BIO4558 Biostatistiques appliquées avec R

Manuel de Laboratoire

Julien Martin

2020-04-01

Contents

Note

Version en cours de développement pour le cours de l'automne 2020

Préface

Les exercices de laboratoire que vous retrouverez dans les pages qui suivent sont conçus de manière à vous permettre de développer une expérience pratique en analyse de données à l'aide d'un logiciel (R). R est un logiciel très puissant, mais comme tous les logiciels, il a des limites. En particulier il ne peut réfléchir à votre place, vous dire si l'analyse que vous tentez d'effectuer est appropriée ou sensée, ou interpréter biologiquement les résultats.

Quelques points importants à retenir

- Avant de commencer une analyse statistique, il faut d'abord vous familiariser son fonctionnement. Cela ne veut pas dire que vous devez connaître les outils mathématiques qui la sous-tendent, mais vous devriez au moins comprendre les principes utilisés lors de cette analyse. Avant de faire un exercice de laboratoire, lisez donc la section correspondante dans les notes de cours. Sans cette lecture préalable, il est très probable que les résultats produits par le logiciel, même si l'analyse a été effectuée correctement, seront indéchiffrables.

- Les laboratoires sont conçus pour compléter les cours théoriques et vice versa. À cause des contraintes d'horaires, il se pourrait que le cours et le laboratoire ne soient pas parfaitement synchronisés. N'hésitez donc pas à poser des questions sur le labo en classe ou des questions théoriques au laboratoire.

- Travaillez sur les exercices de laboratoire à votre propre rythme. Certains exercices prennent beaucoup moins de temps que d'autres et il n'est pas nécessaire de compléter un exercice par séance de laboratoire. En fait deux séances de laboratoire sont prévues pour certains des exercices. Même si vous n'êtes pas notés sur les exercices de laboratoire, soyez conscient que ces exercices sont essentiels. Si vous ne les faites pas, il est très peu probable que vous serez capable de compléter les devoirs et l'examen final. Prenez donc ces exercices de laboratoire au sérieux !

- Le premier laboratoire est conçu pour vous permettre d'acquérir ou de réviser le minimum de connaissances requises pour vous permettre de réaliser les exercices de laboratoires avec R. Il y a presque toujours de multiples façons de faire les choses avec R et vous ne trouverez ici que des méthodes simples. Ceux et celles d'entre vous qui y sont enclins pourront trouver en ligne des instructions plus détaillées et complexes. En particulier, je vous conseille :
  - R pour les débutants http://cran.r-project.org/doc/contrib/Paradis-rdebuts_fr.pdf
  - Using R for psychological research: A simple guide to an elegant package http://www.personality-project.org/r/
  - An introduction to R http://cran.r-project.org/doc/manuals/R-intro.html
  - Si vous préférez des manuels, le site web de CRAN en garde une liste commentée à : http://www.r-project.org/doc/bib/R-books.html
  - Finalement, comme aide-mémoire à garder sous la main, je vous recommande R reference card par Tom Short http://cran.r-project.org/doc/contrib/Short-refcard.pdf

Qu'est-ce que R et pourquoi l'utiliser dans ce cours?

R est un logiciel libre et multiplateforme formant un système statistique et graphique. R est également un langage de programmation spécialisé pour les statistiques. C'est un dialecte du langage S. S-Plus est un autre dialecte, très semblable, et forme un produit commercial qui a un interface graphique que certains trouvent plus convivial.

R a deux très grands avantages pour ce cours, et un inconvénient embêtant initialement mais qui vous forcera à acquérir des excellentes habitudes de travail. Le premier avantage est que vous pouvez tous l'installer sur votre (ou vos) ordinateurs personnel gratuitement. C'est important parce que c'est à l'usage que vous apprendrez et maîtriserez réellement les biostatistiques et cela implique que vous devez avoir un accès facile et illimité à un logiciel statistique. Le deuxième avantage est que R peut tout faire en statistiques. R est conçu pour être extensible et est devenu l'outil de prédilection des statisticiens mondialement. La question n'est plus : " Est-ce que R peut faire ceci? “, mais devient” Comment faire ceci avec R ". Et la recherche internet est votre ami. Aucun autre logiciel n'offre ces deux avantages.

L'inconvénient embêtant initialement est que l'on doit opérer R en tapant des instructions (ou en copiant des sections de code) plutôt qu'en utilisant des menus et en cliquant sur différentes options. Si on ne sait pas quelle commande taper, rien ne se passe. Ce n'est donc pas facile d'utilisation à priori. Cependant, il

est possible d'apprendre rapidement à faire certaines des opérations de base (ouvrir un fichier de données, faire un graphique pour examiner ces données, effectuer un test statistique simple). Et une fois que l'on comprend le principe de la chose, on peut assez facilement trouver sur le web des exemples d'analyses ou de graphiques plus complexes et adapter le code à nos propres besoins. C'est ce que vous ferez dans le premier laboratoire pour vous familiariser avec R.

Pourquoi cet inconvénient est-il d'une certaine façon un avantage? Parce que vous allez sauver du temps en fin de compte. Garanti. Croyez-moi, on ne fait jamais une analyse une seule fois. En cours de route, on découvre des erreurs d'entrée de données, ou que l'on doit faire l'analyse séparément pour des sous-groupes, ou on obtient des données supplémentaires, ou on fait une erreur. On doit alors recommencer l'analyse. Avec une interface graphique et des menus, cela implique recommencer à cliquer ici, entre des paramètres dans des boîtes et sélectionner des boutons. Chaque fois avec possibilité d'erreur. Avec une série de commandes écrites, il suffit de corriger ce qui doit l'être puis de copier-coller l'ensemble pour répéter instantanément. Et vous avez la possibilité de parfaitement documenter ce que vous avez fait. C'est comme cela que les professionnels travaillent et offrent une assurance de qualité de leurs résultats.

Installation

Pour installer R sur un nouvel ordinateur, allez au site http://cran.r-project.org/. Vous y trouverez des versions compilées (binaries) ou non (sources) pour votre système d'exploitation de prédilection (Windows, MacOS, Linux).

Note : R a déjà été installé sur les ordinateurs du laboratoire (la version pourrait être un peu plus ancienne, mais cela devrait être sans conséquences).

Instructions générales pour les laboratoires

- Apporter une clé USB ou son équivalent à chaque séance de labo- ratoire pour sauvegarder votre travail.
- Lire l'exercice de laboratoire AVANT la séance, lire le code R cor- respondant et préparer vos questions sur le code.
- Durant les pré-labs, écouter les instructions et posez vos questions au moment approprié.
- Faites les exercices du manuel de laboratoire à votre rythme, en équipe, puis je vous recommande de commencer (compléter?) le devoir. Profitez de la présence du démonstrateur et du prof...
- Pendant vos analyses, copiez-collez des fragments de sorties de R dans un document (par exemple dans votre traitement de texte favori). Annotez abondamment.

- À chaque fois que vous fermez R, sauvegardez l'historique de vos commandes (ex: labo1.1.rHistory, labo1.2.rHistory, etc). Vous pourrez ainsi refaire le labo instantanément, récupérer des fragments de code, ou plus facilement identifier les erreurs dans vos analyses.
- Créez votre propre librairie de fragments de codes (snippets). Annotez-là abondamment. Vous vous en féliciterez plus tard.

Et pour les dilettantes ou ceux qui sont allergiques à la programmation...

Je vous conseille d'essayer d'apprivoiser R sans interface graphique. Au début ce sera peut-être difficile, mais cela vous forcera à acquérir de bonnes habitudes. Si vous trouvez cela vraiment trop lourd, ou que vous vous heurtez à des problèmes récurrents qui vous font perdre trop de temps, alors vous pouvez installer (du moins sur votre propre ordi, je ne sais pas si vous pouvez au laboratoire) une interface graphique distribuée par CRAN et qui fonctionne sous Windows, Mac et Linux (à ce qu'il paraît, je n'ai essayé que la version Windows). Il s'agit de Rcmdr, que vous devez d'abord installer dans R. Vous devez avoir accès à l'Internet. Dans la fenêtre de commande, entrez la commande

```
install.packages("Rcmdr", dependencies=TRUE)
```

Une fenêtre s'ouvrira vous permettant de choisir l'un des sites distribuant Rcmdr. Le téléchargement et l'installation prendront plusieurs minutes car plusieurs autres packages doivent être téléchargées et installées. Quand ce sera terminé, vous pourrez démarrer l'interface graphique en entrant la commande: library(Rcmdr) Une nouvelle fenêtre s'ouvrira, avec des menus pour accéder à la majorité des commandes usuelles. Une des sous-fenêtres de Rcmdr est un historique des commandes R correspondant à vos choix dans les menus. Étudiez-les pour apprendre, peut-être plus facilement, le language R.

Chapter 1

Introduction à R

Après avoir complété cet exercice de laboratoire, vous pourrez : - Ouvrir des fichiers de données R déjà existants - Importer des ensembles de données rectangulaires - Exporter des donnes de R vers un fichier texte - Vérifier si les données ont été correctement importées - Examiner la distribution des observations d'une variable - Examiner visuellement et tester la normalité d'une variable - Calculer des statistiques descriptives d'une variable - Effectuer des transformations de données

1.1 Importer et exporter des données

1.1.1 Ouvrir et sauvegarder un fichier de données en format R

Les données pour les exercices de laboratoire et pour les devoirs vous sont fournies déjà en format R (fichiers avec extension .Rdata). Pour ouvrir ces fichiers, vous pouvez cliquer dessus et laisser votre système d'exploitation démarrer une nouvelle session de R ou encore, à partir de la console de R, taper sur une ligne de commande :

```
load(file.choose())
```

ce qui ouvrira une boîte de dialogue vous permettant d'aller choisir un fichier sur votre ordinateur. Après avoir fait votre sélection (choisir le fichier `ErablesGatineau.Rdata`), vous reviendrez à la fenêtre Rconsole sans changement apparent.

Vous pouvez aussi utiliser la commande suivante directement.

```
load("data/ErablesGatineau.Rdata")
```

Pour vérifier si les données ont bel et bien été lues, vous pouvez lister les objets en mémoire avec la fonction `ls()` ou en obtenir une liste avec une description plus détaillée avec `ls.str()`

```
ls()
```

```
## [1] "ErablesGatineau"
```

```
ls.str()
```

```
## ErablesGatineau : 'data.frame':  100 obs. of  3 variables:
##  $ station: chr  "A" "A" "A" "A" ...
##  $ diam   : num  22.4 36.1 44.4 24.6 17.7 ...
##  $ biom   : num  732 1171 673 1552 504 ...
```

R confirme avoir en mémoire l'objet `ErablesGatineau`. `ErableGatineau` est un tableau de données rectangulaire (data.frame) contenant 100 observations (lignes) de 3 variables (colonnes): `station`, une variable de type Facteur avec 2 niveaux, et `diam` et `biom` qui sont 2 variables numériques.

1.1.2 Entrer des données

R n'est pas un environnement idéal pour entrer des données. C'est possible, mais la syntaxe est lourde et m'incite à m'arracher les cheveux. Utilisez votre chiffrier préféré pour faire l'entrée de données. Ce sera plus efficace et moins frustrant.

1.1.3 Nettoyer/corriger des données

Une autre opération qui peut être frustrante en R. Mon conseil : ne le faites pas là. Retournez au fichier original, faites la correction là, puis re-exportez les données vers R. Il est finalement plus simple de refaire exécuter les quelques lignes de code par la machine. Vous aurez à la fin une seule version (corrigée) de vos données et un code qui vous permet de refaire votre analyse.

1.1.4 Importer des données à partir d'Excel.

Sauvegarder la matrice rectangulaire de données en dans un fichier en format csv (File>Save as). Importer ces données en R avec la commande `read.csv()`. Par exemple, pour créer un base de données appelé age à partir d'un fichier csv préalablement sauvegardé. Essayez avec le fichier age.csv exporté du fichier excel age.xls, soit en utilisant `file.choose()`, soit en nommant directement le fichier.

```
age <- read.csv(file.choose())
```

```
age <- read.csv("data/age.csv")
```

Attrape : Attention si vous travaillez dans une langue utilisant la virgule au lieu du point décimal. Par défaut, R utilise le point décimal et vous n'obtiendrez pas le résultat escompté. Il existe une version modifiée de `read.csv()` appelée `read.csv2()` qui règle ce problème. Googlez-la si vous en avez besoin.

1.1.5 Exporter des données à partir de R.

Vous pouvez utiliser la fonction, `{r write, eval =FALSE) write.csv(mydata, file = "outfilename.csv", row.names = FALSE)` où mydata est le nom du base de données à exporter et `outfilename.csv` est le nom du fichier à produire. Notez que ce fichier sera créé dans le répertoire de travail (qui peut être changé par le menu à `File>Change dir`, ou par la commande `setwd()`)

1.2 Examen préliminaire des données

La première étape de toute analyse est l'examen des données. Elle nous permet de découvrir si on a bien importé les données, si les nombres enregistrés sont possibles, si toutes les données ont bien été lues, etc. L'examen préliminaire des données permet souvent aussi d'identifier des observations suspectes, possiblement dûes à des erreurs d'entrée de donnée. Finalement, l'examen graphique préliminaire permet en général de visualiser les tendances principales qui seront confirmées par l'analyse statistique en tant que telle. Le fichier `sturgeon.Rdata` contient les données d'une étude effectuée sur les esturgeons de la rivière Saskatchewan. Ces données ont été récoltées, entre autres, pour examiner comment la taille des esturgeons varie entre les sexes (sex), les sites (location), et les années (year).

- Pour recommencer avec une ardoise vide, videz la mémoire de R de tout son contenu en tapant la commande `rm(list=ls())`
- Ouvrez le fichier `sturgeon.Rdata`.
- Pour obtenir un aperçu des éléments du fichier qui ont été chargés en mémoire, taper la commande `ls.str()`.

```
ls.str()
```

```
## age : 'data.frame':  18 obs. of  3 variables:
##  $ ageclass: Factor w/ 9 levels "0-9","10-19",..: 1 1 2 2 3 3 4 4 5 5 ...
##  $ sex     : Factor w/ 2 levels "female","male": 1 2 1 2 1 2 1 2 1 2 ...
##  $ count   : int  17619 17538 17947 18207 21344 21401 19138 18837 13135 12568 ...
## ErablesGatineau : 'data.frame':  100 obs. of  3 variables:
```

```
##  $ station: chr  "A" "A" "A" "A" ...
##  $ diam   : num  22.4 36.1 44.4 24.6 17.7 ...
##  $ biom   : num  732 1171 673 1552 504 ...
## sturgeon : 'data.frame': 186 obs. of  9 variables:
##  $ fklngth : num  37 50.2 28.9 50.2 45.6 ...
##  $ totlngth: num  40.7 54.1 31.3 53.1 49.5 ...
##  $ drlngth : num  23.6 31.5 17.3 32.3 32.1 ...
##  $ rdwght  : num  15.95 NA 6.49 NA 29.92 ...
##  $ age     : num  11 24 7 23 20 23 20 7 23 19 ...
##  $ girth   : num  40.5 53.5 31 52.5 50 54.2 48 28.5 44 39 ...
##  $ sex     : Factor w/ 2 levels "FEMALE","MALE": 2 1 2 1 2 1 1 2 2 2 ...
##  $ location: Factor w/ 2 levels "CUMBERLAND","THE_PAS": 2 2 2 2 2 2 2 2 2 2 ...
##  $ year    : Factor w/ 3 levels "1978","1979",..: 1 1 1 1 1 1 1 1 1 1 ...
```

1.2.1 Sommaire statistique

Pour un sommaire du contenu du base de données appelé sturgeon qui est en mémoire, taper la commande

```
summary(sturgeon)
```

```
##     fklngth         totlngth        drlngth          rdwght      
##  Min.   :24.96   Min.   :28.15   Min.   :14.33   Min.   : 4.73  
##  1st Qu.:41.00   1st Qu.:43.66   1st Qu.:25.00   1st Qu.:18.09  
##  Median :44.06   Median :47.32   Median :27.00   Median :23.10  
##  Mean   :44.15   Mean   :47.45   Mean   :27.29   Mean   :24.87  
##  3rd Qu.:48.00   3rd Qu.:51.97   3rd Qu.:29.72   3rd Qu.:30.27  
##  Max.   :66.85   Max.   :72.05   Max.   :41.93   Max.   :93.72  
##                  NA's   :85      NA's   :13      NA's   :4      
##       age             girth          sex            location     year    
##  Min.   : 7.00   Min.   :11.50   FEMALE:106   CUMBERLAND: 85   1978:45  
##  1st Qu.:17.00   1st Qu.:40.00   MALE  : 80   THE_PAS   :101   1979:68  
##  Median :20.00   Median :44.00                                 1980:73  
##  Mean   :20.24   Mean   :44.33                                          
##  3rd Qu.:23.50   3rd Qu.:48.80                                          
##  Max.   :55.00   Max.   :73.70                                          
##  NA's   :11      NA's   :85
```

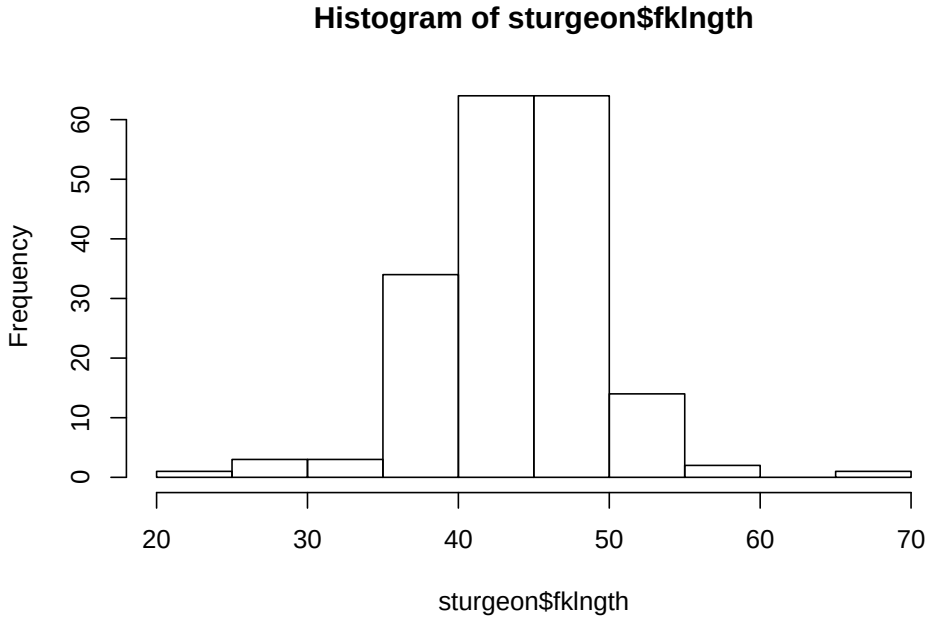
Pour chaque variable, R donne le minimum, le maximum, la médiane qui est la valeur au milieu de la liste des observations ordonnées (appelée le 50 ième percentile), ici, la 93 ième valeur des 186 observations, les valeurs au premier (25%) et troisième quartile (75%), et si il y a des valeurs manquantes dans la colonne. Notez que plusieurs des variables ont des observations manquantes (NA). Donc, seules les variables fklngth (longueur à la fourche), sex, location et year ont 186 observations.

Attrape : Attention aux valeurs manquantes. Plusieurs fonctions de R y réagissent mal et on doit souvent faire les analyses sur des sous- ensembles sans valeur manquante, par des commandes ou des options dans les commandes. On y reviendra, mais prenez l'habitude de noter mentalement si il y a des données manquantes et de vous en rappeler en faisant l'analyse.

1.2.2 Histogramme, densité de probabilité empirique, boxplot et examen visuel de la normalité

Examinons maintenant de plus près la distribution de fklngth. La commande `hist()` permet de tracer un histogramme de la variable fklngth dans le base de données sturgeon.

```
hist(sturgeon$fklngth)
```



Les données semblent suivre approximativement une distribution normale. C'est bon à savoir. Cette syntaxe est un peu lourde puisqu'on doit ajouter le préfixe sturgeon$ devant chaque nom de variable. On peut se faciliter la tâche en utilisant la commande `attach()` qui va nous donner accès directement accès aux variables contenues dans la base de données. **Cependent, cela est fortement déconseillé.**

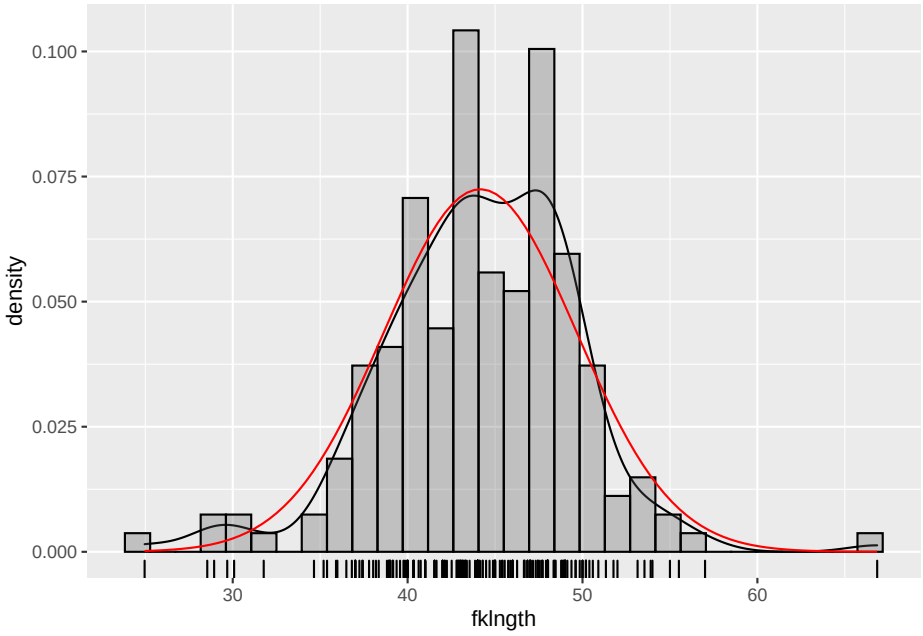
```
attach(sturgeon)
```

Cet histogramme est la représentation classique. Mais les histogrammes ne sont pas parfaits. Leur forme dépend en partie du nombre de catégories utilisées,

surtout pour les petits échantillons. On peut faire mieux, particulièrement si on est intéressé à comparer visuellement la distribution des observations à une distribution normale. Mais il faut programmer un peu (ou savoir copier-coller. . .)

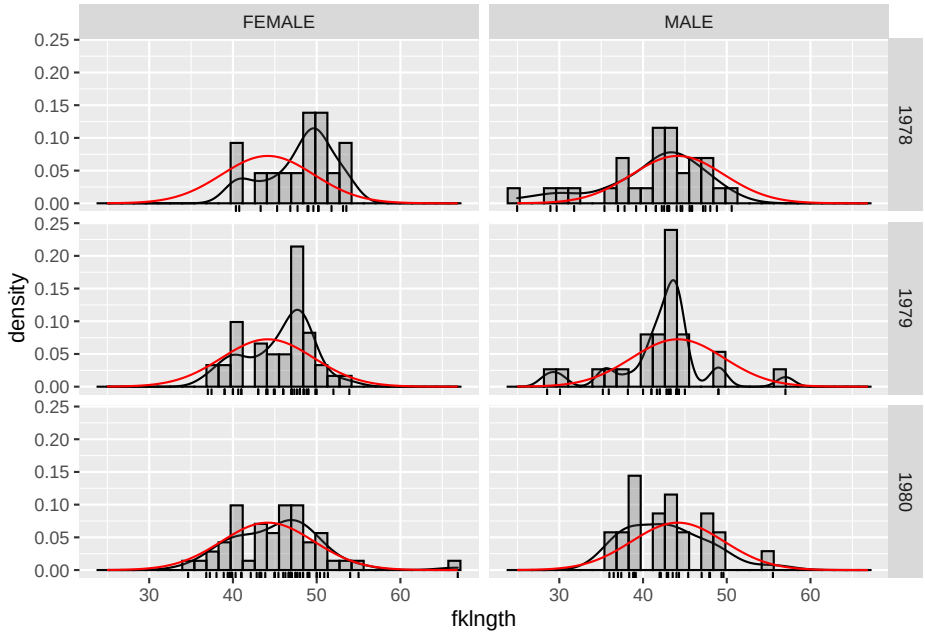
- Copiez-collez le code suivant dans une nouvelle fenêtre script (File->New script, ou Ctrl-n dans Windows), puis exécutez le.

```
library(ggplot2)
#  use "sturgeon" dataframe to make plot called mygraph
#  and define x axis as representing fklngth
mygraph <- ggplot(sturgeon, aes(x = fklngth))
#  add data to the mygraph ggplot
mygraph <- mygraph +
#  add data density smooth
    geom_density() +
#  add rug (bars at the bottom of the plot)
    geom_rug() +
#  add black semitransparent histogram
    geom_histogram(aes(y = ..density..), bins = 30, color = "black",   alpha = 0.3) +
#  add normal curve in red, with mean and sd from fklength
    stat_function(fun = dnorm,
                  args = list(
                    mean = mean(sturgeon$fklngth),
                    sd = sd(sturgeon$fklngth) ),
                  color = "red")
# display graph
    mygraph
```



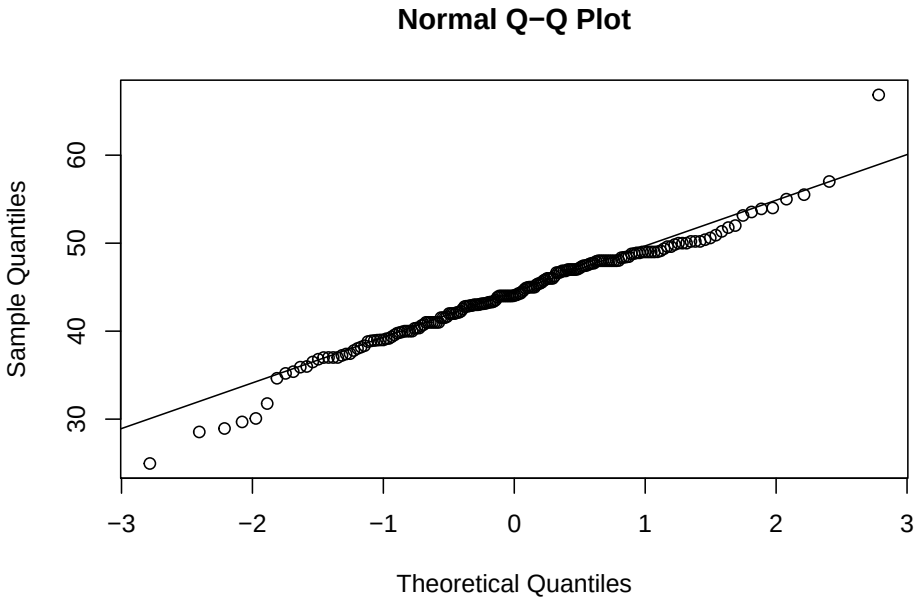
Chaque observation est représentée par une barre sous l'axe des x (rug). En rouge est la distribution normale de données avec la même moyenne et écart-type que les observations. Et l'autre ligne est la densité de probabilité empirique, « lissée » à partir des observations. Si vous êtes plus aventureux, vous pouvez examiner la distribution des observations de fklngth par sous-groupes (par exemple sex et year) avec :

```
mygraph + facet_grid(year ~ sex)
```



Chaque panneau illustre la distribution pour un sexe cette année-là, et la courbe en rouge récurrente représente la distribution normale pour l'ensemble des données. Cette courbe peut servir à mieux évaluer visuellement les différences entre les panneaux. Une autre façon d'évaluer la normalité de données visuellement est de faire un QQ plot avec la paire de commandes `qqnorm()` et `qqline()`.

```
qqnorm(sturgeon$fklngth)
qqline(sturgeon$fklngth)
```



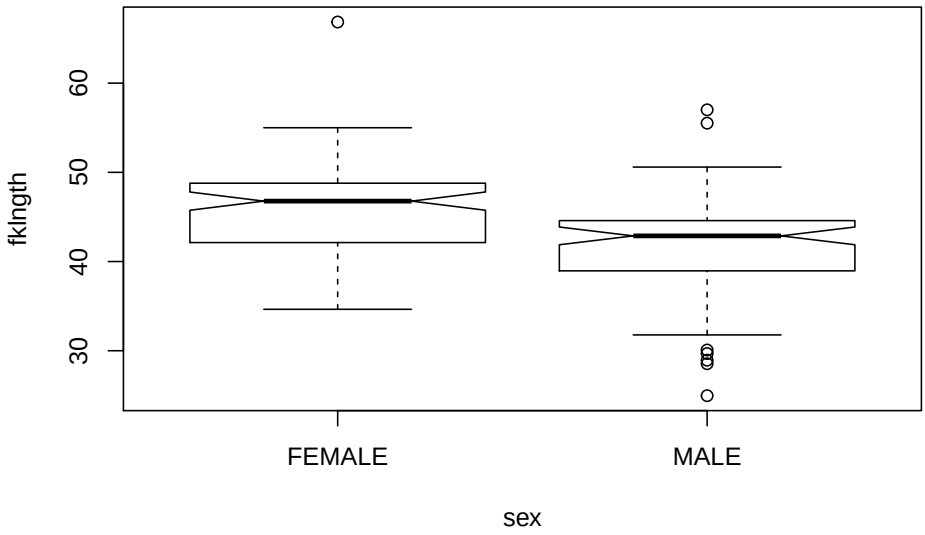
Des données parfaitement normales suivraient la ligne droite diagonale. Ici, il y a des déviations dans les queues de la distribution, et un peu à droite du centre. Comparez cette représentation à celle des deux graphiques précédents. Vous conviendrez sans doute avec moi qu'il est plus facile de visualiser comment la distribution dévie de la normalité sur les histogrammes et les graphiques de la densité empirique de probabilité que sur les QQ plots. Ceci dit, les QQ plots sont souvent utilisés et vous devriez être capable de les interpréter. De plus, on peut facilement éprouver statistiquement l'hypothèse que les données sont distribuées normalement avec R par la commande `shapiro.test()` qui calcule une statistique (W) qui est une mesure de la tendance des points d'un QQ plot à former une ligne parfaite. Si oui, alors W=1. Si W s'éloigne de 1 (vers 0), alors les données s'éloignent de la normalité. Ici,

```
shapiro.test(sturgeon$fklngth)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sturgeon$fklngth
## W = 0.97225, p-value = 0.0009285
```

W n'est pas très loin de 1, mais suffisamment pour que la différence soit significative. L'examen visuel des grands échantillons est souvent compliqué par le fait que plusieurs points se superposent et qu'il devient plus difficile de bien visualiser la tendance centrale. Les boxplots avec "moustaches" (box and whiskers plots) offrent une alternative intéressante. La commande `boxplot()` peut produire un boxplot de fklngth pour chaque niveau de sex, et ajoute les coches.

```
boxplot(fklngth ~ sex, data = sturgeon, notch = TRUE)
```



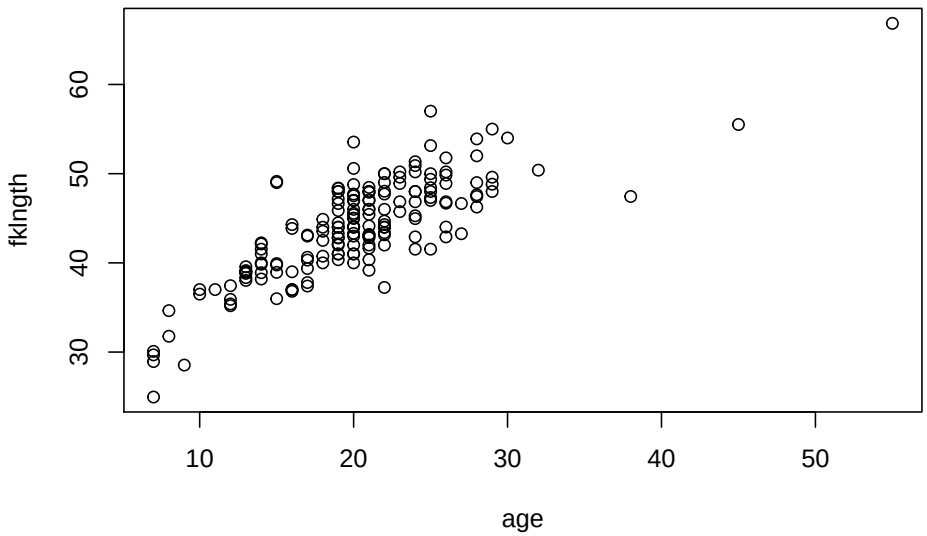
La ligne un peu plus épaisse dans la boîte de la Fig.7 indique la médiane. La coche est proportionnelle à l'incertitude quant à la position de la médiane. On peut visuellement interpréter approximativement les différences entre médianes en examinant si il y a chevauchement entre les coches (ici, il n'y a pas chevauchement, et on conclurait provisoirement que la médiane de fklngth pour les femelles est supérieure à celle des mâles). Les boîtes s'étendent du premier au troisième quartile (du 25ième au 75ième percentile si vous préférez), Les barres (moustaches ou whiskers) au-dessus et en dessous des boîtes s'étendent soit de la valeur minimum à la valeur maximum, ou, si il y a des valeurs extrêmes, de la plus petite à la plus grande valeur à l'intérieur de 1.5x la largeur de l'étendue interquartile . Enfin, les observations qui excèdent les limites des moustaches (donc à plus de 1.5x l'étendue interquartile de chaque côté de la médiane) sont indiquées par des symboles.Ce sont des valeurs qui pourraient être considérées comme extrêmes et possiblement aberrantes.

1.2.3 Diagrammes de dispersion bivariés

En plus des graphiques pour chacune des variables séparément, il est très souvent intéressant de jeter un coup d'oeil aux diagrammes de dispersion . La commande `plot(y~x)` permet de faire le graphique de y sur l'axe vertical (l'ordonnée) en fonction de x sur l'axe horizontal (l'abscisse).

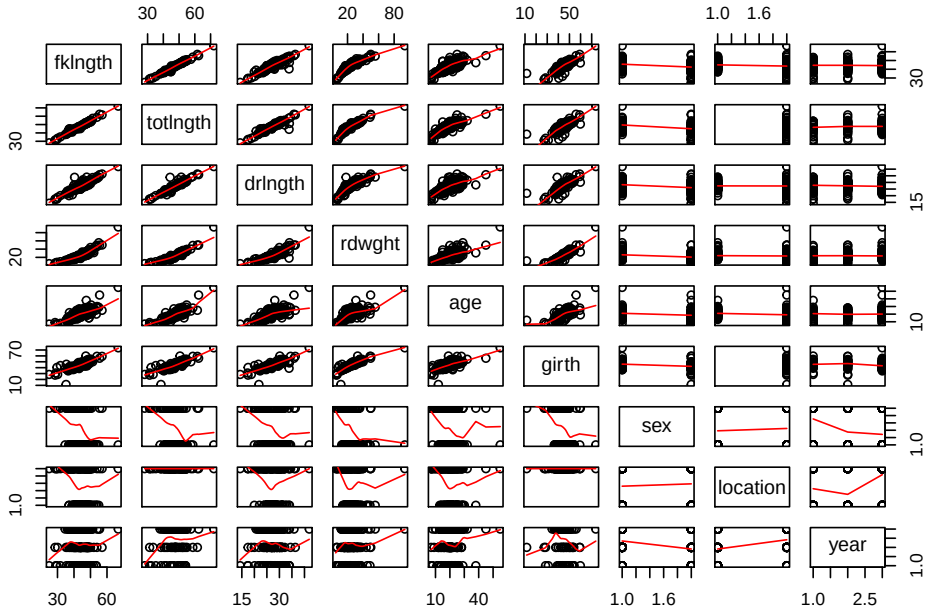
- Faites un graphique de fklngth en fonction de age avec la commande plot. Vous devriez obtenir:

```
plot(fklngth ~ age, data = sturgeon)
```



R a une fonction qui permet la création des graphiques de dispersion de toutes les paires de variables (`pairs()`). Une des option de ¬ est l'ajout d'une trace lowess qui indique la tendance de la relation entre les variables. Pour obtenir la matrice de ces graphiques avec la trace lowess pour toutes les variable dans sturgeon, entrer la commande `pairs(sturgeon, panel=panel.smooth)` et vous devriez obtenir

```
pairs(sturgeon, panel=panel.smooth)
```



1.3 Créer des sous-ensembles de cas

Il arrive fréquemment qu'une analyse se concentre sur un sous-ensemble des observations contenues dans un fichier de données. Les cas sont d'habitude sélectionnés selon un critère en particulier. Pour utiliser un sous-ensemble de vos données en créant un graphique ou en performant une analyse, on peut utiliser la commande `subset()`. Par exemple, pour créer un sous ensemble des données du tableau sturgeon qui ne contient que les femelles capturées en 1978, on peut écrire :

```
sturgeon.female.1978 <- subset(sturgeon, sex == "FEMALE" & year == "1978")
sturgeon.female.1978
```

```
##      fklngth totlngth  drlngth rdwght age girth    sex   location year
## 2   50.19685 54.13386 31.49606     NA  24  53.5 FEMALE    THE_PAS 1978
## 4   50.19685 53.14961 32.28346     NA  23  52.5 FEMALE    THE_PAS 1978
## 6   49.60630 53.93701 31.10236  35.86  23  54.2 FEMALE    THE_PAS 1978
## 7   47.71654 51.37795 33.97638  33.88  20  48.0 FEMALE    THE_PAS 1978
## 15  48.89764 53.93701 29.92126  35.86  23  52.5 FEMALE    THE_PAS 1978
## 105 46.85039       NA 28.34646  23.90  24    NA FEMALE CUMBERLAND 1978
## 106 40.74803       NA 24.80315  17.50  18    NA FEMALE CUMBERLAND 1978
## 107 40.35433       NA 25.59055  20.90  21    NA FEMALE CUMBERLAND 1978
## 109 43.30709       NA 27.95276  24.10  19    NA FEMALE CUMBERLAND 1978
## 113 53.54331       NA 33.85827  48.90  20    NA FEMALE CUMBERLAND 1978
## 114 51.77165       NA 31.49606  35.30  26    NA FEMALE CUMBERLAND 1978
```

```
## 116 45.27559       NA 26.57480  23.70  24    NA FEMALE CUMBERLAND 1978
## 118 53.14961       NA 32.67717  45.30  25    NA FEMALE CUMBERLAND 1978
## 119 50.19685       NA 32.08661  33.90  26    NA FEMALE CUMBERLAND 1978
## 123 49.01575       NA 29.13386  37.50  22    NA FEMALE CUMBERLAND 1978
```

Attrape: Dans ces comparaisons, il faut toujours utiliser `==` pour égal à. Dans ce contexte, si vous utilisez `=` seulement, vous n'obtiendrez pas ce que vous désirez. Dans le tableau qui suit se trouve une liste de commandes communes que vous allez probablement utiliser pour créer des expressions en R.

Operateur	Explication	Operateur	Explication
==	Égal à	!=	Pas égal à
>	Plus que	<	Moins que
>=	Plus que ou égal à	<=	Moins que ou égal à
&	Et vectorisé	|	Ou vectorisé
&&	Et contrôle	||	Ou contrôle
!	Pas		

- En utilisant les commandes `subset()` et `hist()`, essayez de faire un histogramme pour le sous-ensemble de cas correspondant aux femelles capturées en 1979 et 1980 (donc `sex == "FEMALE" & (year == "1979" | year == "1980")`)

1.4 Transformations de données

Il est très fréquemment nécessaire d'effectuer des transformations mathématiques sur les données brutes pour mieux satisfaire aux conditions d'application de tests statistiques. R étant aussi un langage de programmation complet, il peut donc effectuer les transformations désirées. Les fonctions les plus fréquemment utilisées sont:

- `log10()`
- `sqrt()`
- `ifelse()`

On peut employer ces fonctions directement dans les lignes de commandes, ou encore créer de nouvelles variables orphelines ou faisant partie d'un data.frame. Par exemple, pour faire un graphique du logarithme décimal de fklngth en fonction de l'âge, on peut écrire

```
plot(log10(fklngth)~age, data = sturgeon)
```

Pour créer une variable orpheline (i.e. non incluse dans le data.frame) appelée logfklngth et contenant le logarithme décimal de fklngth, on peut écrire

```
logfklngth <- log10(sturgeon$fklngth)
```

Si on veut ajouter cette variable transformée à un tableau de données (data.frame), alors, on doit préfixer le nom de la variable par le nom du base de données et du symbole `$`, par exemple, pour ajouter une variable nommée `lfkl` contenant le log10 de fklngth au tableau sturgeon, on peut écrire:

```
sturgeon$logfkl <- log10(sturgeonfklngth)
```

N'oubliez pas de sauvegarder ce tableau modifié si vous voulez avoir accès à cette nouvelle variable dans le futur. Pour les transformations conditionnelles, on peut utiliser la fonction `ifelse()`. Par exemple, pour créer une nouvelle variable appelée dummy qui sera égale à 1 pour les mâles et 0 pour les femelles, on peut écrire:

```
sturgeon$dummy <- ifelse(sturgeon$sex == "MALE", 1, 0)
```

1.5 Exercice

Vous trouverez dans le fichier salmonella, des valeurs numériques du ratio pour deux milieux (IN VITRO et IN VIVO) pour trois souches. Examinez les données pour ratio et faites des graphiques pour évaluer la normalité de la distribution des ratios pour la souche SAUVAGE.