Prise en main de **R**

M. L. Delignette-Muller
VetAgro Sup - LBBE

4 février 2021

le langage **R**

R est un langage de programmation dédié à l'analyse statistique de données.
Il est gratuit, très flexible, très utilisé dans le milieu scientifique au niveau mondial.
Il permet d'automatiser facilement l'analyse de données et d'en assurer la traçabilité.
Son utilisation requiert un apprentissage.

Démarrage de **R** via **Rstudio**

- Démarrez Rstudio.
- Spécifiez le répertoire de travail (celui où sont vos fichiers de données préalablement chargés depuis VetAgroTice).

Session. Set working Directory. Choose Directory

- Ouvrez un script

File. New file. R script.

- Sauvez-le sous un nom sans caractères spéciaux et terminant par .R

File. Save.

Principe d'utilisation d'un script

- Ecriture de **lignes de code** dans le script ET de **lignes de commentaires** commençant par le caractère # qui ne seront pas compilées.
- **Validation de chaque ligne de code** (ou d'un ensemble de lignes sélectionnées) en utilisant le bouton "Run" de Rstudio (en haut à droite de la fenêtre).
- Correction des lignes erronées directement dans le script
- Nouvelle validation des lignes de code après correction

. . . .

L'objectif est de terminer chaque séance avec un script commenté contenant un code propre facile à réutiliser.

Pour qu'on puisse vous encadrer facilement !

Afin qu'on puisse passer dans les rangs pour voir votre script et vous suivre à votre rythme, agrandissez suffisamment la taille du texte :

View. Zoom In.

ou Ctrl+

Exemple de script

```
# Lecture d'un jeu de données ("ENQ9697.txt")
# à l'aide de la fonction read.table()
# et affectation de la sortie de cette fonction
# à un objet de nom d
d <- read.table("ENQ9697.txt", header = TRUE,
                        stringsAsFactors = TRUE)

# résumé du jeu de données
summary(d)
```

Importation d'un jeu de données

Tapez le code suivant dans le script que vous avez ouvert et lancez-le en sélectionnant les 2 lignes puis en cliquant sur le bouton **Run** :

```
d <- read.table("ENQ9697.txt", header = TRUE,
                               stringsAsFactors = TRUE)
class(d)

## [1] "data.frame"
```

L'argument `header` est ici fixé à `TRUE` pour que la première ligne du fichier "ENQ9697.txt" soit reconnue comme indiquant les intitulés des colonnes (noms des variables).
L'argument `stringsAsFactors` est ici fixé à `TRUE` pour que les colonnes non numériques soient reconnues comme des variables qualitatives (factor).
L'objet créé `d` est de classe `data.frame`.

Structure du jeu de données

Tapez (dans le même script) et lancez le code suivant pour voir la structure du jeu de données et vérifier notamment la bonne classification des variables en quantitatives ou qualitatives :

- Variables quantitatives de classe `int` ou `numeric`
- Variables qualitatives de classe `Factor`

```
str(d)

## 'data.frame': 107 obs. of  7 variables:
##  $ SEXE    : Factor w/ 2 levels "F","M": 1 2 1 2 1 1 2 1 1
##  $ AGE     : int  22 21 19 20 19 21 21 19 20 22 ...
##  $ POIDS   : int  53 67 63 60 48 58 77 61 52 70 ...
##  $ TAILLE  : int  175 175 172 175 167 171 187 170 161 168 .
##  $ CADRE   : Factor w/ 2 levels "C","V": 2 1 2 2 2 1 1 1 1
##  $ DECISION: Factor w/ 3 levels "A","E","T": 2 1 1 1 1 2 1
##  $ FILIERE : Factor w/ 7 levels "A","C","E","I",..: 6 6 3 2
```

Affichage des première lignes du jeu de données

Tapez et lancez le code suivant pour voir les six premières lignes du jeu de données :

```
head(d)
```

```
##   SEXE AGE POIDS TAILLE CADRE DECISION FILIERE
## 1    F  22    53    175     V         E       R
## 2    M  21    67    175     C         A       R
## 3    F  19    63    172     V         A       E
## 4    M  20    60    175     V         A       C
## 5    F  19    48    167     V         A       R
## 6    F  21    58    171     C         E       A
```

Pour voir le jeu de données complet :

```
d
```

En résumé

- Un jeu de données au format texte peut être importé dans **R** à l'aide de la fonction `read.table()`.
- En pratique on utilisera le plus souvent un tableur pour saisir les données puis les sauvegarder au format texte ou CSV (cf. guide en ligne pour les détails : `http://www3.vetagro-sup.fr/ens/biostat/guideRM1.pdf`)
- Dans un jeu de données on a **une colonne par variable et une ligne par observation**.
- Chaque variable peut être **quantitative ou qualitative**. Il est indispensable de coder numériquement les variables quantitatives, et préférable de coder par une chaîne de caractères les variables qualitatives afin qu'elles soient automatiquement reconnues comme telles par **R** en mettant l'argument `stringsAsFactors` à `TRUE`.

Quelques compléments

- On ne peut pas laisser des trous dans un jeux de données. Les **données manquantes doivent être codées par `NA`**.
- Mieux vaut **éviter les accents, caractères spéciaux, espaces dans les noms** de colonne et les noms de modalités des variables qualitatives, pour éviter les problèmes liés à des différences entre systèmes d'exploitation, notamment en terme d'encodage des accents.

Sélection d'une colonne (une variable)

Tapez et lancez le code suivant pour accéder à la variable taille en utilisant le nom de la variable :

```
d$TAILLE
```

```
##   [1] 175 175 172 175 167 171 187 170 161 168 176 170 165 1
##  [16] 161 163 155 169 162 171 173 160 166 166 165 178 178 1
##  [31] 163 165 171 165 171 160 162 160 163 152 168 162 163 1
##  [46] 158 170 157 157 182 183 168 180 175 183 190 185 171 1
##  [61] 160 166 160 169 171 168 175 165 168 178 170 180 178 1
##  [76] 173 161 158 160 185 178 190 164 178 183 190 178 175 1
##  [91] 173 160 163 178 182 175 180 175 182 170 163 168 160 1
## [106] 182 180
```

Vérifiez que vous obtenez le même résultat en utilisant le numéro de la colonne :

```
d[,4]
```

Application de fonctions prédéfinies

Quelques statistiques :

```
mean(d$TAILLE)

## [1] 171

sd(d$TAILLE)

## [1] 8.47

median(d$TAILLE)

## [1] 170

quantile(d$TAILLE, probs = c(0.25, 0.75))

## 25% 75%
## 164 178
```

Création d'une nouvelle variable

Dans **R** les calculs sont vectorialisés, c'est-à-dire automatiquement réalisés sur toutes les lignes.
Utilisez le code suivant pour ajouter au jeu de données une variable avec les indices de masse corporelle

```
d$IMC <- d$POIDS / (d$TAILLE/100)^2
d$IMC
```

```
##   [1] 17.3 21.9 21.3 19.6 17.2 19.8 22.0 21.1 20.1 24.8 20.
##  [13] 19.1 23.9 22.5 18.9 19.6 19.6 16.8 23.6 22.9 19.0 19.
##  [25] 21.4 18.4 18.9 23.7 20.1 19.6 21.5 19.5 20.2 19.8 21.
##  [37] 20.6 21.1 19.9 23.8 19.5 18.3 21.1 17.4 20.3 20.4 18.
##  [49] 23.1 21.7 22.1 20.5 20.1 21.9 22.4 19.9 24.8 20.5 19.
##  [61] 19.9 20.0 20.3 20.0 20.5 21.3 23.8 20.2 19.1 20.5 20.
##  [73] 22.1 21.0 19.9 19.4 21.2 21.6 19.5 23.4 19.6 20.8 19.
##  [85] 22.4 23.5 18.3 22.9 22.3 25.5 20.0 18.7 21.8 23.0 27.
##  [97] 21.6 20.6 20.5 22.5 19.9 20.2 26.2 16.6 29.3 21.4 23.
```

Exemple de graphe : diagramme en boîte de l'IMC

```
boxplot(d$IMC)
```

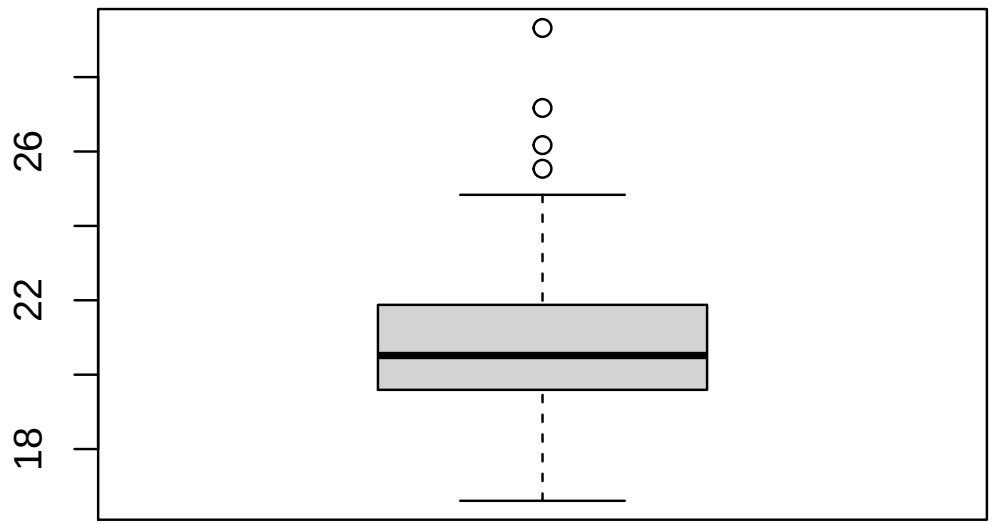


Diagramme des fréquences cumulées de l'IMC

```
plot(ecdf(d$IMC))
```

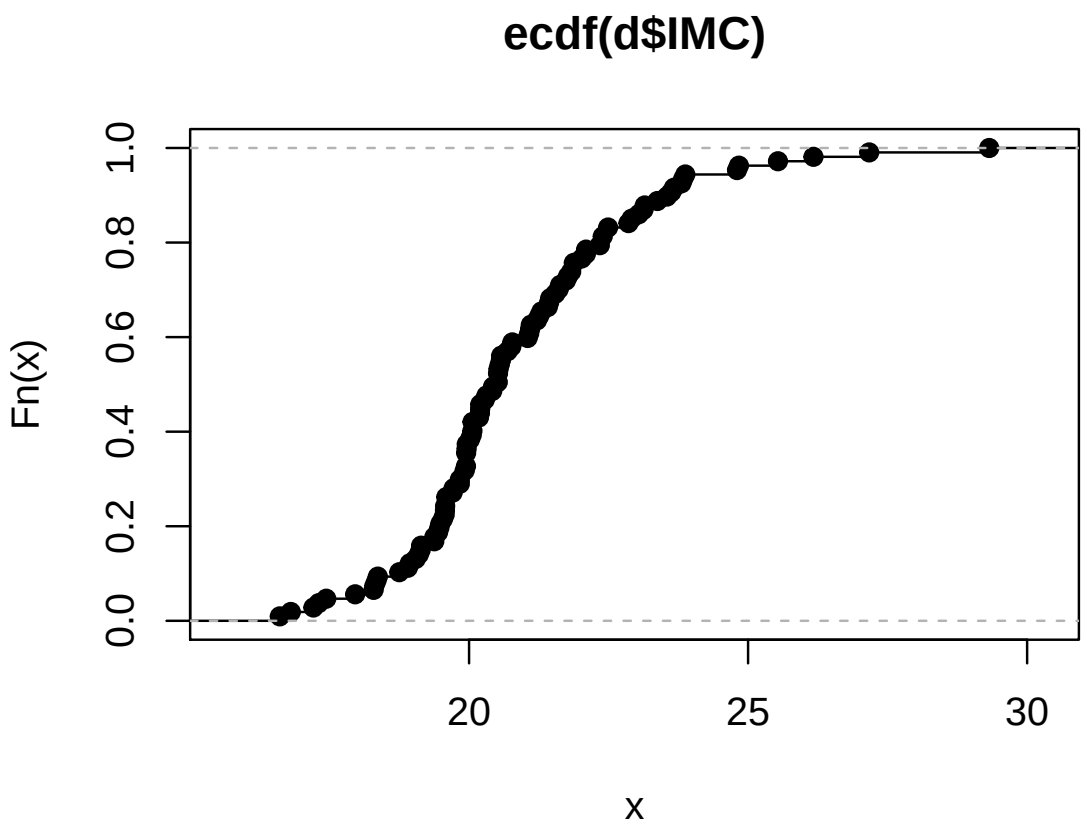
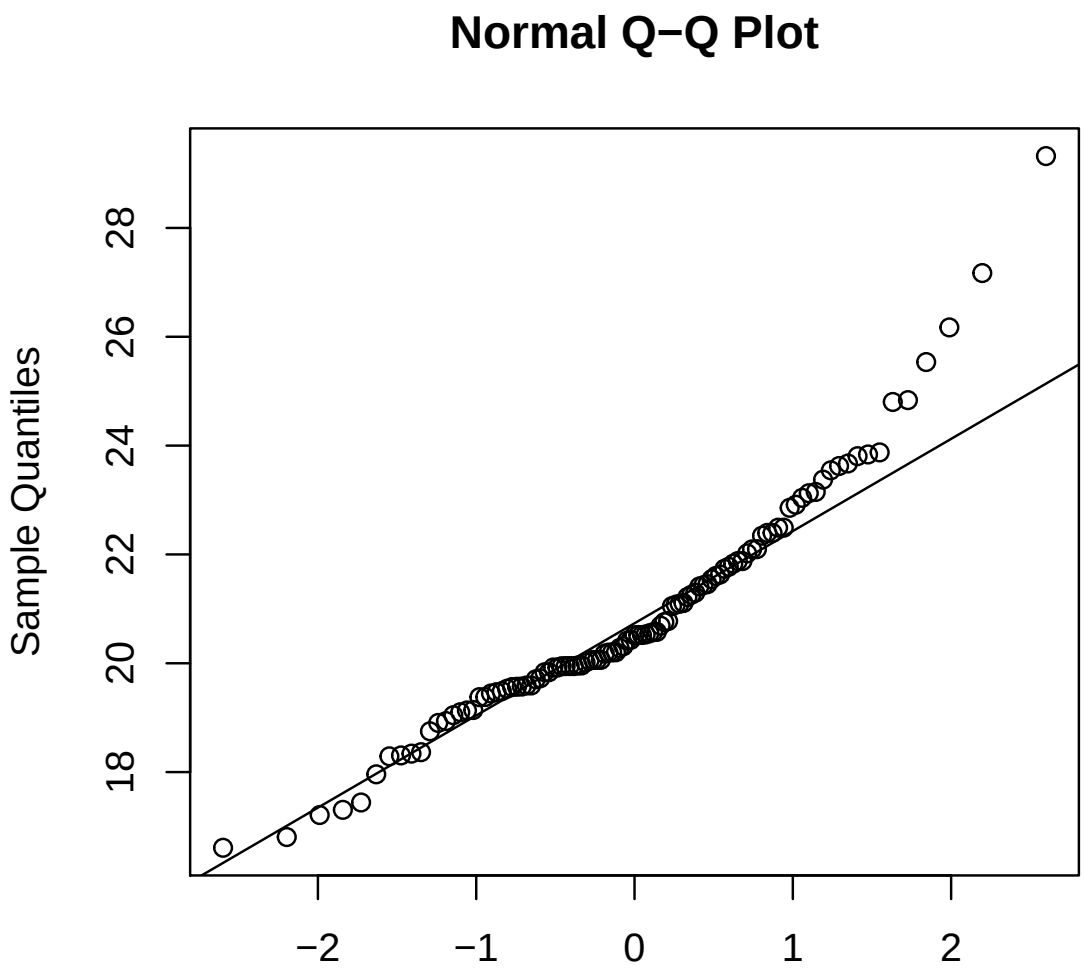


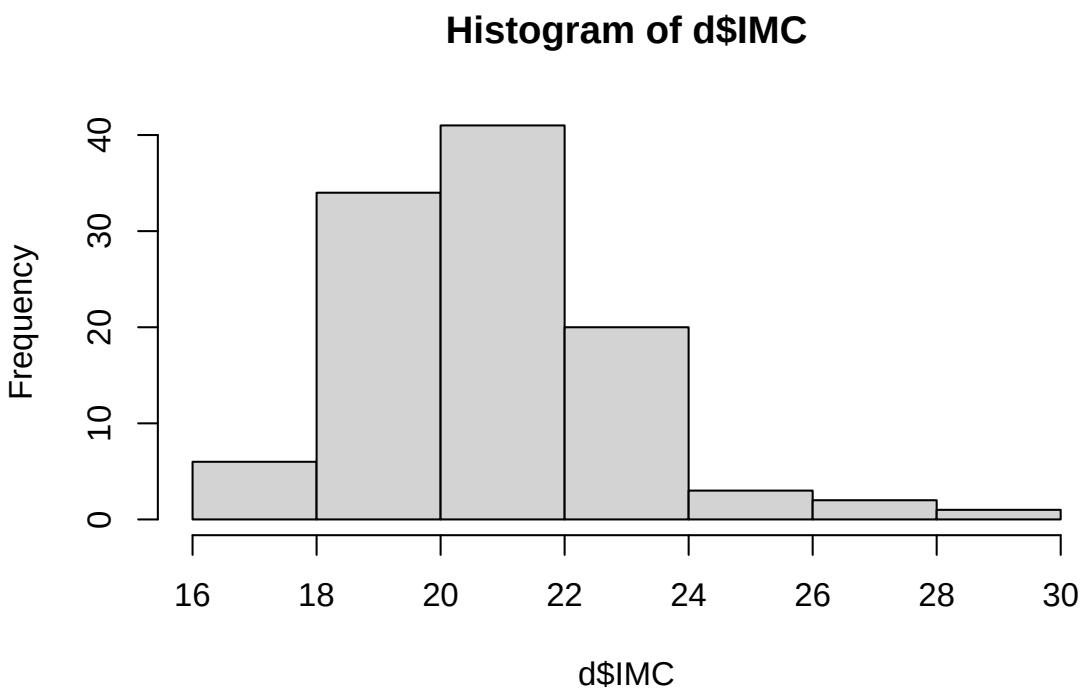
Diagramme Quantile-Quantile de l'IMC

```
qqnorm(d$IMC); qqline(d$IMC)
```



Histogramme de l'IMC

```
hist(d$IMC)
```



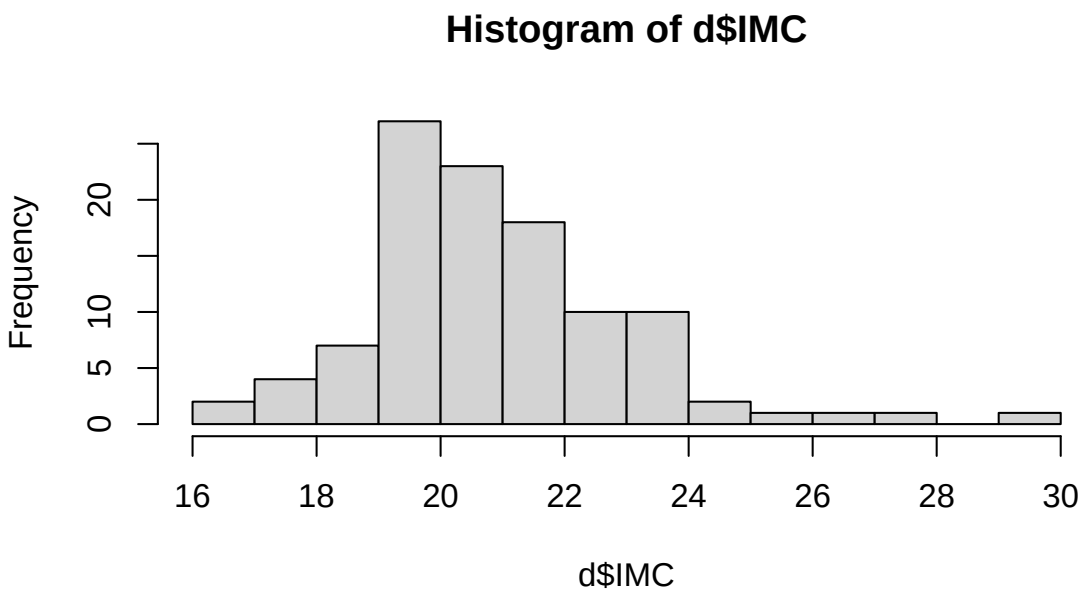
Accès à l'aide en ligne d'une fonction

Et si je veux changer certains arguments fixés par défaut de la fonction, par exemple changer la taille des classes d'un histogramme ?
Je consulte l'aide en ligne de cette fonction.
Essayez :

```
?hist
```

Histogramme de l'IMC avec définition explicite des classes

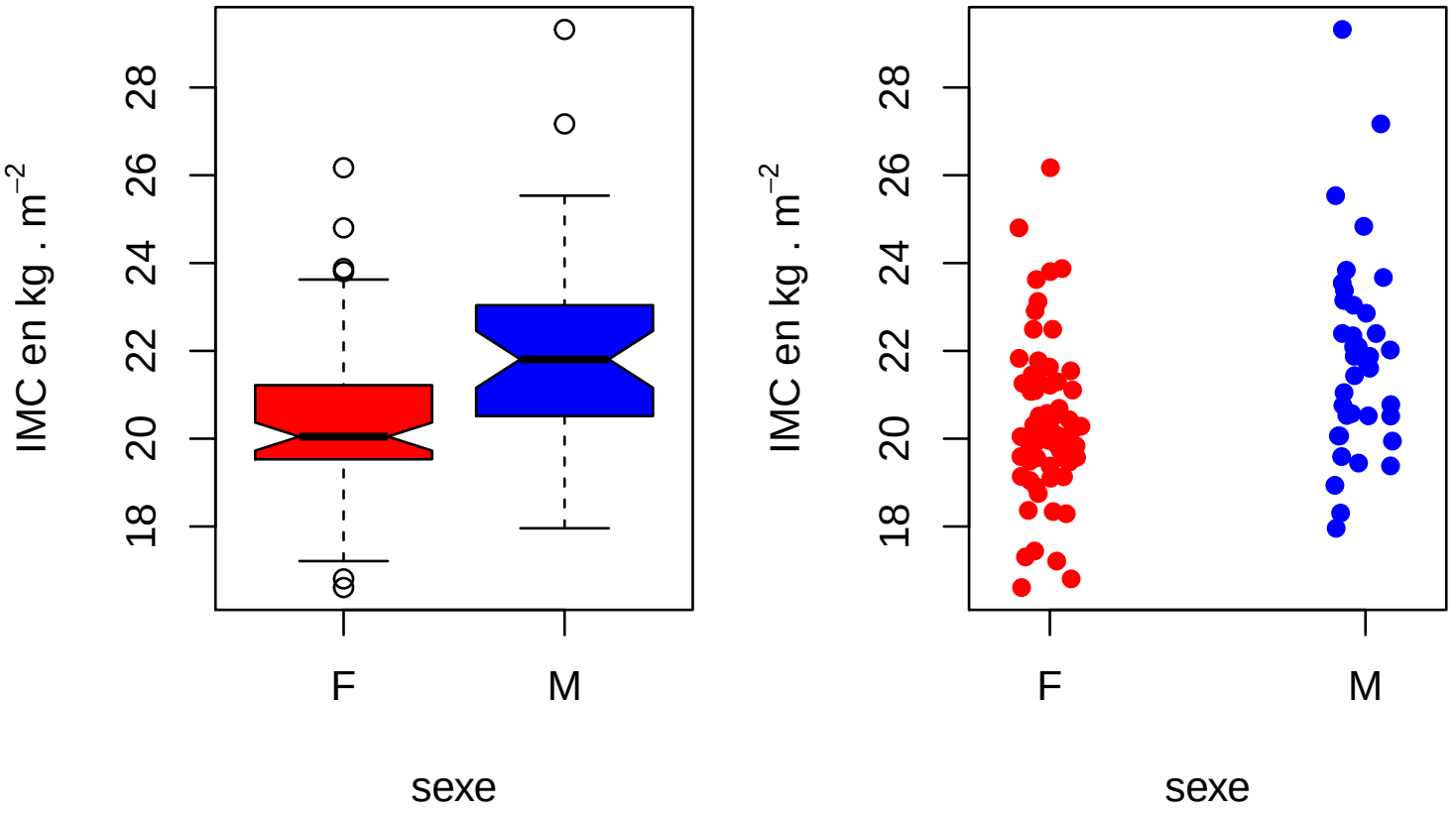
```
hist(d$IMC, breaks = c(16, 17, 18, 19, 20, 21, 22, 23,
                       24, 25, 26, 27, 28, 29, 30))
```




```
# Autres écritures possibles
hist(d$IMC, seq(16, 30, 1))
hist(d$IMC, 16:30)
```

Graphe personnalisé (code)

Il est bien sûr possible de réaliser des graphes plus complexes et de les personnaliser comme ci-dessous (cf. `http://www3.vetagro-sup.fr/ens/biostat/CMgraphesR.pdf`) :



A VOUS DE JOUER !

Toutes les informations générales concernant le langage **R** sont dans la partie introductive du guide (`http://www2.vetagro-sup.fr/ens/biostat/guideRM1.pdf`) :

- Spécification du répertoire courant
- Comment importer les données
- Manipulations et tranformations de données
  - Sélection de lignes dans un jeu de données
  - Création d'une variable qualitative à partir d'une quantitative
  - Changement de l'ordre ou des noms des modalités d'un facteur
  - Transformation d'une variable (ex. log)
  - Création d'une nouvelle variable au sein d'un jeu de données
- Gestion des graphes

A VOUS DE JOUER !

Les méthodes de base y sont ensuite présentées selon le plan :

- une série d'observations
  - variable quantitative
  - variable qualitative
- deux séries indépendantes d'observations
  - variable quantitative
  - variable qualitative
- deux séries dépendantes d'observations
  - variable quantitative
  - variable qualitative
- plusieurs séries indépendantes d'observations
  - variable quantitative
  - variable qualitative
- plusieurs séries dépendantes d'observations
  - variable quantitative
  - variable qualitative