forums et tout particulièrement le forum "Stack Overflow" (en anglais), très actif sur le sujet. Il existe également certains documents résumant toutes les fonctions à connaître pour mener à bien des analyses statistiques comme le site [STHDA]http://www.sthda.com/french/ ou cet aide-mémoire pour les statistiques appliquées à la biologie: https://cran.r-project.org/doc/contrib/Herve-Aide-memoire-statistique.pdf (qui propose notamment un arbre de décision assez didactique pour orienter les choix d'analyses statistiques).