

Initiation à R

Sébastien Déjean

Octobre 2020

Contents

Introduction : comment travailler avec R	2
Installation	2
RStudio	2
Environnement et espace de travail	2
Aide	2
Utilisation de packages	3
R Markdown	3
Structures de données	3
Type de données	3
Vecteur - <code>vector</code>	5
Facteur - <code>factor</code>	7
Matrice - <code>matrix</code>	8
Liste - <code>list</code>	8
Tableau de données - <code>data.frame</code>	9
Importation / exportation de données	14
Les fonctions de base	14
Des packages spécifiques	16
Avec RStudio	17
Manipulation de tableaux de données	17
Agir sur les lignes	17
Agir sur les colonnes	20
Fusionner des tableaux de données	29
Un peu de statistique	41
Statistique univariée	41
Statistique bivariée	42
Représentations graphiques	43
Pour des variables quantitatives	44
Etude de cas	48
Importation des données	48
Premier aperçu	49
Importation des données : le retour	51
Premier aperçu : le retour	51
Quelques analyses descriptives	54
Conclusion	56

Introduction : comment travailler avec R

Installation

R est un logiciel libre et gratuit. Il est téléchargeable à l'adresse cran.r-project.org pour les principaux systèmes d'exploitation Windows, Mac et Linux.

RStudio

Il est recommandé d'utiliser un environnement de développement qui simplifie le travail avec R. RStudio est un produit commercial et payant mais une version gratuite (légalement) est disponible au téléchargement à l'adresse rstudio.com.

Environnement et espace de travail

- Les objets manipulables dans R (variables créées, jeux de données importées, résultats de fonction...) peuvent être stockés dans un **espace de travail** (*workspace*) qui apparaîtra dans le **répertoire de travail** (*working directory*) comme un fichier `.RData`.
- On peut charger un **espace de travail** avec `load()`, lister son contenu avec `ls()` et le sauvegarder avec `save()` ou `save.image()`.
- On peut connaître le **répertoire de travail** courant grâce à la commande `getwd()` et le modifier avec `setwd()`.
- Le contenu du **répertoire de travail** (c'est à dire la liste des fichiers telle qu'elle peut apparaître dans un explorateur ou gestionnaire de fichiers) est accessible avec la commande `dir()`.

Ces opérations sont également réalisables au moyen des menus disponibles dans RStudio.

Aide

L'aide est accessible par un appel à la fonction `help()` ou en utilisant le raccourci `?`. Par exemple, l'aide de la fonction `plot()` est accessible par `help(plot)` ou `?plot`.

De manière générale, une fiche d'aide est composée des rubriques suivantes :

- Description
- Usage
- Arguments
- Details
- Value
- Note
- Authors
- Reference(s)
- See also
- Examples

Le nom de ces rubriques est suffisamment explicite pour désigner ce qu'elle contient. Pour un utilisateur non expérimenté, les deux dernières rubriques sont certainement les plus utiles : **See also** permet d'apprendre le nom de nouvelles fonctions en relation avec la recherche initiale et **Examples** permet de manipuler immédiatement, par copier-coller, toute nouvelle fonction.

La fonction `help.search()` (raccourcie en `??`) permet de rechercher dans l'aide en ligne les fonctions en rapport avec un terme (chaîne de caractères) passé en paramètres. Pour tout savoir sur les fonctions en relation avec le tracé de graphiques, on pourra par exemple saisir `help.search("plot")` ou `??plot`.

Et pour comprendre comment fonctionne `help.search()`, on pourra bien entendu entrer `help(help.search)` et même `help(help)` pour savoir comment utiliser la fonction `help()` !

Tout au long de ce document, vous êtes vivement encouragés à consulter la fiche d'aide de toutes les fonctions utilisées.

Les cheat sheets

Un aide précieuse est apportée par les **cheat sheet** que de plus en plus de développeurs de packages proposent à destination de leurs utilisateurs. Il s'agit de *feuilles de triche*, disons plutôt, *fiche de révision* ou *aide-mémoire* qui synthétisent des informations sur les principales fonctions d'un package. Le site de RStudio en propose quelques-unes à cette adresse rstudio.com/resources/cheatsheets.

Utilisation de packages

Les *packages* sont des ensembles de fonctions (et parfois de données) proposés par la communauté des utilisateurs de R et permettant d'accomplir des tâches spécifiques.

L'utilisation d'une fonction incluse dans un package nécessite 2 étapes préalables (réalisables via le menu **Packages** de Rstudio) :

- L'**installation** : utiliser la fonction `install.packages()` pour récupérer les sources du package et les installer. Cette opération ne doit être accomplie qu'une seule fois.
- Le **chargement** : utiliser la fonction `library(NomDuPackage)` pour charger le package et l'utiliser durant une session. Cette opération est à renouveler chaque fois que l'on souhaite utiliser un package.

Il est recommandé de procéder régulièrement à une mise à jour des packages (menu **Packages** ou fonction `update.packages()`).

R Markdown

Le système **R Markdown** permet de rassembler en un fichier unique l'ensemble des commandes R utilisées et les commentaires associés. Une compilation de ce fichier fournit une sortie au format pdf, html ou Word incluant également les sorties numériques et graphiques générées par les commandes. Son utilisation est relativement simple dans RStudio. Un exemple dont il est fortement recommandé de s'inspirer pour démarrer est disponible lorsque l'on demande la création d'un nouveau fichier R Markdown dans le menu **File** (**Fichier**) de RStudio. Un clic sur le bouton *Knit* assure la compilation du fichier et la production de la sortie au format souhaité.

Le présent document a été préparé avec R Markdown.

Structures de données

Type de données

Numérique, caractère, logique

R peut gérer des données numériques, des chaînes de caractères ainsi que des valeurs logiques TRUE / FALSE.

```
2+2
```

```
## [1] 4
```

```
2.5+1.56
```

```
## [1] 4.06
```

```
"hello"
```

```
## [1] "hello"
```

```
1 > 2
```

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

Cette caractéristique d'un objet R peut être testée avec des fonctions *ad hoc*.

```
is.numeric(2)
```

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

```
is.numeric(2.5)
```

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

```
is.numeric("hello")
```

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

```
is.character("hello")
```

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

```
is.logical(1>2)
```

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

La fonction `typeof` permet d'identifier le type d'un objet.

```
typeof(2)
```

```
## [1] "double"
```

```
typeof("bonjour")
```

```
## [1] "character"
```

```
typeof(TRUE)
```

```
## [1] "logical"
```

Données manquantes

Les données manquantes sont codées NA pour Not Available.

```
NA
```

```
## [1] NA
```

```
is.na(NA)
```

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

Il est important de les repérer et d'identifier comment une fonction réagit en cas de données manquantes.

Les dates

Les dates sont un cas particulier de données. On peut les manipuler dans R avec des outils spécifiques.

```
date()
```

```
## [1] "Thu Nov 26 17:17:27 2020"
```

```
armistice <- as.Date("1918-11-11")
```

```
weekdays(armistice)
```

```
## [1] "lundi"
```

```
months(armistice)
```

```
## [1] "novembre"
```

```
quarters(armistice)
```

```
## [1] "Q4"
```

```
aujourd'hui <- as.Date("2020-10-15")
```

```
aujourd'hui - armistice
```

```
## Time difference of 37229 days
```

L'opérateur <- permet de stocker le résultat d'une commande dans un objet.

Cette opération n'implique pas une visualisation du contenu de l'objet. Il faut pour cela saisir le nom de l'objet pour voir son contenu (voir l'exemple ci-dessous avec l'objet `UnVecteur`).

Vecteur - vector

Un vecteur est un objet qui contient un ou plusieurs éléments **du même type**.

Des valeurs numériques

```
UnVecteur <- c(2,5,9,8,12)
```

```
UnVecteur
```

```
## [1] 2 5 9 8 12
```

```
UnAutreVecteur <- 1:24
```

```
EncoreUn <- rep(5, times = 10)
```

```
EtUnDernier <- seq(from=1, to=60, by=7)
```

```
length(EtUnDernier)
```

```
## [1] 9
```

Une donnée manquante ne pose pas de problème dans un vecteur.

```
UnVecteurAvecNA <- c(2,5,NA,9,8,12,NA)
```

```
UnVecteurAvecNA
```

```
## [1] 2 5 NA 9 8 12 NA
```

Pour extraire tous les éléments différents de 9 dans un vecteur.

```
UnVecteur[UnVecteur != 9]
```

```
## [1] 2 5 8 12
```

Des chaînes de caractères

```
UnePhrase <- c("Bonjour", "les", "amis")
```

```
UnePhrase
```

```
## [1] "Bonjour" "les" "amis"
```

```
length(UnePhrase)
```

```
## [1] 3
```

La fonction `length` renvoie le nombre d'éléments du vecteur, pas le nombre de caractères d'une chaîne de caractères. Pour cela, on peut utiliser la fonction `nchar` qui, utilisée sur un vecteur, renvoie le nombre de caractères de chaque élément du vecteur (voir plus loin).

```
nchar(UnePhrase)
```

```
## [1] 7 3 4
```

Des valeurs logiques

```
VraiouFaux <- c(TRUE, TRUE, FALSE, TRUE, FALSE)
```

```
VraiouFaux
```

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

Mais pas de mélange

```
UnEssai <- c(2,3,"bonjour")
```

```
UnEssai
```

```
## [1] "2"      "3"      "bonjour"
```

La commande ci-dessus ne renvoie pas d'erreur mais les guillemets autour de 2 et 3 indiquent que R a converti tous les éléments du vecteur en chaîne de caractères. C'est en effet le format dans lequel R a su rassembler tous les éléments du vecteur. Convertir la valeur numérique 2 en la chaîne de caractères "2" est possible ; convertir la chaîne de caractères "bonjour" en numérique ne l'est pas.

Opérations sur un vecteur

On peut appliquer des opérations à un vecteur. Elles s'appliqueront pour tous les éléments.

```
EtUnDernier * 10
```

```
## [1] 10 80 150 220 290 360 430 500 570
```

```
log(EtUnDernier)
```

```
## [1] 0.000000 2.079442 2.708050 3.091042 3.367296 3.583519 3.761200 3.912023
```

```
## [9] 4.043051
```

```
UnVecteurAvecNA^2
```

```
## [1] 4 25 NA 81 64 144 NA
```

```
UnVecteur > 5
```

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

```
nchar(UnePhrase)
```

```
## [1] 7 3 4
```

Sélection de parties d'un vecteur

La sélection de parties d'un vecteur se fait en utilisant l'opérateur `[]`.

```
UnVecteur[1:3] # les 3 premiers éléments
```

```
## [1] 2 5 9
```

```
UnVecteur[-2] # tous les éléments sauf le 2ème
```

```
## [1] 2 9 8 12
```

```
UnVecteur[c(1,3)] # les 1er et 3ème éléments
```

```
## [1] 2 9
```

Facteur - factor

Les facteurs ressemblent beaucoup à des vecteurs de chaînes de caractères, mais il dispose de modalités pré-définies.

```
MesCategories <- c("Rien", "Peu", "Beaucoup", "Tout")
MesCategories
```

```
## [1] "Rien"      "Peu"      "Beaucoup" "Tout"
```

```
RecueilAvis <-
  sample(x = MesCategories, size=20, replace=TRUE)
RecueilAvis
```

```
## [1] "Tout"      "Peu"      "Beaucoup" "Peu"      "Tout"      "Rien"
## [7] "Tout"      "Beaucoup" "Beaucoup" "Tout"      "Peu"      "Peu"
## [13] "Tout"      "Beaucoup" "Tout"      "Beaucoup" "Tout"      "Beaucoup"
## [19] "Rien"      "Tout"
```

La ligne ci-dessus réalise un tirage aléatoire de taille 20 (`size=20`) avec remise (`replace=TRUE`) à partir du vecteur de chaînes de caractères de longueur 4 `MesCategories`.

```
RecueilAvisFactor <- factor(
  x = RecueilAvis,
  levels = MesCategories)
levels(RecueilAvisFactor)
```

```
## [1] "Rien"      "Peu"      "Beaucoup" "Tout"
```

La ligne ci-dessous renvoie une erreur car `Plein` ne fait pas partie des niveaux du facteur

```
RecueilAvisFactor[3] <- "Plein"
```

alors qu'une commande similaire sur un vecteur de chaîne de caractères ne pose aucun problème.

```
RecueilAvis[3] <- "Plein"
RecueilAvis
```

```
## [1] "Tout"      "Peu"      "Plein"     "Peu"      "Tout"      "Rien"
## [7] "Tout"      "Beaucoup" "Beaucoup" "Tout"      "Peu"      "Peu"
## [13] "Tout"      "Beaucoup" "Tout"      "Beaucoup" "Tout"      "Beaucoup"
## [19] "Rien"      "Tout"
```

A noter qu'une catégorie peut très bien ne pas apparaître dans un échantillonnage. Dans ce cas, le comptage des occurrences de chaque catégorie fera apparaître un 0 dans la catégorie correspondante.

```
EssaiFactor <- factor(c("Peu", "Rien", "Beaucoup", "Rien", "Peu"),
  levels = MesCategories)
table(EssaiFactor)
```

```
## EssaiFactor
##      Rien      Peu Beaucoup      Tout
##         2         2         1         0
```

```
table(c("Peu", "Rien", "Beaucoup", "Rien", "Peu"))
```

```
##
## Beaucoup      Peu      Rien
##           1         2         2
```

Matrice - matrix

Pour faire simple : une matrice est un vecteur à 2 dimensions. Comme le vecteur, elle n'autorise que des éléments du même type. La sélection de partie d'une matrice se fait avec 2 indices fournis entre crochets: [ligne,colonne].

```
UneMatrice <- matrix(1:24, ncol=3)
UneMatrice
```

```
##      [,1] [,2] [,3]
## [1,]   1   9  17
## [2,]   2  10  18
## [3,]   3  11  19
## [4,]   4  12  20
## [5,]   5  13  21
## [6,]   6  14  22
## [7,]   7  15  23
## [8,]   8  16  24
```

```
UneAutreMatrice <- UneMatrice
UneAutreMatrice[2,3] <- "mot"
UneAutreMatrice
```

```
##      [,1] [,2] [,3]
## [1,] "1"  "9"  "17"
## [2,] "2"  "10" "mot"
## [3,] "3"  "11" "19"
## [4,] "4"  "12" "20"
## [5,] "5"  "13" "21"
## [6,] "6"  "14" "22"
## [7,] "7"  "15" "23"
## [8,] "8"  "16" "24"
```

```
UneMatrice[5:7,1:2]
```

```
##      [,1] [,2]
## [1,]   5  13
## [2,]   6  14
## [3,]   7  15
```

Liste - list

A l'opposé du vecteur et de la matrice, un objet de type `list` permet de stocker des éléments de natures différentes.

```
MaListe <- list(Mot = "chien", Nombre = 8,
               VecteurNum = 1:6, Matrice = UneMatrice)
```

```
MaListe
```

```
## $Mot
## [1] "chien"
##
## $Nombre
## [1] 8
```

```
##
## $VecteurNum
## [1] 1 2 3 4 5 6
##
## $Matrice
##      [,1] [,2] [,3]
## [1,]  1   9  17
## [2,]  2  10  18
## [3,]  3  11  19
## [4,]  4  12  20
## [5,]  5  13  21
## [6,]  6  14  22
## [7,]  7  15  23
## [8,]  8  16  24

names(MaListe)

## [1] "Mot"          "Nombre"         "VecteurNum" "Matrice"

MaListe$Mot

## [1] "chien"

MaListe$VecteurNum

## [1] 1 2 3 4 5 6

MaListe$VecteurNum * 10

## [1] 10 20 30 40 50 60
```

L'opérateur \$ permet d'accéder aux différents composants d'un objet de type list.

La list est une structure de données très utile en R pour renvoyer les différents éléments du résultat d'une fonction. Consulter par exemple l'aide de la fonction `t.test` qui met en oeuvre le test de Student pour constater que la rubrique Value mentionne : A list with class "htest" containing the following components...

Tableau de données - data.frame

Le `data.frame`, comme son nom l'indique, est une structure pour stocker des données. Ce type hérite à la fois de la matrice (les données sont dans une structure rectangulaire caractérisée par des lignes et des colonnes) et de la liste (les colonnes d'un `data.frame` peuvent contenir des données de natures différentes).

```
taille <- round(runif(20,150,180),1)
masse <- round(runif(20,50,90),1)
coul.yeux <- c("bleu","noir","vert","marron")
coul.cheveux <- c("blond","chatain","noir")
yeux <- sample(coul.yeux,20,replace=T)
cheveux <- sample(coul.cheveux, 20, rep = TRUE)
table(yeux)
```

```
## yeux
##   bleu marron   noir   vert
##     3     2     6     9
```

```
table(cheveux)
```

```
## cheveux
##   blond chatain   noir
```

```
##      7      6      7
```

```
table(yeux, cheveux)
```

```
##      cheveux
## yeux  blond chatain noir
## bleu   1      1      1
## marron 1      0      1
## noir   3      2      1
## vert  2      3      4
```

```
MonDataFrame <- data.frame(taille,masse,yeux,cheveux)
```

```
MonDataFrame
```

```
##   taille masse  yeux cheveux
## 1  163.7  64.0 marron  blond
## 2  162.5  75.6  vert   noir
## 3  166.1  80.7  vert   noir
## 4  169.7  78.0  noir   blond
## 5  154.3  59.2  vert   noir
## 6  170.0  69.3  bleu   blond
## 7  169.4  83.2  noir   noir
## 8  168.7  58.0  noir  chatain
## 9  151.7  70.1  bleu  chatain
## 10 159.0  73.9  vert  chatain
## 11 178.8  80.6  vert  chatain
## 12 176.9  73.9  bleu  noir
## 13 175.9  82.3  vert  blond
## 14 154.1  78.6 marron  noir
## 15 161.4  59.8  vert  blond
## 16 171.5  89.4  vert  noir
## 17 175.9  68.3  noir  blond
## 18 176.5  53.2  noir  blond
## 19 160.5  58.9  vert  chatain
## 20 175.1  88.8  noir  chatain
```

```
MonDataFrame$taille
```

```
## [1] 163.7 162.5 166.1 169.7 154.3 170.0 169.4 168.7 151.7 159.0 178.8 176.9
## [13] 175.9 154.1 161.4 171.5 175.9 176.5 160.5 175.1
```

```
MonDataFrame$yeux
```

```
## [1] "marron" "vert"   "vert"   "noir"   "vert"   "bleu"   "noir"   "noir"
## [9] "bleu"   "vert"   "vert"   "bleu"   "vert"   "marron" "vert"   "vert"
## [17] "noir"   "noir"   "vert"   "noir"
```

```
IMC <- MonDataFrame$masse / MonDataFrame$taille^2
```

tibble

Les tibble sont une évolution récente de la structure `data.frame`, ils rendent la manipulation de jeux de données plus efficaces. Pour utiliser les tibbles et leurs fonctionnalités associées, il faut au préalable installer le package `install.packages("tibble")` ou la suite des packages `tidyverse` que nous utiliserons plus loin.

```
library(tibble)
```

```
iris.tib <- as_tibble(iris)
```

```
iris.tib
```

```
## # A tibble: 150 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##   <dbl>         <dbl>         <dbl>         <dbl> <fct>
## 1         5.1         3.5           1.4           0.2 setosa
## 2         4.9         3             1.4           0.2 setosa
## 3         4.7         3.2           1.3           0.2 setosa
## 4         4.6         3.1           1.5           0.2 setosa
## 5         5           3.6           1.4           0.2 setosa
## 6         5.4         3.9           1.7           0.4 setosa
## 7         4.6         3.4           1.4           0.3 setosa
## 8         5           3.4           1.5           0.2 setosa
## 9         4.4         2.9           1.4           0.2 setosa
## 10        4.9         3.1           1.5           0.1 setosa
## # ... with 140 more rows
```

```
iris
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1         5.1         3.5           1.4           0.2 setosa
## 2         4.9         3.0           1.4           0.2 setosa
## 3         4.7         3.2           1.3           0.2 setosa
## 4         4.6         3.1           1.5           0.2 setosa
## 5         5.0         3.6           1.4           0.2 setosa
## 6         5.4         3.9           1.7           0.4 setosa
## 7         4.6         3.4           1.4           0.3 setosa
## 8         5.0         3.4           1.5           0.2 setosa
## 9         4.4         2.9           1.4           0.2 setosa
## 10        4.9         3.1           1.5           0.1 setosa
## 11        5.4         3.7           1.5           0.2 setosa
## 12        4.8         3.4           1.6           0.2 setosa
## 13        4.8         3.0           1.4           0.1 setosa
## 14        4.3         3.0           1.1           0.1 setosa
## 15        5.8         4.0           1.2           0.2 setosa
## 16        5.7         4.4           1.5           0.4 setosa
## 17        5.4         3.9           1.3           0.4 setosa
## 18        5.1         3.5           1.4           0.3 setosa
## 19        5.7         3.8           1.7           0.3 setosa
## 20        5.1         3.8           1.5           0.3 setosa
## 21        5.4         3.4           1.7           0.2 setosa
## 22        5.1         3.7           1.5           0.4 setosa
## 23        4.6         3.6           1.0           0.2 setosa
## 24        5.1         3.3           1.7           0.5 setosa
## 25        4.8         3.4           1.9           0.2 setosa
## 26        5.0         3.0           1.6           0.2 setosa
## 27        5.0         3.4           1.6           0.4 setosa
## 28        5.2         3.5           1.5           0.2 setosa
## 29        5.2         3.4           1.4           0.2 setosa
## 30        4.7         3.2           1.6           0.2 setosa
## 31        4.8         3.1           1.6           0.2 setosa
## 32        5.4         3.4           1.5           0.4 setosa
## 33        5.2         4.1           1.5           0.1 setosa
## 34        5.5         4.2           1.4           0.2 setosa
## 35        4.9         3.1           1.5           0.2 setosa
## 36        5.0         3.2           1.2           0.2 setosa
## 37        5.5         3.5           1.3           0.2 setosa
```

## 38	4.9	3.6	1.4	0.1	setosa
## 39	4.4	3.0	1.3	0.2	setosa
## 40	5.1	3.4	1.5	0.2	setosa
## 41	5.0	3.5	1.3	0.3	setosa
## 42	4.5	2.3	1.3	0.3	setosa
## 43	4.4	3.2	1.3	0.2	setosa
## 44	5.0	3.5	1.6	0.6	setosa
## 45	5.1	3.8	1.9	0.4	setosa
## 46	4.8	3.0	1.4	0.3	setosa
## 47	5.1	3.8	1.6	0.2	setosa
## 48	4.6	3.2	1.4	0.2	setosa
## 49	5.3	3.7	1.5	0.2	setosa
## 50	5.0	3.3	1.4	0.2	setosa
## 51	7.0	3.2	4.7	1.4	versicolor
## 52	6.4	3.2	4.5	1.5	versicolor
## 53	6.9	3.1	4.9	1.5	versicolor
## 54	5.5	2.3	4.0	1.3	versicolor
## 55	6.5	2.8	4.6	1.5	versicolor
## 56	5.7	2.8	4.5	1.3	versicolor
## 57	6.3	3.3	4.7	1.6	versicolor
## 58	4.9	2.4	3.3	1.0	versicolor
## 59	6.6	2.9	4.6	1.3	versicolor
## 60	5.2	2.7	3.9	1.4	versicolor
## 61	5.0	2.0	3.5	1.0	versicolor
## 62	5.9	3.0	4.2	1.5	versicolor
## 63	6.0	2.2	4.0	1.0	versicolor
## 64	6.1	2.9	4.7	1.4	versicolor
## 65	5.6	2.9	3.6	1.3	versicolor
## 66	6.7	3.1	4.4	1.4	versicolor
## 67	5.6	3.0	4.5	1.5	versicolor
## 68	5.8	2.7	4.1	1.0	versicolor
## 69	6.2	2.2	4.5	1.5	versicolor
## 70	5.6	2.5	3.9	1.1	versicolor
## 71	5.9	3.2	4.8	1.8	versicolor
## 72	6.1	2.8	4.0	1.3	versicolor
## 73	6.3	2.5	4.9	1.5	versicolor
## 74	6.1	2.8	4.7	1.2	versicolor
## 75	6.4	2.9	4.3	1.3	versicolor
## 76	6.6	3.0	4.4	1.4	versicolor
## 77	6.8	2.8	4.8	1.4	versicolor
## 78	6.7	3.0	5.0	1.7	versicolor
## 79	6.0	2.9	4.5	1.5	versicolor
## 80	5.7	2.6	3.5	1.0	versicolor
## 81	5.5	2.4	3.8	1.1	versicolor
## 82	5.5	2.4	3.7	1.0	versicolor
## 83	5.8	2.7	3.9	1.2	versicolor
## 84	6.0	2.7	5.1	1.6	versicolor
## 85	5.4	3.0	4.5	1.5	versicolor
## 86	6.0	3.4	4.5	1.6	versicolor
## 87	6.7	3.1	4.7	1.5	versicolor
## 88	6.3	2.3	4.4	1.3	versicolor
## 89	5.6	3.0	4.1	1.3	versicolor
## 90	5.5	2.5	4.0	1.3	versicolor
## 91	5.5	2.6	4.4	1.2	versicolor

## 92	6.1	3.0	4.6	1.4	versicolor
## 93	5.8	2.6	4.0	1.2	versicolor
## 94	5.0	2.3	3.3	1.0	versicolor
## 95	5.6	2.7	4.2	1.3	versicolor
## 96	5.7	3.0	4.2	1.2	versicolor
## 97	5.7	2.9	4.2	1.3	versicolor
## 98	6.2	2.9	4.3	1.3	versicolor
## 99	5.1	2.5	3.0	1.1	versicolor
## 100	5.7	2.8	4.1	1.3	versicolor
## 101	6.3	3.3	6.0	2.5	virginica
## 102	5.8	2.7	5.1	1.9	virginica
## 103	7.1	3.0	5.9	2.1	virginica
## 104	6.3	2.9	5.6	1.8	virginica
## 105	6.5	3.0	5.8	2.2	virginica
## 106	7.6	3.0	6.6	2.1	virginica
## 107	4.9	2.5	4.5	1.7	virginica
## 108	7.3	2.9	6.3	1.8	virginica
## 109	6.7	2.5	5.8	1.8	virginica
## 110	7.2	3.6	6.1	2.5	virginica
## 111	6.5	3.2	5.1	2.0	virginica
## 112	6.4	2.7	5.3	1.9	virginica
## 113	6.8	3.0	5.5	2.1	virginica
## 114	5.7	2.5	5.0	2.0	virginica
## 115	5.8	2.8	5.1	2.4	virginica
## 116	6.4	3.2	5.3	2.3	virginica
## 117	6.5	3.0	5.5	1.8	virginica
## 118	7.7	3.8	6.7	2.2	virginica
## 119	7.7	2.6	6.9	2.3	virginica
## 120	6.0	2.2	5.0	1.5	virginica
## 121	6.9	3.2	5.7	2.3	virginica
## 122	5.6	2.8	4.9	2.0	virginica
## 123	7.7	2.8	6.7	2.0	virginica
## 124	6.3	2.7	4.9	1.8	virginica
## 125	6.7	3.3	5.7	2.1	virginica
## 126	7.2	3.2	6.0	1.8	virginica
## 127	6.2	2.8	4.8	1.8	virginica
## 128	6.1	3.0	4.9	1.8	virginica
## 129	6.4	2.8	5.6	2.1	virginica
## 130	7.2	3.0	5.8	1.6	virginica
## 131	7.4	2.8	6.1	1.9	virginica
## 132	7.9	3.8	6.4	2.0	virginica
## 133	6.4	2.8	5.6	2.2	virginica
## 134	6.3	2.8	5.1	1.5	virginica
## 135	6.1	2.6	5.6	1.4	virginica
## 136	7.7	3.0	6.1	2.3	virginica
## 137	6.3	3.4	5.6	2.4	virginica
## 138	6.4	3.1	5.5	1.8	virginica
## 139	6.0	3.0	4.8	1.8	virginica
## 140	6.9	3.1	5.4	2.1	virginica
## 141	6.7	3.1	5.6	2.4	virginica
## 142	6.9	3.1	5.1	2.3	virginica
## 143	5.8	2.7	5.1	1.9	virginica
## 144	6.8	3.2	5.9	2.3	virginica
## 145	6.7	3.3	5.7	2.5	virginica

```
## 146      6.7      3.0      5.2      2.3 virginica
## 147      6.3      2.5      5.0      1.9 virginica
## 148      6.5      3.0      5.2      2.0 virginica
## 149      6.2      3.4      5.4      2.3 virginica
## 150      5.9      3.0      5.1      1.8 virginica
```

Une des différences majeures avec les `data.frame` réside dans l’affichage qui est limité aux 10 premières lignes.

A noter également que l’affichage tient compte du nombre de colonnes à afficher.

Importation / exportation de données

Les fonctions de base

Exportation

Commençons par l’exportation car cela nous simplifie les exemples à suivre pour l’importation. La fonction de base pour exporter des données au format texte est `write.table`.

```
write.table(MonDataFrame, "Fichier_MonDataFrame.txt")
dir()
```

```
## [1] "Fichier_MonDataFrame_V2.csv"
## [2] "Fichier_MonDataFrame.csv"
## [3] "Fichier_MonDataFrame.txt"
## [4] "Formation.RData"
## [5] "fr-esr-parcours-des-bacheliers-en-paces.csv"
## [6] "intro_R.html"
## [7] "intro_R.pdf"
## [8] "intro_R.rmd"
## [9] "Intro_Shiny-figure"
## [10] "Intro_Shiny-rpubs.html"
## [11] "Intro_Shiny.html"
## [12] "Intro_Shiny.md"
## [13] "Intro_Shiny.Rpres"
## [14] "MonAppli1.R"
## [15] "MonAppli2.R"
## [16] "MonAppli3.R"
## [17] "MonAppli4.R"
## [18] "Survey_package_JSS.pdf"
## [19] "survey-wss-2010.pdf"
```

Sans précision d’un chemin, le fichier `.txt` ainsi créé apparaît dans le répertoire de travail. Ce que l’on peut vérifier au moyen de la commande `dir` ou dans le panneau Files de RStudio.

On peut ensuite jouer avec les paramètres pour modifier le séparateur (`sep`), la présence d’un entête (`col.names`) ou de noms de ligne (`row.names`)... ou utiliser les fonctions associées `write.csv` et `write.csv2` qui proposent des paramétrages par défaut différents.

```
write.csv(MonDataFrame, "Fichier_MonDataFrame.csv")
write.csv2(MonDataFrame, "Fichier_MonDataFrame_V2.csv")
```

Importation

Nous allons maintenant jouer la partition dans l’autre sens en important les données des fichiers précédemment créés.

Commençons par effacer le data.frame `MonDataFrame` et recréons le data.frame en important les fichiers.

```
rm(MonDataFrame)
```

```
MonDataFrameLeRetourTxt <- read.table("Fichier_MonDataFrame.txt")  
MonDataFrameLeRetourTxt
```

```
##      taille masse  yeux cheveux  
## 1    163.7  64.0 marron  blond  
## 2    162.5  75.6  vert   noir  
## 3    166.1  80.7  vert   noir  
## 4    169.7  78.0  noir   blond  
## 5    154.3  59.2  vert   noir  
## 6    170.0  69.3  bleu   blond  
## 7    169.4  83.2  noir   noir  
## 8    168.7  58.0  noir  chatain  
## 9    151.7  70.1  bleu  chatain  
## 10   159.0  73.9  vert  chatain  
## 11   178.8  80.6  vert  chatain  
## 12   176.9  73.9  bleu   noir  
## 13   175.9  82.3  vert   blond  
## 14   154.1  78.6  marron  noir  
## 15   161.4  59.8  vert   blond  
## 16   171.5  89.4  vert   noir  
## 17   175.9  68.3  noir   blond  
## 18   176.5  53.2  noir   blond  
## 19   160.5  58.9  vert  chatain  
## 20   175.1  88.8  noir  chatain
```

Tout semble s'être bien passé. Et les autres versions ?

```
MonDataFrameLeRetourCsv <- read.table("Fichier_MonDataFrame.csv")  
MonDataFrameLeRetourCsv
```

```
##      V1      V2  
## 1  NA , "taille", "masse", "yeux", "cheveux"  
## 2    1      ,163.7,64, "marron", "blond"  
## 3    2      ,162.5,75.6, "vert", "noir"  
## 4    3      ,166.1,80.7, "vert", "noir"  
## 5    4      ,169.7,78, "noir", "blond"  
## 6    5      ,154.3,59.2, "vert", "noir"  
## 7    6      ,170,69.3, "bleu", "blond"  
## 8    7      ,169.4,83.2, "noir", "noir"  
## 9    8      ,168.7,58, "noir", "chatain"  
## 10   9      ,151.7,70.1, "bleu", "chatain"  
## 11  10      ,159,73.9, "vert", "chatain"  
## 12  11      ,178.8,80.6, "vert", "chatain"  
## 13  12      ,176.9,73.9, "bleu", "noir"  
## 14  13      ,175.9,82.3, "vert", "blond"  
## 15  14      ,154.1,78.6, "marron", "noir"  
## 16  15      ,161.4,59.8, "vert", "blond"  
## 17  16      ,171.5,89.4, "vert", "noir"  
## 18  17      ,175.9,68.3, "noir", "blond"  
## 19  18      ,176.5,53.2, "noir", "blond"  
## 20  19      ,160.5,58.9, "vert", "chatain"  
## 21  20      ,175.1,88.8, "noir", "chatain"
```

Hum, hum, petit souci, non ?

```
MonDataFrameLeRetourCsvOK <- read.table("Fichier_MonDataFrame.csv", sep=";",  
                                         header = TRUE, row.names=1)
```

MonDataFrameLeRetourCsvOK

```
##   taille masse   yeux cheveux  
## 1  163.7  64.0 marron   blond  
## 2  162.5  75.6   vert    noir  
## 3  166.1  80.7   vert    noir  
## 4  169.7  78.0   noir    blond  
## 5  154.3  59.2   vert    noir  
## 6  170.0  69.3   bleu    blond  
## 7  169.4  83.2   noir    noir  
## 8  168.7  58.0   noir   chatain  
## 9  151.7  70.1   bleu   chatain  
## 10 159.0  73.9   vert   chatain  
## 11 178.8  80.6   vert   chatain  
## 12 176.9  73.9   bleu    noir  
## 13 175.9  82.3   vert    blond  
## 14 154.1  78.6 marron    noir  
## 15 161.4  59.8   vert    blond  
## 16 171.5  89.4   vert    noir  
## 17 175.9  68.3   noir    blond  
## 18 176.5  53.2   noir    blond  
## 19 160.5  58.9   vert   chatain  
## 20 175.1  88.8   noir   chatain
```

C'est mieux non ? Le même résultat est accessible avec

```
MonDataFrameLeRetourCsvOK2 <- read.csv("Fichier_MonDataFrame.csv")
```

Et de la même façon avec la fonction `read.csv2` pour importer le fichier avec le ; comme séparateur de champs et la , comme séparateur décimal.

Pour choisir la bonne fonction et son paramétrage, il est vivement recommandé de regarder le fichier à importer dans un éditeur de texte (Bloc-Notes, gedit, emacs...) et pas dans un tableur qui ne montrera pas le séparateur de champs.

Des packages spécifiques

readr

Le package `readr` propose une évolution des fonctions d'importation (et d'exportation) de fichiers au format texte. Il dispose d'une *cheat sheet* (data-import) présentant ses fonctions principales. Après installation et chargement du package, un coup d'oeil à l'aide en ligne permet de compléter la vue sur les possibilités de ce package.

readxl

Comme son nom l'indique, le package `readxl` permet d'importer des fichiers Excel aux formats xls etxlsx. Les fonctions `read_excel`, `read_xls` et `read_xlsx` permettent d'importer certaines feuilles d'un classeur Excel, certaines pages de données...

Installer et charger le package puis consulter l'aide en ligne des fonctions pour voir les spécificités proposées.

haven

Le package `haven` permet d'importer des fichiers données dans des formats propres à d'autres logiciels de statistique : SPSS, Stat et SAS. Cela peut être grandement utile lorsque l'on ne dispose plus de ces logiciels pour convertir les données dans un format texte.

Avec RStudio

L'importation des données peut aussi se faire en utilisant le menu *Import Dataset* de l'onglet Environment. Il s'agit d'une interface vers les packages cités précédemment. Le code équivalent à la manipulation de l'interface graphique est fourni à l'utilisateur.

Même si l'importation par le menu semble plus simple d'accès, il est à noter qu'une telle manipulation n'est pas reproductible, à l'inverse d'une ligne de commande que l'on pourra exécuter à nouveau (à condition de ne pas avoir déplacé le fichier à importer).

Manipulation de tableaux de données

Récupérons, pour commencer, un data frame au nom plus court.

```
MonDataFrame <- MonDataFrameLeRetourTxt
```

Agir sur les lignes

Ajouter des lignes

Pour ajouter des lignes à un tableau de données, on peut utiliser la fonction `rbind`.

```
MonDataFrame2 <- data.frame(taille = round(runif(5,150,180),1),
                             masse = round(runif(5,50,90),1),
                             yeux = sample(coul.yeux,5,replace=T),
                             cheveux = sample(coul.cheveux, 5, rep = TRUE))
MonDataFrameComplet <- rbind(MonDataFrame, MonDataFrame2)
dim(MonDataFrameComplet)
```

```
## [1] 25 4
```

```
MonDataFrameComplet
```

```
##      taille masse  yeux cheveux
## 1    163.7  64.0 marron  blond
## 2    162.5  75.6  vert   noir
## 3    166.1  80.7  vert   noir
## 4    169.7  78.0  noir   blond
## 5    154.3  59.2  vert   noir
## 6    170.0  69.3  bleu   blond
## 7    169.4  83.2  noir   noir
## 8    168.7  58.0  noir  chatain
## 9    151.7  70.1  bleu  chatain
## 10   159.0  73.9  vert  chatain
## 11   178.8  80.6  vert  chatain
## 12   176.9  73.9  bleu   noir
## 13   175.9  82.3  vert   blond
## 14   154.1  78.6 marron  noir
## 15   161.4  59.8  vert   blond
## 16   171.5  89.4  vert   noir
## 17   175.9  68.3  noir   blond
```

```
## 18  176.5  53.2  noir  blond
## 19  160.5  58.9  vert  chatain
## 20  175.1  88.8  noir  chatain
## 110 152.8  56.9  bleu  noir
## 21  158.7  61.8  bleu  blond
## 31  175.5  67.3  noir  chatain
## 41  163.0  87.5  noir  blond
## 51  174.4  74.6  bleu  blond
```

On a ainsi ajouter 5 lignes supplémentaires à l'objet `MonDataFrame` en concaténant `MonDataFrame` et `MonDataFrame2`. Cette opération n'est possible que parce que le nombre de colonnes est le même dans les 2 `data.frame` que l'on a collé en ligne.

```
rbind(MonDataFrame, MonDataFrame2[,1:3])
```

Cette commande renvoie un erreur car les 2 objets que l'on souhaite coller n'ont pas le même nombre de colonnes.

Supprimer des lignes

La suppression de ligne se fait en écrasant le contenu du `data.frame` courant ou en le stockant dans un autre `data.frame` (ce qui semble plus sûr), en procédant par exemple ainsi :

```
MonDataFrameSuppr <- MonDataFrame[-(1:5),]
MonDataFrameSuppr
```

```
##      taille masse  yeux cheveux
## 6    170.0  69.3  bleu  blond
## 7    169.4  83.2  noir   noir
## 8    168.7  58.0  noir  chatain
## 9    151.7  70.1  bleu  chatain
## 10   159.0  73.9  vert  chatain
## 11   178.8  80.6  vert  chatain
## 12   176.9  73.9  bleu  noir
## 13   175.9  82.3  vert  blond
## 14   154.1  78.6  marron noir
## 15   161.4  59.8  vert  blond
## 16   171.5  89.4  vert  noir
## 17   175.9  68.3  noir  blond
## 18   176.5  53.2  noir  blond
## 19   160.5  58.9  vert  chatain
## 20   175.1  88.8  noir  chatain
```

Les 5 premières lignes ont été supprimées et le `data.frame` résultant stocké dans le fichier `MonDataFrameSuppr`.

Pour sélectionner des lignes non consécutives, on peut utiliser la fonction `c`.

```
MonDataFrameSuppr <- MonDataFrame[-c(1,2,14,17),]
MonDataFrameSuppr
```

```
##      taille masse  yeux cheveux
## 3    166.1  80.7  vert   noir
## 4    169.7  78.0  noir   blond
## 5    154.3  59.2  vert   noir
## 6    170.0  69.3  bleu   blond
## 7    169.4  83.2  noir   noir
## 8    168.7  58.0  noir  chatain
## 9    151.7  70.1  bleu  chatain
```

```
## 10 159.0 73.9 vert chatain
## 11 178.8 80.6 vert chatain
## 12 176.9 73.9 bleu  noir
## 13 175.9 82.3 vert  blond
## 15 161.4 59.8 vert  blond
## 16 171.5 89.4 vert  noir
## 18 176.5 53.2 noir  blond
## 19 160.5 58.9 vert chatain
## 20 175.1 88.8 noir chatain
```

Le paragraphe qui suit permet également de filtrer les lignes d'un tableau de données selon différents critères.

Sélectionner des lignes

Pour sélectionner des lignes, la fonction `filter` du package `dplyr` propose des solutions relativement simples et intuitives. Pour être certain d'utiliser la fonction du package `dplyr`, et pas une fonction portant le même nom dans un autre package, on peut le mentionner explicitement avec l'opérateur `::`.

```
library(dplyr)
dplyr::filter(MonDataFrame, yeux=="noir")
```

```
##   taille masse yeux cheveux
## 4   169.7  78.0 noir  blond
## 7   169.4  83.2 noir  noir
## 8   168.7  58.0 noir chatain
## 17  175.9  68.3 noir  blond
## 18  176.5  53.2 noir  blond
## 20  175.1  88.8 noir chatain
```

```
filter(MonDataFrame, taille > 165)
```

```
##   taille masse yeux cheveux
## 3   166.1  80.7 vert  noir
## 4   169.7  78.0 noir  blond
## 6   170.0  69.3 bleu  blond
## 7   169.4  83.2 noir  noir
## 8   168.7  58.0 noir chatain
## 11  178.8  80.6 vert chatain
## 12  176.9  73.9 bleu  noir
## 13  175.9  82.3 vert  blond
## 16  171.5  89.4 vert  noir
## 17  175.9  68.3 noir  blond
## 18  176.5  53.2 noir  blond
## 20  175.1  88.8 noir chatain
```

```
filter(MonDataFrame, between(masse, 65, 70))
```

```
##   taille masse yeux cheveux
## 6   170.0  69.3 bleu  blond
## 17  175.9  68.3 noir  blond
```

```
filter(MonDataFrame, taille < 170 & cheveux=="blond")
```

```
##   taille masse  yeux cheveux
## 1   163.7  64.0 marron  blond
## 4   169.7  78.0  noir  blond
## 15  161.4  59.8  vert  blond
```

```
filter(MonDataFrame, taille < 170 & (cheveux=="blond" | cheveux=="chatain"))
```

```
##      taille masse  yeux cheveux
## 1    163.7  64.0 marron  blond
## 4    169.7  78.0  noir   blond
## 8    168.7  58.0  noir   chatain
## 9    151.7  70.1  bleu   chatain
## 10   159.0  73.9  vert   chatain
## 15   161.4  59.8  vert   blond
## 19   160.5  58.9  vert   chatain
```

```
dim(MonDataFrame)
```

```
## [1] 20  4
```

Noter que ces commandes ne modifient pas l'objet `MonDataFrame` et, si on souhaite conserver un extrait des données dans un autre objet, il faut le demander avec l'opérateur `<-`.

```
MonDataFrameExtraitTailleSup165 <- filter(MonDataFrame, taille > 165)
MonDataFrameExtraitTailleSup165
```

```
##      taille masse yeux cheveux
## 3    166.1  80.7  vert   noir
## 4    169.7  78.0  noir   blond
## 6    170.0  69.3  bleu   blond
## 7    169.4  83.2  noir   noir
## 8    168.7  58.0  noir   chatain
## 11   178.8  80.6  vert   chatain
## 12   176.9  73.9  bleu   noir
## 13   175.9  82.3  vert   blond
## 16   171.5  89.4  vert   noir
## 17   175.9  68.3  noir   blond
## 18   176.5  53.2  noir   blond
## 20   175.1  88.8  noir   chatain
```

Agir sur les colonnes

Les commandes vues ici sont similaires à celles vues précédemment pour la manipulation des lignes.

Ajouter des colonnes

De façon analogue à `rbind`, la fonction `cbind` permet de coller côte-à-côte deux tableaux de données ayant le même nombre de lignes.

```
ComplementMonDataFrame <- data.frame(
  Avis = sample(c("OK","NOK"), size=20, replace=TRUE),
  Note = sample(0:20, size=20, replace = TRUE))
MonDataFramePlusDeColonnes <- cbind(MonDataFrame, ComplementMonDataFrame)
MonDataFramePlusDeColonnes
```

```
##      taille masse  yeux cheveux Avis Note
## 1    163.7  64.0 marron  blond   OK    0
## 2    162.5  75.6  vert   noir   OK    20
## 3    166.1  80.7  vert   noir  NOK    1
## 4    169.7  78.0  noir   blond  NOK    17
## 5    154.3  59.2  vert   noir   OK    8
## 6    170.0  69.3  bleu   blond  NOK    9
```

```
## 7 169.4 83.2 noir noir NOK 15
## 8 168.7 58.0 noir chatain OK 13
## 9 151.7 70.1 bleu chatain OK 4
## 10 159.0 73.9 vert chatain NOK 2
## 11 178.8 80.6 vert chatain NOK 14
## 12 176.9 73.9 bleu noir OK 8
## 13 175.9 82.3 vert blond OK 0
## 14 154.1 78.6 marron noir OK 5
## 15 161.4 59.8 vert blond OK 20
## 16 171.5 89.4 vert noir NOK 13
## 17 175.9 68.3 noir blond OK 16
## 18 176.5 53.2 noir blond OK 8
## 19 160.5 58.9 vert chatain OK 17
## 20 175.1 88.8 noir chatain OK 12
```

Supprimer des colonnes

La suppression de colonnes peut là aussi se gérer de la même façon que la suppression de lignes.

```
MonDataFrameSupprCol <- MonDataFrame[,-c(1,3)]
MonDataFrameSupprCol
```

```
## masse cheveux
## 1 64.0 blond
## 2 75.6 noir
## 3 80.7 noir
## 4 78.0 blond
## 5 59.2 noir
## 6 69.3 blond
## 7 83.2 noir
## 8 58.0 chatain
## 9 70.1 chatain
## 10 73.9 chatain
## 11 80.6 chatain
## 12 73.9 noir
## 13 82.3 blond
## 14 78.6 noir
## 15 59.8 blond
## 16 89.4 noir
## 17 68.3 blond
## 18 53.2 blond
## 19 58.9 chatain
## 20 88.8 chatain
```

Sélectionner des colonnes

La fonction `select` du package `dplyr`

```
select(MonDataFrame, taille)
```

```
## taille
## 1 163.7
## 2 162.5
## 3 166.1
## 4 169.7
## 5 154.3
```

```
## 6 170.0
## 7 169.4
## 8 168.7
## 9 151.7
## 10 159.0
## 11 178.8
## 12 176.9
## 13 175.9
## 14 154.1
## 15 161.4
## 16 171.5
## 17 175.9
## 18 176.5
## 19 160.5
## 20 175.1
```

```
select(MonDataFrame, taille, masse)
```

```
##   taille masse
## 1  163.7  64.0
## 2  162.5  75.6
## 3  166.1  80.7
## 4  169.7  78.0
## 5  154.3  59.2
## 6  170.0  69.3
## 7  169.4  83.2
## 8  168.7  58.0
## 9  151.7  70.1
## 10 159.0  73.9
## 11 178.8  80.6
## 12 176.9  73.9
## 13 175.9  82.3
## 14 154.1  78.6
## 15 161.4  59.8
## 16 171.5  89.4
## 17 175.9  68.3
## 18 176.5  53.2
## 19 160.5  58.9
## 20 175.1  88.8
```

```
select(MonDataFrame, ends_with("eux"))
```

```
##   yeux cheveux
## 1 marron  blond
## 2  vert   noir
## 3  vert   noir
## 4  noir  blond
## 5  vert   noir
## 6  bleu  blond
## 7  noir  noir
## 8  noir chatain
## 9  bleu chatain
## 10 vert chatain
## 11 vert chatain
## 12 bleu  noir
```

```
## 13  vert  blond
## 14 marron  noir
## 15  vert  blond
## 16  vert  noir
## 17  noir  blond
## 18  noir  blond
## 19  vert chatain
## 20  noir chatain
```

Sur le jeu de données iris.

```
select(iris, starts_with("Petal"))
```

```
##      Petal.Length Petal.Width
## 1           1.4         0.2
## 2           1.4         0.2
## 3           1.3         0.2
## 4           1.5         0.2
## 5           1.4         0.2
## 6           1.7         0.4
## 7           1.4         0.3
## 8           1.5         0.2
## 9           1.4         0.2
## 10          1.5         0.1
## 11          1.5         0.2
## 12          1.6         0.2
## 13          1.4         0.1
## 14          1.1         0.1
## 15          1.2         0.2
## 16          1.5         0.4
## 17          1.3         0.4
## 18          1.4         0.3
## 19          1.7         0.3
## 20          1.5         0.3
## 21          1.7         0.2
## 22          1.5         0.4
## 23          1.0         0.2
## 24          1.7         0.5
## 25          1.9         0.2
## 26          1.6         0.2
## 27          1.6         0.4
## 28          1.5         0.2
## 29          1.4         0.2
## 30          1.6         0.2
## 31          1.6         0.2
## 32          1.5         0.4
## 33          1.5         0.1
## 34          1.4         0.2
## 35          1.5         0.2
## 36          1.2         0.2
## 37          1.3         0.2
## 38          1.4         0.1
## 39          1.3         0.2
## 40          1.5         0.2
## 41          1.3         0.3
```

## 42	1.3	0.3
## 43	1.3	0.2
## 44	1.6	0.6
## 45	1.9	0.4
## 46	1.4	0.3
## 47	1.6	0.2
## 48	1.4	0.2
## 49	1.5	0.2
## 50	1.4	0.2
## 51	4.7	1.4
## 52	4.5	1.5
## 53	4.9	1.5
## 54	4.0	1.3
## 55	4.6	1.5
## 56	4.5	1.3
## 57	4.7	1.6
## 58	3.3	1.0
## 59	4.6	1.3
## 60	3.9	1.4
## 61	3.5	1.0
## 62	4.2	1.5
## 63	4.0	1.0
## 64	4.7	1.4
## 65	3.6	1.3
## 66	4.4	1.4
## 67	4.5	1.5
## 68	4.1	1.0
## 69	4.5	1.5
## 70	3.9	1.1
## 71	4.8	1.8
## 72	4.0	1.3
## 73	4.9	1.5
## 74	4.7	1.2
## 75	4.3	1.3
## 76	4.4	1.4
## 77	4.8	1.4
## 78	5.0	1.7
## 79	4.5	1.5
## 80	3.5	1.0
## 81	3.8	1.1
## 82	3.7	1.0
## 83	3.9	1.2
## 84	5.1	1.6
## 85	4.5	1.5
## 86	4.5	1.6
## 87	4.7	1.5
## 88	4.4	1.3
## 89	4.1	1.3
## 90	4.0	1.3
## 91	4.4	1.2
## 92	4.6	1.4
## 93	4.0	1.2
## 94	3.3	1.0
## 95	4.2	1.3

## 96	4.2	1.2
## 97	4.2	1.3
## 98	4.3	1.3
## 99	3.0	1.1
## 100	4.1	1.3
## 101	6.0	2.5
## 102	5.1	1.9
## 103	5.9	2.1
## 104	5.6	1.8
## 105	5.8	2.2
## 106	6.6	2.1
## 107	4.5	1.7
## 108	6.3	1.8
## 109	5.8	1.8
## 110	6.1	2.5
## 111	5.1	2.0
## 112	5.3	1.9
## 113	5.5	2.1
## 114	5.0	2.0
## 115	5.1	2.4
## 116	5.3	2.3
## 117	5.5	1.8
## 118	6.7	2.2
## 119	6.9	2.3
## 120	5.0	1.5
## 121	5.7	2.3
## 122	4.9	2.0
## 123	6.7	2.0
## 124	4.9	1.8
## 125	5.7	2.1
## 126	6.0	1.8
## 127	4.8	1.8
## 128	4.9	1.8
## 129	5.6	2.1
## 130	5.8	1.6
## 131	6.1	1.9
## 132	6.4	2.0
## 133	5.6	2.2
## 134	5.1	1.5
## 135	5.6	1.4
## 136	6.1	2.3
## 137	5.6	2.4
## 138	5.5	1.8
## 139	4.8	1.8
## 140	5.4	2.1
## 141	5.6	2.4
## 142	5.1	2.3
## 143	5.1	1.9
## 144	5.9	2.3
## 145	5.7	2.5
## 146	5.2	2.3
## 147	5.0	1.9
## 148	5.2	2.0
## 149	5.4	2.3

```
## 150          5.1          1.8
```

```
select(iris.tib, starts_with("Petal"))
```

```
## # A tibble: 150 x 2
##   Petal.Length Petal.Width
##         <dbl>         <dbl>
## 1           1.4           0.2
## 2           1.4           0.2
## 3           1.3           0.2
## 4           1.5           0.2
## 5           1.4           0.2
## 6           1.7           0.4
## 7           1.4           0.3
## 8           1.5           0.2
## 9           1.4           0.2
## 10          1.5           0.1
## # ... with 140 more rows
```

```
select(iris, contains("Width"))
```

```
##   Sepal.Width Petal.Width
## 1           3.5           0.2
## 2           3.0           0.2
## 3           3.2           0.2
## 4           3.1           0.2
## 5           3.6           0.2
## 6           3.9           0.4
## 7           3.4           0.3
## 8           3.4           0.2
## 9           2.9           0.2
## 10          3.1           0.1
## 11          3.7           0.2
## 12          3.4           0.2
## 13          3.0           0.1
## 14          3.0           0.1
## 15          4.0           0.2
## 16          4.4           0.4
## 17          3.9           0.4
## 18          3.5           0.3
## 19          3.8           0.3
## 20          3.8           0.3
## 21          3.4           0.2
## 22          3.7           0.4
## 23          3.6           0.2
## 24          3.3           0.5
## 25          3.4           0.2
## 26          3.0           0.2
## 27          3.4           0.4
## 28          3.5           0.2
## 29          3.4           0.2
## 30          3.2           0.2
## 31          3.1           0.2
## 32          3.4           0.4
## 33          4.1           0.1
```

## 34	4.2	0.2
## 35	3.1	0.2
## 36	3.2	0.2
## 37	3.5	0.2
## 38	3.6	0.1
## 39	3.0	0.2
## 40	3.4	0.2
## 41	3.5	0.3
## 42	2.3	0.3
## 43	3.2	0.2
## 44	3.5	0.6
## 45	3.8	0.4
## 46	3.0	0.3
## 47	3.8	0.2
## 48	3.2	0.2
## 49	3.7	0.2
## 50	3.3	0.2
## 51	3.2	1.4
## 52	3.2	1.5
## 53	3.1	1.5
## 54	2.3	1.3
## 55	2.8	1.5
## 56	2.8	1.3
## 57	3.3	1.6
## 58	2.4	1.0
## 59	2.9	1.3
## 60	2.7	1.4
## 61	2.0	1.0
## 62	3.0	1.5
## 63	2.2	1.0
## 64	2.9	1.4
## 65	2.9	1.3
## 66	3.1	1.4
## 67	3.0	1.5
## 68	2.7	1.0
## 69	2.2	1.5
## 70	2.5	1.1
## 71	3.2	1.8
## 72	2.8	1.3
## 73	2.5	1.5
## 74	2.8	1.2
## 75	2.9	1.3
## 76	3.0	1.4
## 77	2.8	1.4
## 78	3.0	1.7
## 79	2.9	1.5
## 80	2.6	1.0
## 81	2.4	1.1
## 82	2.4	1.0
## 83	2.7	1.2
## 84	2.7	1.6
## 85	3.0	1.5
## 86	3.4	1.6
## 87	3.1	1.5

## 88	2.3	1.3
## 89	3.0	1.3
## 90	2.5	1.3
## 91	2.6	1.2
## 92	3.0	1.4
## 93	2.6	1.2
## 94	2.3	1.0
## 95	2.7	1.3
## 96	3.0	1.2
## 97	2.9	1.3
## 98	2.9	1.3
## 99	2.5	1.1
## 100	2.8	1.3
## 101	3.3	2.5
## 102	2.7	1.9
## 103	3.0	2.1
## 104	2.9	1.8
## 105	3.0	2.2
## 106	3.0	2.1
## 107	2.5	1.7
## 108	2.9	1.8
## 109	2.5	1.8
## 110	3.6	2.5
## 111	3.2	2.0
## 112	2.7	1.9
## 113	3.0	2.1
## 114	2.5	2.0
## 115	2.8	2.4
## 116	3.2	2.3
## 117	3.0	1.8
## 118	3.8	2.2
## 119	2.6	2.3
## 120	2.2	1.5
## 121	3.2	2.3
## 122	2.8	2.0
## 123	2.8	2.0
## 124	2.7	1.8
## 125	3.3	2.1
## 126	3.2	1.8
## 127	2.8	1.8
## 128	3.0	1.8
## 129	2.8	2.1
## 130	3.0	1.6
## 131	2.8	1.9
## 132	3.8	2.0
## 133	2.8	2.2
## 134	2.8	1.5
## 135	2.6	1.4
## 136	3.0	2.3
## 137	3.4	2.4
## 138	3.1	1.8
## 139	3.0	1.8
## 140	3.1	2.1
## 141	3.1	2.4

```
## 142      3.1      2.3
## 143      2.7      1.9
## 144      3.2      2.3
## 145      3.3      2.5
## 146      3.0      2.3
## 147      2.5      1.9
## 148      3.0      2.0
## 149      3.4      2.3
## 150      3.0      1.8
```

Pour stocker les résultats d'une telle sélection, il suffit d'assigner le résultat de la commande à un objet.

```
MonDataFrameCouleurs <- select(MonDataFrame, ends_with("eux"))
MonDataFrameCouleurs
```

```
##      yeux cheveux
## 1  marron  blond
## 2   vert   noir
## 3   vert   noir
## 4   noir  blond
## 5   vert   noir
## 6   bleu  blond
## 7   noir   noir
## 8   noir chatain
## 9   bleu chatain
## 10  vert chatain
## 11  vert chatain
## 12  bleu   noir
## 13  vert  blond
## 14 marron  noir
## 15  vert  blond
## 16  vert   noir
## 17  noir  blond
## 18  noir  blond
## 19  vert chatain
## 20  noir chatain
```

Fusionner des tableaux de données

Commençons par avoir deux tableaux de données avec un identifiant commun (`Prenom`) ; les 2 tableaux étant dans des ordres différents.

```
Prenom <- c("Emma", "Gabriel", "Louise", "Raphael", "Alice", "Leo", "Chloe", "Louis",
           "Lina", "Lucas", "Rose", "Adam", "Lea", "Arthur", "Anna", "Hugo", "Mila",
           "Jules", "Mia", "Mael")
MonDataFramePrenom <- cbind(Prenom, MonDataFrame)
UnAutreDataFramePrenom <- data.frame(Prenom = sort(Prenom), Note = 1:20)
MonDataFramePrenom
```

```
##      Prenom taille masse  yeux cheveux
## 1   Emma   163.7   64.0 marron  blond
## 2 Gabriel   162.5   75.6  vert   noir
## 3  Louise   166.1   80.7  vert   noir
## 4 Raphael   169.7   78.0  noir   blond
## 5   Alice   154.3   59.2  vert   noir
## 6    Leo   170.0   69.3  bleu   blond
```

```
## 7   Chloe 169.4 83.2  noir   noir
## 8   Louis 168.7 58.0  noir  chatain
## 9   Lina  151.7 70.1  bleu  chatain
## 10  Lucas 159.0 73.9  vert  chatain
## 11  Rose  178.8 80.6  vert  chatain
## 12  Adam  176.9 73.9  bleu   noir
## 13  Lea   175.9 82.3  vert   blond
## 14  Arthur 154.1 78.6  marron noir
## 15  Anna  161.4 59.8  vert   blond
## 16  Hugo  171.5 89.4  vert   noir
## 17  Mila  175.9 68.3  noir   blond
## 18  Jules 176.5 53.2  noir   blond
## 19  Mia   160.5 58.9  vert  chatain
## 20  Mael  175.1 88.8  noir  chatain
```

```
UnAutreDataFramePrenom
```

```
##      Prenom Note
## 1      Adam     1
## 2      Alice     2
## 3      Anna     3
## 4      Arthur     4
## 5      Chloe     5
## 6      Emma     6
## 7      Gabriel    7
## 8      Hugo     8
## 9      Jules     9
## 10     Lea     10
## 11     Leo     11
## 12     Lina     12
## 13     Louis    13
## 14     Louise   14
## 15     Lucas    15
## 16     Mael     16
## 17     Mia     17
## 18     Mila    18
## 19     Raphael  19
## 20     Rose     20
```

La fusion des 2 tableaux de données peut se réaliser au moyen de la fonction `merge`.

```
merge(MonDataFramePrenom, UnAutreDataFramePrenom)
```

```
##      Prenom taille masse  yeux cheveux Note
## 1      Adam 176.9 73.9  bleu   noir     1
## 2      Alice 154.3 59.2  vert   noir     2
## 3      Anna 161.4 59.8  vert   blond    3
## 4      Arthur 154.1 78.6  marron noir     4
## 5      Chloe 169.4 83.2  noir   noir     5
## 6      Emma 163.7 64.0  marron blond    6
## 7      Gabriel 162.5 75.6  vert   noir     7
## 8      Hugo 171.5 89.4  vert   noir     8
## 9      Jules 176.5 53.2  noir   blond    9
## 10     Lea 175.9 82.3  vert   blond   10
## 11     Leo 170.0 69.3  bleu   blond   11
## 12     Lina 151.7 70.1  bleu  chatain 12
```

```
## 13 Louis 168.7 58.0 noir chatain 13
## 14 Louise 166.1 80.7 vert noir 14
## 15 Lucas 159.0 73.9 vert chatain 15
## 16 Mael 175.1 88.8 noir chatain 16
## 17 Mia 160.5 58.9 vert chatain 17
## 18 Mila 175.9 68.3 noir blond 18
## 19 Raphael 169.7 78.0 noir blond 19
## 20 Rose 178.8 80.6 vert chatain 20
```

```
merge(UnAutreDataFramePrenom, MonDataFramePrenom)
```

```
##      Prenom Note taille masse  yeux cheveux
## 1      Adam   1  176.9  73.9  bleu   noir
## 2      Alice   2  154.3  59.2  vert   noir
## 3      Anna   3  161.4  59.8  vert   blond
## 4      Arthur  4  154.1  78.6 marron  noir
## 5      Chloe  5  169.4  83.2  noir   noir
## 6      Emma   6  163.7  64.0 marron  blond
## 7      Gabriel 7  162.5  75.6  vert   noir
## 8      Hugo   8  171.5  89.4  vert   noir
## 9      Jules  9  176.5  53.2  noir   blond
## 10     Lea   10  175.9  82.3  vert   blond
## 11     Leo   11  170.0  69.3  bleu   blond
## 12     Lina  12  151.7  70.1  bleu  chatain
## 13     Louis 13  168.7  58.0  noir  chatain
## 14     Louise 14  166.1  80.7  vert   noir
## 15     Lucas 15  159.0  73.9  vert  chatain
## 16     Mael  16  175.1  88.8  noir  chatain
## 17     Mia   17  160.5  58.9  vert  chatain
## 18     Mila  18  175.9  68.3  noir   blond
## 19     Raphael 19  169.7  78.0  noir   blond
## 20     Rose  20  178.8  80.6  vert  chatain
```

By default the data frames are merged on the columns with names they both have.

- Si les colonnes ne portent pas le même nom, il faut le préciser, sinon cela va poser problème...

```
UnAutreDataFramePrenom <- data.frame(FirstName = sort(Prenom), Note = 1:20)
UnAutreDataFramePrenom
```

```
##      FirstName Note
## 1         Adam   1
## 2         Alice   2
## 3         Anna   3
## 4        Arthur   4
## 5         Chloe   5
## 6         Emma   6
## 7        Gabriel   7
## 8         Hugo   8
## 9         Jules   9
## 10        Lea   10
## 11         Leo   11
## 12        Lina   12
## 13        Louis  13
## 14       Louise  14
## 15        Lucas  15
## 16        Mael  16
```

```
## 17      Mia    17
## 18      Mila   18
## 19  Raphael  19
## 20      Rose   20
```

```
merge(MonDataFramePrenom, UnAutreDataFramePrenom)
```

```
##      Prenom taille masse  yeux cheveux FirstName Note
## 1      Emma  163.7  64.0 marron  blond      Adam    1
## 2  Gabriel  162.5  75.6  vert   noir      Adam    1
## 3    Louise  166.1  80.7  vert   noir      Adam    1
## 4  Raphael  169.7  78.0  noir   blond     Adam    1
## 5      Alice  154.3  59.2  vert   noir      Adam    1
## 6      Leo   170.0  69.3  bleu   blond     Adam    1
## 7    Chloe  169.4  83.2  noir   noir      Adam    1
## 8    Louis  168.7  58.0  noir  chatain   Adam    1
## 9      Lina  151.7  70.1  bleu  chatain   Adam    1
## 10   Lucas  159.0  73.9  vert  chatain   Adam    1
## 11   Rose   178.8  80.6  vert  chatain   Adam    1
## 12   Adam   176.9  73.9  bleu   noir      Adam    1
## 13   Lea    175.9  82.3  vert   blond     Adam    1
## 14  Arthur  154.1  78.6 marron  noir      Adam    1
## 15   Anna   161.4  59.8  vert   blond     Adam    1
## 16   Hugo   171.5  89.4  vert   noir      Adam    1
## 17   Mila   175.9  68.3  noir   blond     Adam    1
## 18   Jules  176.5  53.2  noir   blond     Adam    1
## 19   Mia    160.5  58.9  vert  chatain   Adam    1
## 20   Mael   175.1  88.8  noir  chatain   Adam    1
## 21   Emma   163.7  64.0 marron  blond     Alice    2
## 22  Gabriel  162.5  75.6  vert   noir      Alice    2
## 23   Louise  166.1  80.7  vert   noir      Alice    2
## 24  Raphael  169.7  78.0  noir   blond     Alice    2
## 25   Alice  154.3  59.2  vert   noir      Alice    2
## 26   Leo    170.0  69.3  bleu   blond     Alice    2
## 27   Chloe  169.4  83.2  noir   noir      Alice    2
## 28   Louis  168.7  58.0  noir  chatain   Alice    2
## 29   Lina   151.7  70.1  bleu  chatain   Alice    2
## 30   Lucas  159.0  73.9  vert  chatain   Alice    2
## 31   Rose   178.8  80.6  vert  chatain   Alice    2
## 32   Adam   176.9  73.9  bleu   noir      Alice    2
## 33   Lea    175.9  82.3  vert   blond     Alice    2
## 34  Arthur  154.1  78.6 marron  noir      Alice    2
## 35   Anna   161.4  59.8  vert   blond     Alice    2
## 36   Hugo   171.5  89.4  vert   noir      Alice    2
## 37   Mila   175.9  68.3  noir   blond     Alice    2
## 38   Jules  176.5  53.2  noir   blond     Alice    2
## 39   Mia    160.5  58.9  vert  chatain   Alice    2
## 40   Mael   175.1  88.8  noir  chatain   Alice    2
## 41   Emma   163.7  64.0 marron  blond     Anna    3
## 42  Gabriel  162.5  75.6  vert   noir      Anna    3
## 43   Louise  166.1  80.7  vert   noir      Anna    3
## 44  Raphael  169.7  78.0  noir   blond     Anna    3
## 45   Alice  154.3  59.2  vert   noir      Anna    3
## 46   Leo    170.0  69.3  bleu   blond     Anna    3
## 47   Chloe  169.4  83.2  noir   noir      Anna    3
```

## 48	Louis	168.7	58.0	noir chatain	Anna	3
## 49	Lina	151.7	70.1	bleu chatain	Anna	3
## 50	Lucas	159.0	73.9	vert chatain	Anna	3
## 51	Rose	178.8	80.6	vert chatain	Anna	3
## 52	Adam	176.9	73.9	bleu noir	Anna	3
## 53	Lea	175.9	82.3	vert blond	Anna	3
## 54	Arthur	154.1	78.6	marron noir	Anna	3
## 55	Anna	161.4	59.8	vert blond	Anna	3
## 56	Hugo	171.5	89.4	vert noir	Anna	3
## 57	Mila	175.9	68.3	noir blond	Anna	3
## 58	Jules	176.5	53.2	noir blond	Anna	3
## 59	Mia	160.5	58.9	vert chatain	Anna	3
## 60	Mael	175.1	88.8	noir chatain	Anna	3
## 61	Emmma	163.7	64.0	marron blond	Arthur	4
## 62	Gabriel	162.5	75.6	vert noir	Arthur	4
## 63	Louise	166.1	80.7	vert noir	Arthur	4
## 64	Raphael	169.7	78.0	noir blond	Arthur	4
## 65	Alice	154.3	59.2	vert noir	Arthur	4
## 66	Leo	170.0	69.3	bleu blond	Arthur	4
## 67	Chloe	169.4	83.2	noir noir	Arthur	4
## 68	Louis	168.7	58.0	noir chatain	Arthur	4
## 69	Lina	151.7	70.1	bleu chatain	Arthur	4
## 70	Lucas	159.0	73.9	vert chatain	Arthur	4
## 71	Rose	178.8	80.6	vert chatain	Arthur	4
## 72	Adam	176.9	73.9	bleu noir	Arthur	4
## 73	Lea	175.9	82.3	vert blond	Arthur	4
## 74	Arthur	154.1	78.6	marron noir	Arthur	4
## 75	Anna	161.4	59.8	vert blond	Arthur	4
## 76	Hugo	171.5	89.4	vert noir	Arthur	4
## 77	Mila	175.9	68.3	noir blond	Arthur	4
## 78	Jules	176.5	53.2	noir blond	Arthur	4
## 79	Mia	160.5	58.9	vert chatain	Arthur	4
## 80	Mael	175.1	88.8	noir chatain	Arthur	4
## 81	Emmma	163.7	64.0	marron blond	Chloe	5
## 82	Gabriel	162.5	75.6	vert noir	Chloe	5
## 83	Louise	166.1	80.7	vert noir	Chloe	5
## 84	Raphael	169.7	78.0	noir blond	Chloe	5
## 85	Alice	154.3	59.2	vert noir	Chloe	5
## 86	Leo	170.0	69.3	bleu blond	Chloe	5
## 87	Chloe	169.4	83.2	noir noir	Chloe	5
## 88	Louis	168.7	58.0	noir chatain	Chloe	5
## 89	Lina	151.7	70.1	bleu chatain	Chloe	5
## 90	Lucas	159.0	73.9	vert chatain	Chloe	5
## 91	Rose	178.8	80.6	vert chatain	Chloe	5
## 92	Adam	176.9	73.9	bleu noir	Chloe	5
## 93	Lea	175.9	82.3	vert blond	Chloe	5
## 94	Arthur	154.1	78.6	marron noir	Chloe	5
## 95	Anna	161.4	59.8	vert blond	Chloe	5
## 96	Hugo	171.5	89.4	vert noir	Chloe	5
## 97	Mila	175.9	68.3	noir blond	Chloe	5
## 98	Jules	176.5	53.2	noir blond	Chloe	5
## 99	Mia	160.5	58.9	vert chatain	Chloe	5
## 100	Mael	175.1	88.8	noir chatain	Chloe	5
## 101	Emmma	163.7	64.0	marron blond	Emmma	6

## 102	Gabriel	162.5	75.6	vert	noir	Emmma	6
## 103	Louise	166.1	80.7	vert	noir	Emmma	6
## 104	Raphael	169.7	78.0	noir	blond	Emmma	6
## 105	Alice	154.3	59.2	vert	noir	Emmma	6
## 106	Leo	170.0	69.3	bleu	blond	Emmma	6
## 107	Chloe	169.4	83.2	noir	noir	Emmma	6
## 108	Louis	168.7	58.0	noir	chatain	Emmma	6
## 109	Lina	151.7	70.1	bleu	chatain	Emmma	6
## 110	Lucas	159.0	73.9	vert	chatain	Emmma	6
## 111	Rose	178.8	80.6	vert	chatain	Emmma	6
## 112	Adam	176.9	73.9	bleu	noir	Emmma	6
## 113	Lea	175.9	82.3	vert	blond	Emmma	6
## 114	Arthur	154.1	78.6	marron	noir	Emmma	6
## 115	Anna	161.4	59.8	vert	blond	Emmma	6
## 116	Hugo	171.5	89.4	vert	noir	Emmma	6
## 117	Mila	175.9	68.3	noir	blond	Emmma	6
## 118	Jules	176.5	53.2	noir	blond	Emmma	6
## 119	Mia	160.5	58.9	vert	chatain	Emmma	6
## 120	Mael	175.1	88.8	noir	chatain	Emmma	6
## 121	Emmma	163.7	64.0	marron	blond	Gabriel	7
## 122	Gabriel	162.5	75.6	vert	noir	Gabriel	7
## 123	Louise	166.1	80.7	vert	noir	Gabriel	7
## 124	Raphael	169.7	78.0	noir	blond	Gabriel	7
## 125	Alice	154.3	59.2	vert	noir	Gabriel	7
## 126	Leo	170.0	69.3	bleu	blond	Gabriel	7
## 127	Chloe	169.4	83.2	noir	noir	Gabriel	7
## 128	Louis	168.7	58.0	noir	chatain	Gabriel	7
## 129	Lina	151.7	70.1	bleu	chatain	Gabriel	7
## 130	Lucas	159.0	73.9	vert	chatain	Gabriel	7
## 131	Rose	178.8	80.6	vert	chatain	Gabriel	7
## 132	Adam	176.9	73.9	bleu	noir	Gabriel	7
## 133	Lea	175.9	82.3	vert	blond	Gabriel	7
## 134	Arthur	154.1	78.6	marron	noir	Gabriel	7
## 135	Anna	161.4	59.8	vert	blond	Gabriel	7
## 136	Hugo	171.5	89.4	vert	noir	Gabriel	7
## 137	Mila	175.9	68.3	noir	blond	Gabriel	7
## 138	Jules	176.5	53.2	noir	blond	Gabriel	7
## 139	Mia	160.5	58.9	vert	chatain	Gabriel	7
## 140	Mael	175.1	88.8	noir	chatain	Gabriel	7
## 141	Emmma	163.7	64.0	marron	blond	Hugo	8
## 142	Gabriel	162.5	75.6	vert	noir	Hugo	8
## 143	Louise	166.1	80.7	vert	noir	Hugo	8
## 144	Raphael	169.7	78.0	noir	blond	Hugo	8
## 145	Alice	154.3	59.2	vert	noir	Hugo	8
## 146	Leo	170.0	69.3	bleu	blond	Hugo	8
## 147	Chloe	169.4	83.2	noir	noir	Hugo	8
## 148	Louis	168.7	58.0	noir	chatain	Hugo	8
## 149	Lina	151.7	70.1	bleu	chatain	Hugo	8
## 150	Lucas	159.0	73.9	vert	chatain	Hugo	8
## 151	Rose	178.8	80.6	vert	chatain	Hugo	8
## 152	Adam	176.9	73.9	bleu	noir	Hugo	8
## 153	Lea	175.9	82.3	vert	blond	Hugo	8
## 154	Arthur	154.1	78.6	marron	noir	Hugo	8
## 155	Anna	161.4	59.8	vert	blond	Hugo	8

## 156	Hugo	171.5	89.4	vert	noir	Hugo	8
## 157	Mila	175.9	68.3	noir	blond	Hugo	8
## 158	Jules	176.5	53.2	noir	blond	Hugo	8
## 159	Mia	160.5	58.9	vert	chatain	Hugo	8
## 160	Mael	175.1	88.8	noir	chatain	Hugo	8
## 161	Emmma	163.7	64.0	marron	blond	Jules	9
## 162	Gabriel	162.5	75.6	vert	noir	Jules	9
## 163	Louise	166.1	80.7	vert	noir	Jules	9
## 164	Raphael	169.7	78.0	noir	blond	Jules	9
## 165	Alice	154.3	59.2	vert	noir	Jules	9
## 166	Leo	170.0	69.3	bleu	blond	Jules	9
## 167	Chloe	169.4	83.2	noir	noir	Jules	9
## 168	Louis	168.7	58.0	noir	chatain	Jules	9
## 169	Lina	151.7	70.1	bleu	chatain	Jules	9
## 170	Lucas	159.0	73.9	vert	chatain	Jules	9
## 171	Rose	178.8	80.6	vert	chatain	Jules	9
## 172	Adam	176.9	73.9	bleu	noir	Jules	9
## 173	Lea	175.9	82.3	vert	blond	Jules	9
## 174	Arthur	154.1	78.6	marron	noir	Jules	9
## 175	Anna	161.4	59.8	vert	blond	Jules	9
## 176	Hugo	171.5	89.4	vert	noir	Jules	9
## 177	Mila	175.9	68.3	noir	blond	Jules	9
## 178	Jules	176.5	53.2	noir	blond	Jules	9
## 179	Mia	160.5	58.9	vert	chatain	Jules	9
## 180	Mael	175.1	88.8	noir	chatain	Jules	9
## 181	Emmma	163.7	64.0	marron	blond	Lea	10
## 182	Gabriel	162.5	75.6	vert	noir	Lea	10
## 183	Louise	166.1	80.7	vert	noir	Lea	10
## 184	Raphael	169.7	78.0	noir	blond	Lea	10
## 185	Alice	154.3	59.2	vert	noir	Lea	10
## 186	Leo	170.0	69.3	bleu	blond	Lea	10
## 187	Chloe	169.4	83.2	noir	noir	Lea	10
## 188	Louis	168.7	58.0	noir	chatain	Lea	10
## 189	Lina	151.7	70.1	bleu	chatain	Lea	10
## 190	Lucas	159.0	73.9	vert	chatain	Lea	10
## 191	Rose	178.8	80.6	vert	chatain	Lea	10
## 192	Adam	176.9	73.9	bleu	noir	Lea	10
## 193	Lea	175.9	82.3	vert	blond	Lea	10
## 194	Arthur	154.1	78.6	marron	noir	Lea	10
## 195	Anna	161.4	59.8	vert	blond	Lea	10
## 196	Hugo	171.5	89.4	vert	noir	Lea	10
## 197	Mila	175.9	68.3	noir	blond	Lea	10
## 198	Jules	176.5	53.2	noir	blond	Lea	10
## 199	Mia	160.5	58.9	vert	chatain	Lea	10
## 200	Mael	175.1	88.8	noir	chatain	Lea	10
## 201	Emmma	163.7	64.0	marron	blond	Leo	11
## 202	Gabriel	162.5	75.6	vert	noir	Leo	11
## 203	Louise	166.1	80.7	vert	noir	Leo	11
## 204	Raphael	169.7	78.0	noir	blond	Leo	11
## 205	Alice	154.3	59.2	vert	noir	Leo	11
## 206	Leo	170.0	69.3	bleu	blond	Leo	11
## 207	Chloe	169.4	83.2	noir	noir	Leo	11
## 208	Louis	168.7	58.0	noir	chatain	Leo	11
## 209	Lina	151.7	70.1	bleu	chatain	Leo	11

## 210	Lucas	159.0	73.9	vert	chatain	Leo	11
## 211	Rose	178.8	80.6	vert	chatain	Leo	11
## 212	Adam	176.9	73.9	bleu	noir	Leo	11
## 213	Lea	175.9	82.3	vert	blond	Leo	11
## 214	Arthur	154.1	78.6	marron	noir	Leo	11
## 215	Anna	161.4	59.8	vert	blond	Leo	11
## 216	Hugo	171.5	89.4	vert	noir	Leo	11
## 217	Mila	175.9	68.3	noir	blond	Leo	11
## 218	Jules	176.5	53.2	noir	blond	Leo	11
## 219	Mia	160.5	58.9	vert	chatain	Leo	11
## 220	Mael	175.1	88.8	noir	chatain	Leo	11
## 221	Emmma	163.7	64.0	marron	blond	Lina	12
## 222	Gabriel	162.5	75.6	vert	noir	Lina	12
## 223	Louise	166.1	80.7	vert	noir	Lina	12
## 224	Raphael	169.7	78.0	noir	blond	Lina	12
## 225	Alice	154.3	59.2	vert	noir	Lina	12
## 226	Leo	170.0	69.3	bleu	blond	Lina	12
## 227	Chloe	169.4	83.2	noir	noir	Lina	12
## 228	Louis	168.7	58.0	noir	chatain	Lina	12
## 229	Lina	151.7	70.1	bleu	chatain	Lina	12
## 230	Lucas	159.0	73.9	vert	chatain	Lina	12
## 231	Rose	178.8	80.6	vert	chatain	Lina	12
## 232	Adam	176.9	73.9	bleu	noir	Lina	12
## 233	Lea	175.9	82.3	vert	blond	Lina	12
## 234	Arthur	154.1	78.6	marron	noir	Lina	12
## 235	Anna	161.4	59.8	vert	blond	Lina	12
## 236	Hugo	171.5	89.4	vert	noir	Lina	12
## 237	Mila	175.9	68.3	noir	blond	Lina	12
## 238	Jules	176.5	53.2	noir	blond	Lina	12
## 239	Mia	160.5	58.9	vert	chatain	Lina	12
## 240	Mael	175.1	88.8	noir	chatain	Lina	12
## 241	Emmma	163.7	64.0	marron	blond	Louis	13
## 242	Gabriel	162.5	75.6	vert	noir	Louis	13
## 243	Louise	166.1	80.7	vert	noir	Louis	13
## 244	Raphael	169.7	78.0	noir	blond	Louis	13
## 245	Alice	154.3	59.2	vert	noir	Louis	13
## 246	Leo	170.0	69.3	bleu	blond	Louis	13
## 247	Chloe	169.4	83.2	noir	noir	Louis	13
## 248	Louis	168.7	58.0	noir	chatain	Louis	13
## 249	Lina	151.7	70.1	bleu	chatain	Louis	13
## 250	Lucas	159.0	73.9	vert	chatain	Louis	13
## 251	Rose	178.8	80.6	vert	chatain	Louis	13
## 252	Adam	176.9	73.9	bleu	noir	Louis	13
## 253	Lea	175.9	82.3	vert	blond	Louis	13
## 254	Arthur	154.1	78.6	marron	noir	Louis	13
## 255	Anna	161.4	59.8	vert	blond	Louis	13
## 256	Hugo	171.5	89.4	vert	noir	Louis	13
## 257	Mila	175.9	68.3	noir	blond	Louis	13
## 258	Jules	176.5	53.2	noir	blond	Louis	13
## 259	Mia	160.5	58.9	vert	chatain	Louis	13
## 260	Mael	175.1	88.8	noir	chatain	Louis	13
## 261	Emmma	163.7	64.0	marron	blond	Louise	14
## 262	Gabriel	162.5	75.6	vert	noir	Louise	14
## 263	Louise	166.1	80.7	vert	noir	Louise	14

## 264	Raphael	169.7	78.0	noir	blond	Louise	14
## 265	Alice	154.3	59.2	vert	noir	Louise	14
## 266	Leo	170.0	69.3	bleu	blond	Louise	14
## 267	Chloe	169.4	83.2	noir	noir	Louise	14
## 268	Louis	168.7	58.0	noir	chatain	Louise	14
## 269	Lina	151.7	70.1	bleu	chatain	Louise	14
## 270	Lucas	159.0	73.9	vert	chatain	Louise	14
## 271	Rose	178.8	80.6	vert	chatain	Louise	14
## 272	Adam	176.9	73.9	bleu	noir	Louise	14
## 273	Lea	175.9	82.3	vert	blond	Louise	14
## 274	Arthur	154.1	78.6	marron	noir	Louise	14
## 275	Anna	161.4	59.8	vert	blond	Louise	14
## 276	Hugo	171.5	89.4	vert	noir	Louise	14
## 277	Mila	175.9	68.3	noir	blond	Louise	14
## 278	Jules	176.5	53.2	noir	blond	Louise	14
## 279	Mia	160.5	58.9	vert	chatain	Louise	14
## 280	Mael	175.1	88.8	noir	chatain	Louise	14
## 281	Emmma	163.7	64.0	marron	blond	Lucas	15
## 282	Gabriel	162.5	75.6	vert	noir	Lucas	15
## 283	Louise	166.1	80.7	vert	noir	Lucas	15
## 284	Raphael	169.7	78.0	noir	blond	Lucas	15
## 285	Alice	154.3	59.2	vert	noir	Lucas	15
## 286	Leo	170.0	69.3	bleu	blond	Lucas	15
## 287	Chloe	169.4	83.2	noir	noir	Lucas	15
## 288	Louis	168.7	58.0	noir	chatain	Lucas	15
## 289	Lina	151.7	70.1	bleu	chatain	Lucas	15
## 290	Lucas	159.0	73.9	vert	chatain	Lucas	15
## 291	Rose	178.8	80.6	vert	chatain	Lucas	15
## 292	Adam	176.9	73.9	bleu	noir	Lucas	15
## 293	Lea	175.9	82.3	vert	blond	Lucas	15
## 294	Arthur	154.1	78.6	marron	noir	Lucas	15
## 295	Anna	161.4	59.8	vert	blond	Lucas	15
## 296	Hugo	171.5	89.4	vert	noir	Lucas	15
## 297	Mila	175.9	68.3	noir	blond	Lucas	15
## 298	Jules	176.5	53.2	noir	blond	Lucas	15
## 299	Mia	160.5	58.9	vert	chatain	Lucas	15
## 300	Mael	175.1	88.8	noir	chatain	Lucas	15
## 301	Emmma	163.7	64.0	marron	blond	Mael	16
## 302	Gabriel	162.5	75.6	vert	noir	Mael	16
## 303	Louise	166.1	80.7	vert	noir	Mael	16
## 304	Raphael	169.7	78.0	noir	blond	Mael	16
## 305	Alice	154.3	59.2	vert	noir	Mael	16
## 306	Leo	170.0	69.3	bleu	blond	Mael	16
## 307	Chloe	169.4	83.2	noir	noir	Mael	16
## 308	Louis	168.7	58.0	noir	chatain	Mael	16
## 309	Lina	151.7	70.1	bleu	chatain	Mael	16
## 310	Lucas	159.0	73.9	vert	chatain	Mael	16
## 311	Rose	178.8	80.6	vert	chatain	Mael	16
## 312	Adam	176.9	73.9	bleu	noir	Mael	16
## 313	Lea	175.9	82.3	vert	blond	Mael	16
## 314	Arthur	154.1	78.6	marron	noir	Mael	16
## 315	Anna	161.4	59.8	vert	blond	Mael	16
## 316	Hugo	171.5	89.4	vert	noir	Mael	16
## 317	Mila	175.9	68.3	noir	blond	Mael	16

## 318	Jules	176.5	53.2	noir	blond	Mael	16
## 319	Mia	160.5	58.9	vert	chatain	Mael	16
## 320	Mael	175.1	88.8	noir	chatain	Mael	16
## 321	Emmma	163.7	64.0	marron	blond	Mia	17
## 322	Gabriel	162.5	75.6	vert	noir	Mia	17
## 323	Louise	166.1	80.7	vert	noir	Mia	17
## 324	Raphael	169.7	78.0	noir	blond	Mia	17
## 325	Alice	154.3	59.2	vert	noir	Mia	17
## 326	Leo	170.0	69.3	bleu	blond	Mia	17
## 327	Chloe	169.4	83.2	noir	noir	Mia	17
## 328	Louis	168.7	58.0	noir	chatain	Mia	17
## 329	Lina	151.7	70.1	bleu	chatain	Mia	17
## 330	Lucas	159.0	73.9	vert	chatain	Mia	17
## 331	Rose	178.8	80.6	vert	chatain	Mia	17
## 332	Adam	176.9	73.9	bleu	noir	Mia	17
## 333	Lea	175.9	82.3	vert	blond	Mia	17
## 334	Arthur	154.1	78.6	marron	noir	Mia	17
## 335	Anna	161.4	59.8	vert	blond	Mia	17
## 336	Hugo	171.5	89.4	vert	noir	Mia	17
## 337	Mila	175.9	68.3	noir	blond	Mia	17
## 338	Jules	176.5	53.2	noir	blond	Mia	17
## 339	Mia	160.5	58.9	vert	chatain	Mia	17
## 340	Mael	175.1	88.8	noir	chatain	Mia	17
## 341	Emmma	163.7	64.0	marron	blond	Mila	18
## 342	Gabriel	162.5	75.6	vert	noir	Mila	18
## 343	Louise	166.1	80.7	vert	noir	Mila	18
## 344	Raphael	169.7	78.0	noir	blond	Mila	18
## 345	Alice	154.3	59.2	vert	noir	Mila	18
## 346	Leo	170.0	69.3	bleu	blond	Mila	18
## 347	Chloe	169.4	83.2	noir	noir	Mila	18
## 348	Louis	168.7	58.0	noir	chatain	Mila	18
## 349	Lina	151.7	70.1	bleu	chatain	Mila	18
## 350	Lucas	159.0	73.9	vert	chatain	Mila	18
## 351	Rose	178.8	80.6	vert	chatain	Mila	18
## 352	Adam	176.9	73.9	bleu	noir	Mila	18
## 353	Lea	175.9	82.3	vert	blond	Mila	18
## 354	Arthur	154.1	78.6	marron	noir	Mila	18
## 355	Anna	161.4	59.8	vert	blond	Mila	18
## 356	Hugo	171.5	89.4	vert	noir	Mila	18
## 357	Mila	175.9	68.3	noir	blond	Mila	18
## 358	Jules	176.5	53.2	noir	blond	Mila	18
## 359	Mia	160.5	58.9	vert	chatain	Mila	18
## 360	Mael	175.1	88.8	noir	chatain	Mila	18
## 361	Emmma	163.7	64.0	marron	blond	Raphael	19
## 362	Gabriel	162.5	75.6	vert	noir	Raphael	19
## 363	Louise	166.1	80.7	vert	noir	Raphael	19
## 364	Raphael	169.7	78.0	noir	blond	Raphael	19
## 365	Alice	154.3	59.2	vert	noir	Raphael	19
## 366	Leo	170.0	69.3	bleu	blond	Raphael	19
## 367	Chloe	169.4	83.2	noir	noir	Raphael	19
## 368	Louis	168.7	58.0	noir	chatain	Raphael	19
## 369	Lina	151.7	70.1	bleu	chatain	Raphael	19
## 370	Lucas	159.0	73.9	vert	chatain	Raphael	19
## 371	Rose	178.8	80.6	vert	chatain	Raphael	19

```

## 372 Adam 176.9 73.9 bleu noir Raphael 19
## 373 Lea 175.9 82.3 vert blond Raphael 19
## 374 Arthur 154.1 78.6 marron noir Raphael 19
## 375 Anna 161.4 59.8 vert blond Raphael 19
## 376 Hugo 171.5 89.4 vert noir Raphael 19
## 377 Mila 175.9 68.3 noir blond Raphael 19
## 378 Jules 176.5 53.2 noir blond Raphael 19
## 379 Mia 160.5 58.9 vert chatain Raphael 19
## 380 Mael 175.1 88.8 noir chatain Raphael 19
## 381 Emma 163.7 64.0 marron blond Rose 20
## 382 Gabriel 162.5 75.6 vert noir Rose 20
## 383 Louise 166.1 80.7 vert noir Rose 20
## 384 Raphael 169.7 78.0 noir blond Rose 20
## 385 Alice 154.3 59.2 vert noir Rose 20
## 386 Leo 170.0 69.3 bleu blond Rose 20
## 387 Chloe 169.4 83.2 noir noir Rose 20
## 388 Louis 168.7 58.0 noir chatain Rose 20
## 389 Lina 151.7 70.1 bleu chatain Rose 20
## 390 Lucas 159.0 73.9 vert chatain Rose 20
## 391 Rose 178.8 80.6 vert chatain Rose 20
## 392 Adam 176.9 73.9 bleu noir Rose 20
## 393 Lea 175.9 82.3 vert blond Rose 20
## 394 Arthur 154.1 78.6 marron noir Rose 20
## 395 Anna 161.4 59.8 vert blond Rose 20
## 396 Hugo 171.5 89.4 vert noir Rose 20
## 397 Mila 175.9 68.3 noir blond Rose 20
## 398 Jules 176.5 53.2 noir blond Rose 20
## 399 Mia 160.5 58.9 vert chatain Rose 20
## 400 Mael 175.1 88.8 noir chatain Rose 20

```

```

merge(MonDataFramePrenom, UnAutreDataFramePrenom,
      by.x = "Prenom", by.y = "FirstName")

```

```

## Prenom taille masse yeux cheveux Note
## 1 Adam 176.9 73.9 bleu noir 1
## 2 Alice 154.3 59.2 vert noir 2
## 3 Anna 161.4 59.8 vert blond 3
## 4 Arthur 154.1 78.6 marron noir 4
## 5 Chloe 169.4 83.2 noir noir 5
## 6 Emma 163.7 64.0 marron blond 6
## 7 Gabriel 162.5 75.6 vert noir 7
## 8 Hugo 171.5 89.4 vert noir 8
## 9 Jules 176.5 53.2 noir blond 9
## 10 Lea 175.9 82.3 vert blond 10
## 11 Leo 170.0 69.3 bleu blond 11
## 12 Lina 151.7 70.1 bleu chatain 12
## 13 Louis 168.7 58.0 noir chatain 13
## 14 Louise 166.1 80.7 vert noir 14
## 15 Lucas 159.0 73.9 vert chatain 15
## 16 Mael 175.1 88.8 noir chatain 16
## 17 Mia 160.5 58.9 vert chatain 17
## 18 Mila 175.9 68.3 noir blond 18
## 19 Raphael 169.7 78.0 noir blond 19
## 20 Rose 178.8 80.6 vert chatain 20

```

- Les 2 tableaux peuvent ne pas avoir le même nombre de lignes. Dans ce cas, il faut préciser sur quel tableau on souhaite s'aligner.

```
UnAutreDataFramePrenom <- data.frame(FirstName = sort(Prenom[1:18]), Note = 1:18)
EssaiMerge1 <- merge(MonDataFramePrenom, UnAutreDataFramePrenom,
  by.x = "Prenom", by.y = "FirstName")
dim(EssaiMerge1)
```

```
## [1] 18 6
```

```
EssaiMerge1
```

```
##      Prenom taille masse  yeux cheveux Note
## 1      Adam  176.9  73.9  bleu   noir    1
## 2      Alice  154.3  59.2  vert   noir    2
## 3      Anna  161.4  59.8  vert   blond   3
## 4      Arthur 154.1  78.6 marron noir    4
## 5      Chloe  169.4  83.2 noir   noir    5
## 6      Emma   163.7  64.0 marron blond   6
## 7      Gabriel 162.5  75.6 vert   noir    7
## 8      Hugo   171.5  89.4 vert   noir    8
## 9      Jules  176.5  53.2 noir   blond   9
## 10     Lea    175.9  82.3 vert   blond  10
## 11     Leo    170.0  69.3 bleu   blond  11
## 12     Lina   151.7  70.1 bleu  chatain 12
## 13     Louis  168.7  58.0 noir  chatain 13
## 14     Louise 166.1  80.7 vert   noir    14
## 15     Lucas  159.0  73.9 vert  chatain 15
## 16     Mila   175.9  68.3 noir   blond  16
## 17     Raphael 169.7  78.0 noir   blond  17
## 18     Rose   178.8  80.6 vert  chatain 18
```

Noter ici que l'on a perdu 2 éléments car le data.frame UnAutreDataFramePrenom n'a que 18 lignes.

Pour conserver les 20 lignes du data.frame MonDataFramePrenom, il faut le spécifier.

```
EssaiMerge2 <- merge(MonDataFramePrenom, UnAutreDataFramePrenom,
  by.x = "Prenom", by.y = "FirstName", all.x=TRUE)
dim(EssaiMerge2)
```

```
## [1] 20 6
```

```
EssaiMerge2
```

```
##      Prenom taille masse  yeux cheveux Note
## 1      Adam  176.9  73.9  bleu   noir    1
## 2      Alice  154.3  59.2  vert   noir    2
## 3      Anna  161.4  59.8  vert   blond   3
## 4      Arthur 154.1  78.6 marron noir    4
## 5      Chloe  169.4  83.2 noir   noir    5
## 6      Emma   163.7  64.0 marron blond   6
## 7      Gabriel 162.5  75.6 vert   noir    7
## 8      Hugo   171.5  89.4 vert   noir    8
## 9      Jules  176.5  53.2 noir   blond   9
## 10     Lea    175.9  82.3 vert   blond  10
## 11     Leo    170.0  69.3 bleu   blond  11
## 12     Lina   151.7  70.1 bleu  chatain 12
## 13     Louis  168.7  58.0 noir  chatain 13
```

```
## 14 Louise 166.1 80.7 vert noir 14
## 15 Lucas 159.0 73.9 vert chatain 15
## 16 Mael 175.1 88.8 noir chatain NA
## 17 Mia 160.5 58.9 vert chatain NA
## 18 Mila 175.9 68.3 noir blond 16
## 19 Raphael 169.7 78.0 noir blond 17
## 20 Rose 178.8 80.6 vert chatain 18
```

On peut alors constater que les données correspondant aux 2 individus manquants ont été notées comme NA (données manquantes).

Le package `dplyr` propose un ensemble de fonctions permettant de fusionner des tableaux dans des circonstances plus spécifiques et moins favorables.

Un peu de statistique

Statistique univariée

La fonction `summary`

```
summary(MonDataFrame)
```

```
##      taille      masse      yeux      cheveux
## Min.   :151.7   Min.   :53.20   Length:20   Length:20
## 1st Qu.:161.2   1st Qu.:62.95   Class :character   Class :character
## Median :169.1   Median :73.90   Mode  :character   Mode  :character
## Mean   :167.1   Mean    :72.29
## 3rd Qu.:175.3   3rd Qu.:80.62
## Max.   :178.8   Max.    :89.40
```

Si les variables sont enregistrées comme des facteurs, le `summary` est plus intéressant.

```
MonDataFrame$yeux <- as.factor(MonDataFrame$yeux)
MonDataFrame$cheveux <- as.factor(MonDataFrame$cheveux)
summary(MonDataFrame)
```

```
##      taille      masse      yeux      cheveux
## Min.   :151.7   Min.   :53.20   bleu :3   blond :7
## 1st Qu.:161.2   1st Qu.:62.95   marron:2   chatain:6
## Median :169.1   Median :73.90   noir  :6   noir  :7
## Mean   :167.1   Mean    :72.29   vert  :9
## 3rd Qu.:175.3   3rd Qu.:80.62
## Max.   :178.8   Max.    :89.40
```

Tri à plat, tri croisé

```
table(MonDataFrame$yeux)
```

```
##
##  bleu marron  noir  vert
##    3     2     6     9
```

```
table(MonDataFrame$yeux, MonDataFrame$cheveux)
```

```
##
##      blond chatain noir
##  bleu      1      1      1
```

```
## marron      1      0      1
## noir       3      2      1
## vert       2      3      4
```

La fonction apply

```
apply(MonDataFrame[,1:2], 2 , mean)
```

```
## taille  masse
## 167.085  72.290
```

```
apply(MonDataFrame[,1:2], 2 , var)
```

```
##   taille    masse
## 69.89082 115.08200
```

```
apply(MonDataFrame[,1:2], 2 , sd)^2
```

```
##   taille    masse
## 69.89082 115.08200
```

Statistique bivariée

Le calcul d'indicateurs de liaison entre deux variables quantitatives est possible à l'aide des fonctions cov et cor.

```
cov(MonDataFrame$taille, MonDataFrame$masse)
```

```
## [1] 24.87511
```

```
MyCor <- cor(MonDataFrame$taille, MonDataFrame$masse)
round(MyCor, digits = 2)
```

```
## [1] 0.28
```

```
cor(MonDataFrame$taille, MonDataFrame$masse, method="spearman")
```

```
## [1] 0.2460497
```

```
cor(MonDataFrame$taille, MonDataFrame$masse, method="kendall")
```

```
## [1] 0.1693122
```

Face à un data.frame, ces fonctions renvoient l'ensemble de la matrice de covariance ou de corrélation, entre les colonnes deux-à-deux.

Sur les données iris restreintes aux 4 premières colonnes (la cinquième n'étant pas quantitative), cela donne.

```
cov(iris[,1:4])
```

```
##           Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length  0.6856935 -0.0424340  1.2743154  0.5162707
## Sepal.Width  -0.0424340  0.1899794 -0.3296564 -0.1216394
## Petal.Length  1.2743154 -0.3296564  3.1162779  1.2956094
## Petal.Width   0.5162707 -0.1216394  1.2956094  0.5810063
```

```
round(cor(iris[, -5]), 3)
```

```
##           Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length      1.000      -0.118      0.872      0.818
## Sepal.Width      -0.118      1.000      -0.428     -0.366
```

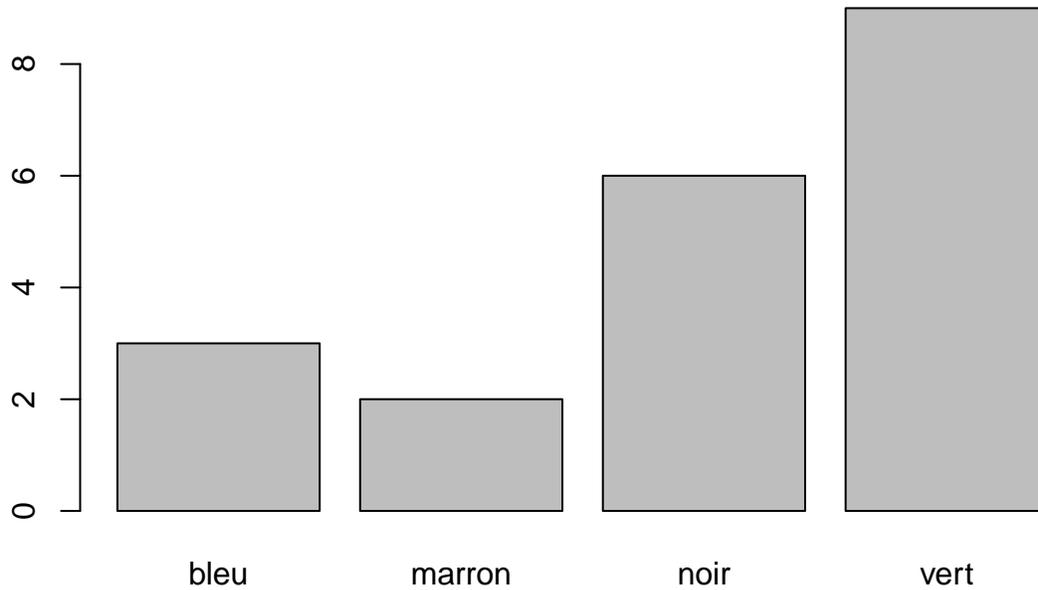
```
## Petal.Length      0.872      -0.428      1.000      0.963
## Petal.Width       0.818      -0.366      0.963      1.000
```

Représentations graphiques

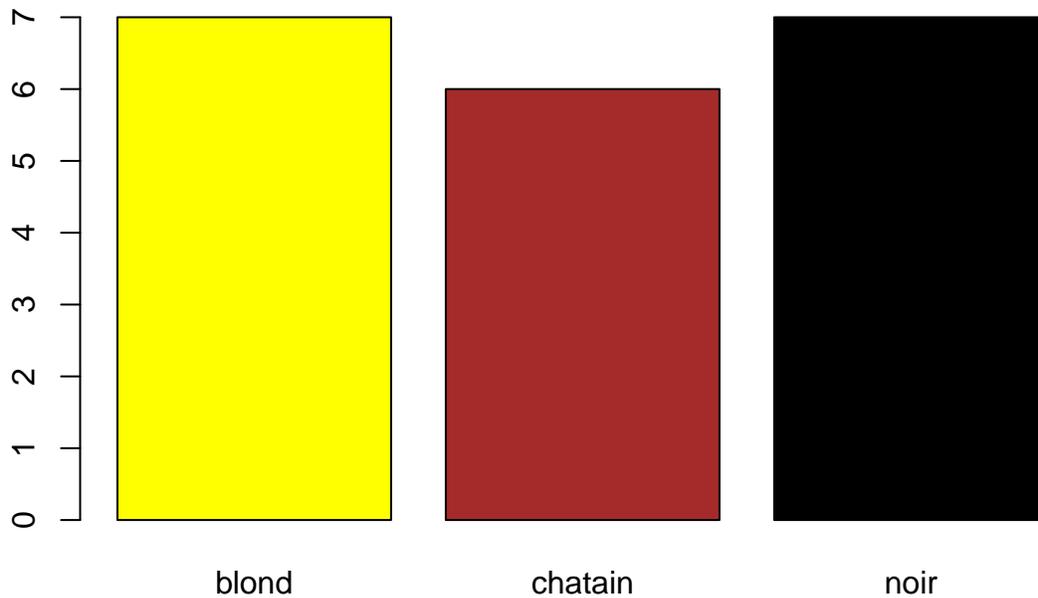
L'ensemble des représentations graphiques classiques sont disponibles dans R.

Pour les effectifs selon les modalités d'un variable qualitative

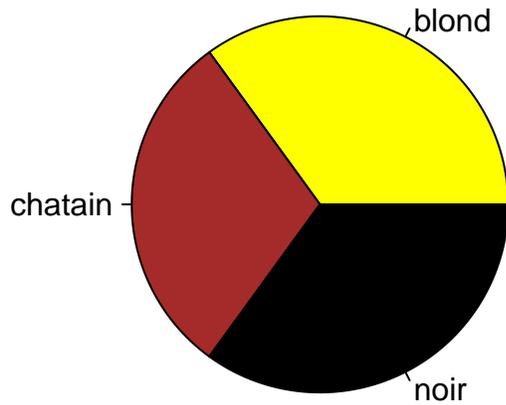
```
EffectifCouleurYeux <- table(MonDataFrame$yeux)
barplot(EffectifCouleurYeux)
```



```
EffectifCouleurCheveux <- table(MonDataFrame$cheveux)
barplot(EffectifCouleurCheveux, col = c("yellow", "brown", "black"))
```



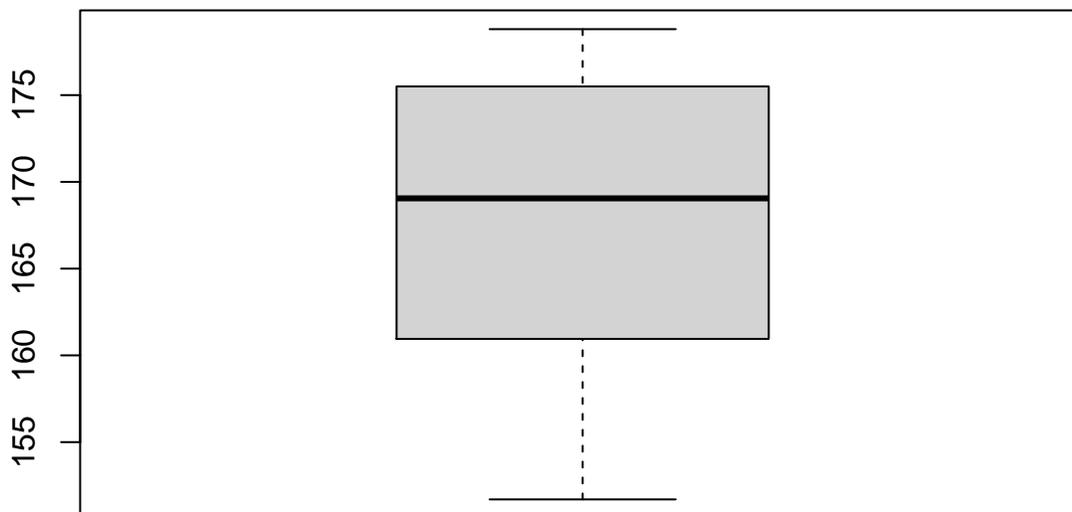
```
pie(EffectifCouleurCheveux, col = c("yellow","brown","black"))
```



Les graphiques sont très fortement personnalisables. Les exemples de l'aide en ligne des fonctions proposées en proposent des illustrations diverses et variées.

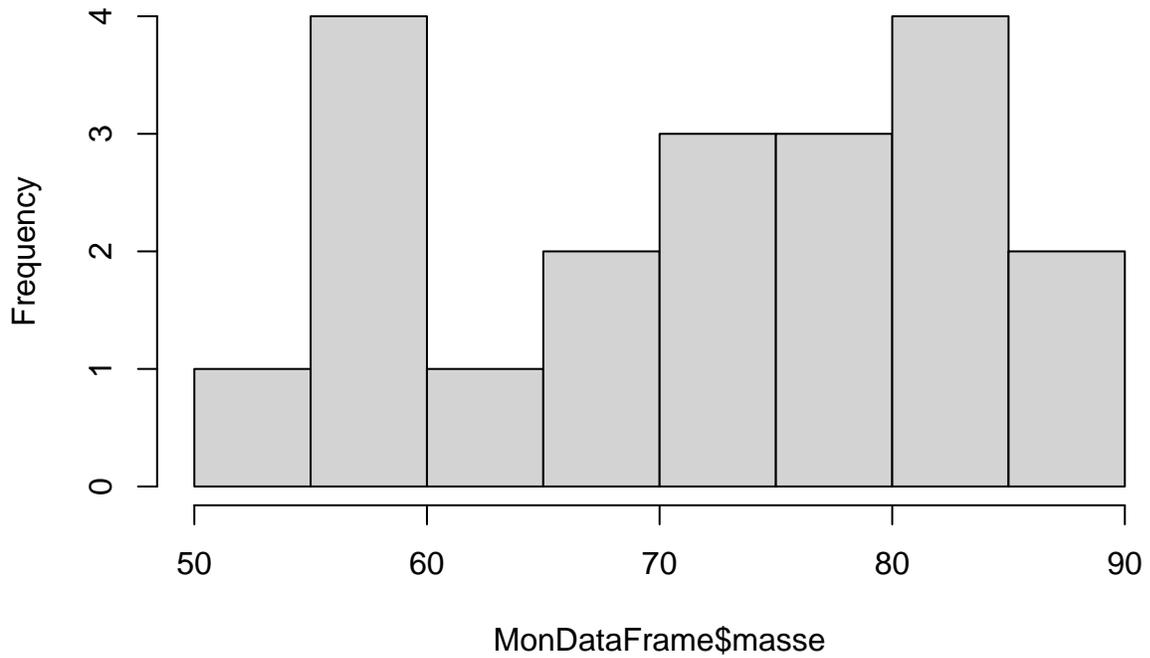
Pour des variables quantitatives

```
boxplot(MonDataFrame$taille)
```

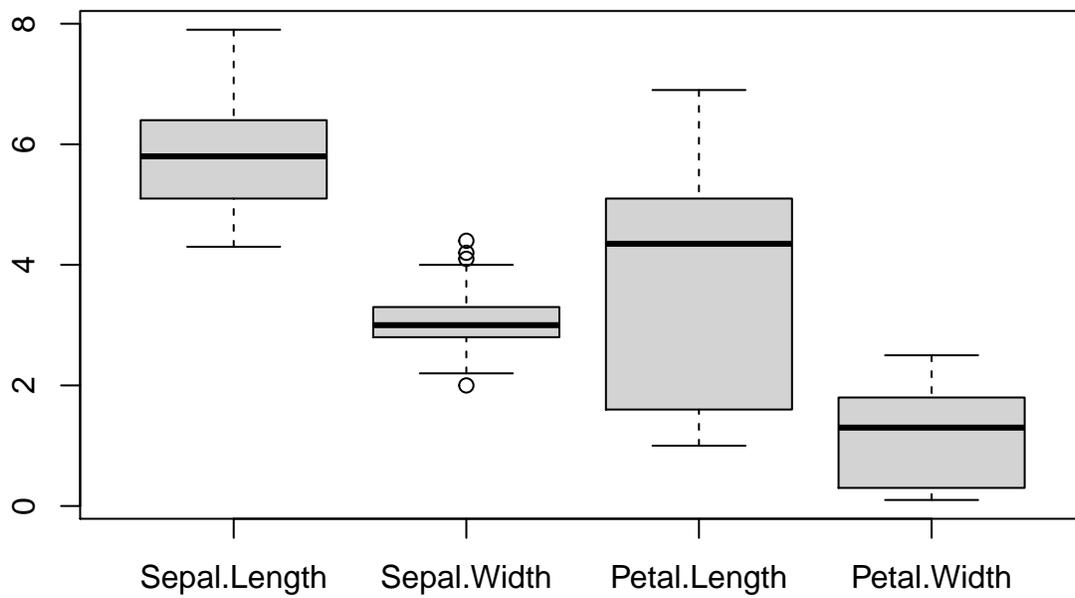


```
hist(MonDataFrame$masse)
```

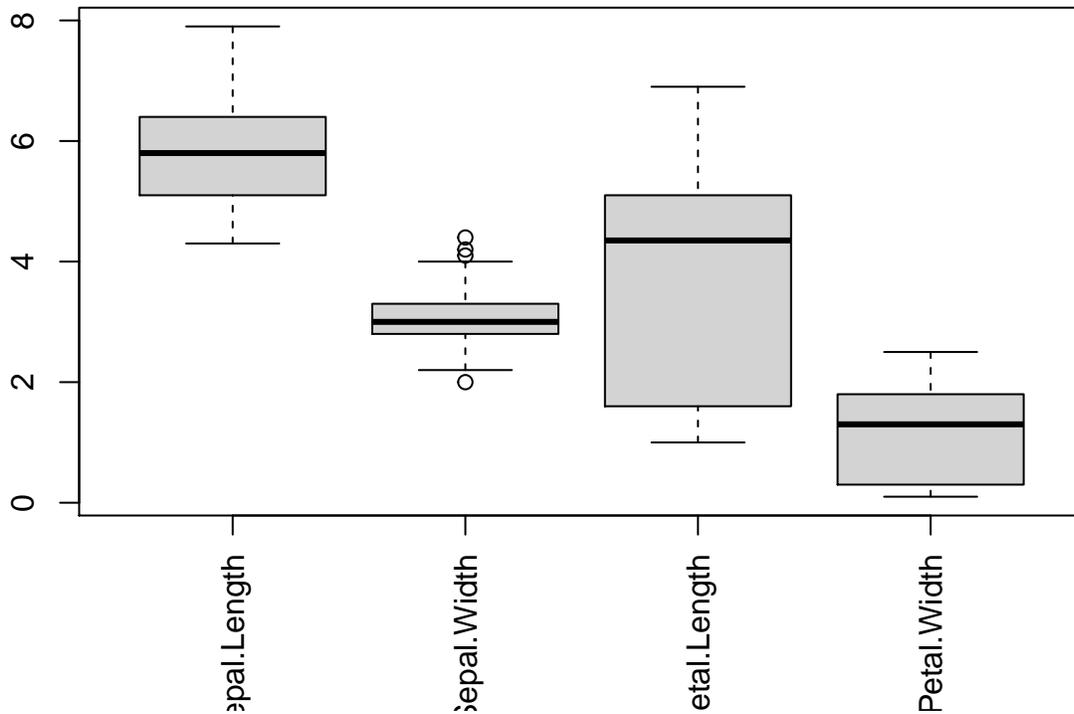
Histogram of MonDataFrame\$masse



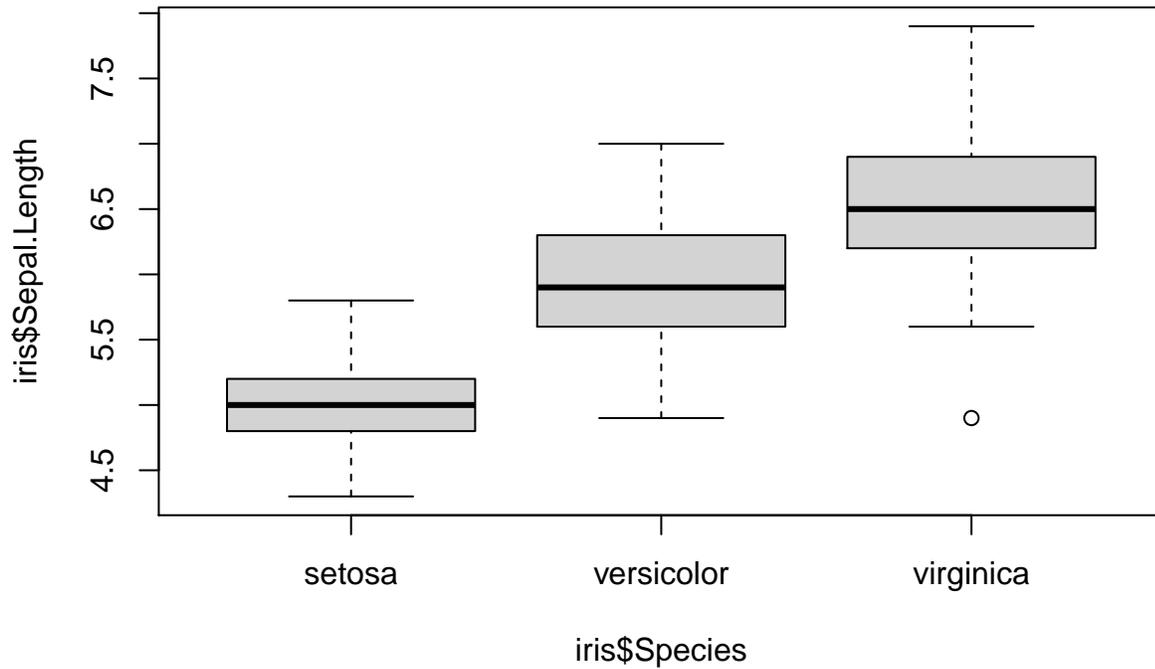
```
boxplot(iris[,-5])
```



```
boxplot(iris[,-5], las=3)
```



```
boxplot(iris$Sepal.Length~iris$Species)
```

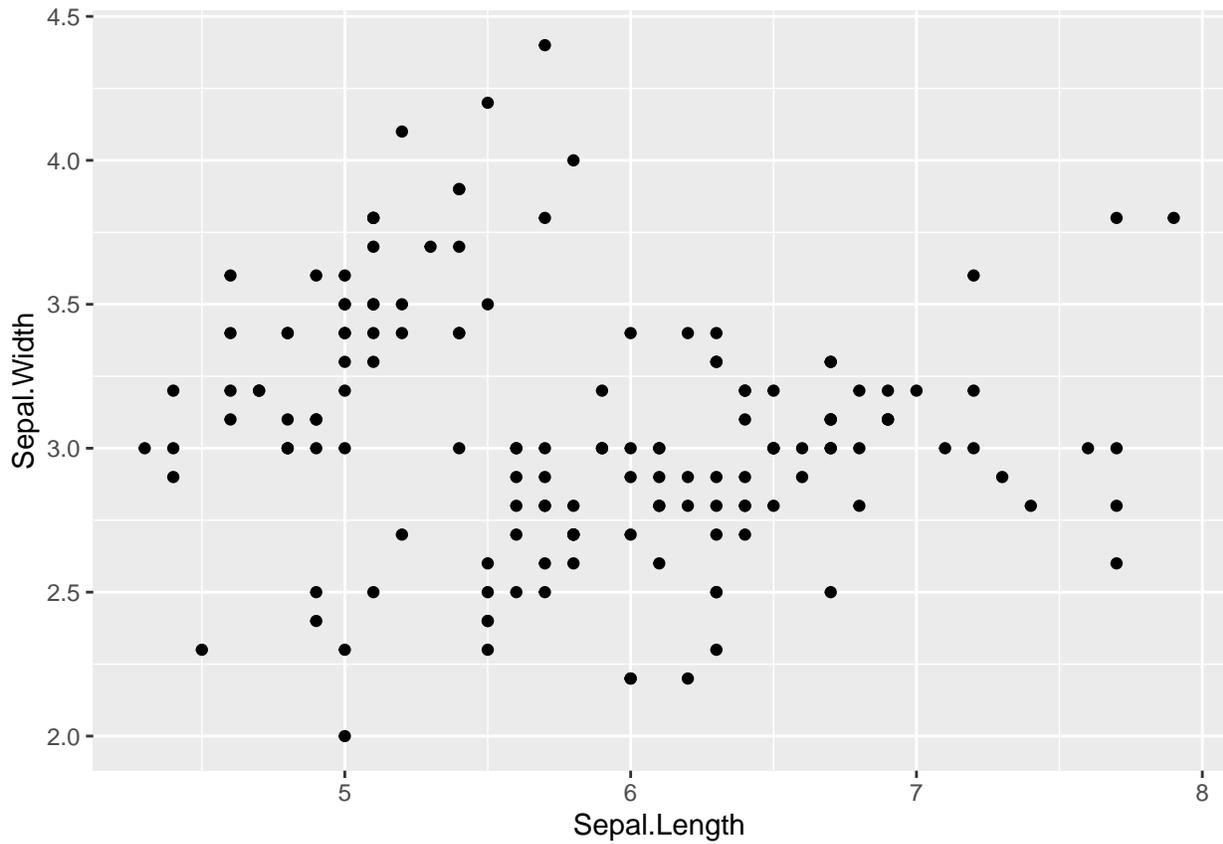


Là aussi, de nombreuses options permettent de personnaliser ces graphiques.

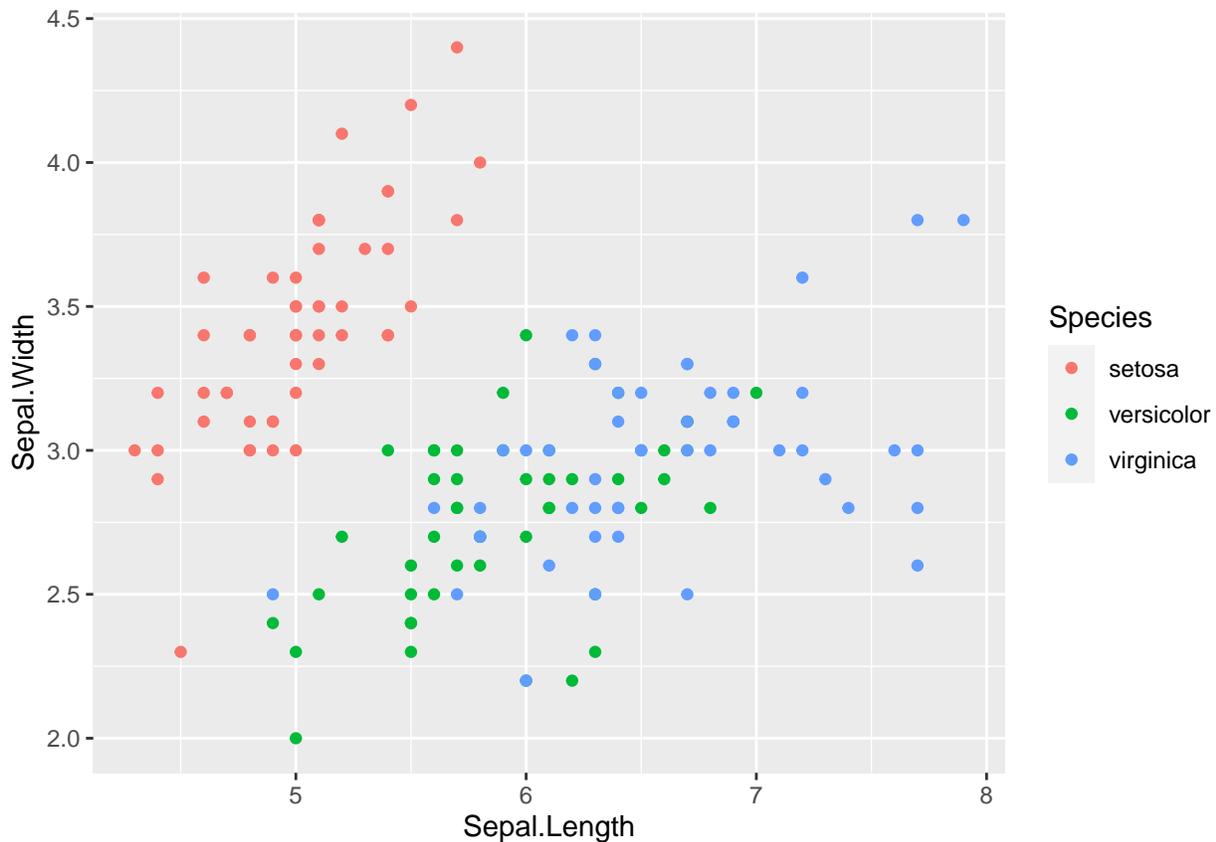
ggplot2

Le package `ggplot2` propose une autre approche pour la réalisation de graphiques. Elle est basée sur une nouvelle **grammar of graphics** (le `gg` de `ggplot2`). En voici une petite illustration.

```
library(ggplot2)
ggplot(data = iris, aes(x=Sepal.Length, y=Sepal.Width)) + geom_point()
```



```
ggplot(data = iris, aes(x=Sepal.Length, y=Sepal.Width, colour=Species)) +  
geom_point()
```



Pour en savoir plus, un livre consacré à `ggplot2` est disponible en ligne : ggplot2-book.org.

Etude de cas

Les données que nous regardons ici sont disponibles et décrites à cette adresse : data.enseignementsup-recherche.gouv.fr/explore/dataset/fr-esr-parcours-des-bacheliers-en-paces/information/

Importation des données

```
library(readr)
fr_esr_parcours_des_bacheliers_en_paces <- read_csv2("fr-esr-parcours-des-bacheliers-en-paces.csv")

## Using ',' as decimal and '.' as grouping mark. Use read_delim() for more control.
## Parsed with column specification:
## cols(
##   `Id Série ou type de Bac` = col_double(),
##   `Série ou type de Bac` = col_character(),
##   `Id Âge au bac` = col_character(),
##   `Âge au bac` = col_character(),
##   `Id Sexe` = col_double(),
##   Sexe = col_character(),
##   `Id Mention au Bac` = col_character(),
##   `Mention au Bac` = col_character(),
##   `Année de cohorte (Année de première inscription)` = col_double(),
##   `Effectif de néobacheliers de la cohorte` = col_double(),
##   `Passage en 2ème année d'études de santé en 1 an` = col_double(),
```

```

## `Passage en 2ème année d'études de santé en 2 ans` = col_double(),
## `Passage en 2ème année d'études de santé en 1 ou 2 ans` = col_double(),
## `Passage en 2ème année d'études de santé dans la filière Médecine en 1 ou 2 ans` = col_double(),
## `Passage en 2ème année d'études de santé dans la filière Pharmacie en 1 ou 2 ans` = col_double(),
## `Passage en 2ème année d'études de santé dans la filière Chirurgie dentaire en 1 ou 2 ans` = col_double(),
## `Passage en 2ème année d'études de santé dans la filière Maïeutique en 1 ou 2 ans` = col_double(),
## Redoublement = col_double(),
## `Passage dans d'autres diplômes de santé` = col_double()
## )

```

Premier aperçu

```
fr_esr_parcours_des_bacheliers_en_paces
```

```

## # A tibble: 150 x 19
##   `Id Série ou ty~` `Série ou type ~` `Id Âge au bac` `Âge au bac` `Id Sexe`
##   <dbl> <chr>           <chr>           <chr>           <dbl>
## 1         2 BAC ES          R1              En retard d-     1
## 2         2 BAC ES          R2              En retard d-     2
## 3         3 BAC S           R1              En retard d-     1
## 4         5 BAC technologi~ R1              En retard d-     1
## 5         1 BAC L            R1              En retard d-     1
## 6         2 BAC ES          R0              A l'heure o-     1
## 7         3 BAC S           R1              En retard d-     1
## 8         3 BAC S           R1              En retard d-     2
## 9         6 BAC professionn~ R4              Non pris en-     2
## 10        1 BAC L            R0              A l'heure o-     2
## # ... with 140 more rows, and 14 more variables: Sexe <chr>, `Id Mention au
## # Bac` <chr>, `Mention au Bac` <chr>, `Année de cohorte (Année de première
## # inscription)` <dbl>, `Effectif de néobacheliers de la cohorte` <dbl>,
## # `Passage en 2ème année d'études de santé en 1 an` <dbl>, `Passage en 2ème
## # année d'études de santé en 2 ans` <dbl>, `Passage en 2ème année d'études de
## # santé en 1 ou 2 ans` <dbl>, `Passage en 2ème année d'études de santé dans
## # la filière Médecine en 1 ou 2 ans` <dbl>, `Passage en 2ème année d'études
## # de santé dans la filière Pharmacie en 1 ou 2 ans` <dbl>, `Passage en 2ème
## # année d'études de santé dans la filière Chirurgie dentaire en 1 ou 2
## # ans` <dbl>, `Passage en 2ème année d'études de santé dans la filière
## # Maïeutique en 1 ou 2 ans` <dbl>, Redoublement <dbl>, `Passage dans d'autres
## # diplômes de santé` <dbl>

```

```
summary(fr_esr_parcours_des_bacheliers_en_paces)
```

```

## Id Série ou type de Bac Série ou type de Bac Id Âge au bac
## Min.      :1.00           Length:150           Length:150
## 1st Qu.:2.00             Class :character     Class :character
## Median :3.00             Mode  :character     Mode  :character
## Mean    :3.44
## 3rd Qu.:5.00
## Max.    :6.00
##   Âge au bac           Id Sexe           Sexe           Id Mention au Bac
## Length:150           Min.      :1.000     Length:150     Length:150
## Class :character     1st Qu.:1.000     Class :character Class :character
## Mode  :character     Median :2.000     Mode  :character Mode  :character
##                               Mean    :1.553
##                               3rd Qu.:2.000

```

```

##                               Max.    :2.000
## Mention au Bac               Année de cohorte (Année de première inscription)
## Length:150                   Min.    :2014
## Class :character             1st Qu.:2014
## Mode  :character             Median  :2014
##                               Mean    :2014
##                               3rd Qu.:2014
##                               Max.    :2014
## Effectif de néobacheliers de la cohorte
## Min.    : 1.00
## 1st Qu.: 3.25
## Median  : 10.00
## Mean    : 235.83
## 3rd Qu.: 44.50
## Max.    :5470.00
## Passage en 2ème année d'études de santé en 1 an
## Min.    : 0.00
## 1st Qu.: 0.00
## Median  : 0.00
## Mean    : 27.08
## 3rd Qu.: 0.00
## Max.    :1643.00
## Passage en 2ème année d'études de santé en 2 ans
## Min.    : 0.00
## 1st Qu.: 0.00
## Median  : 0.00
## Mean    : 52.37
## 3rd Qu.: 2.00
## Max.    :1912.00
## Passage en 2ème année d'études de santé en 1 ou 2 ans
## Min.    : 0.00
## 1st Qu.: 0.00
## Median  : 0.00
## Mean    : 79.35
## 3rd Qu.: 2.00
## Max.    :3146.00
## Passage en 2ème année d'études de santé dans la filière Médecine en 1 ou 2 ans
## Min.    : 0.00
## 1st Qu.: 0.00
## Median  : 0.00
## Mean    : 47.47
## 3rd Qu.: 1.00
## Max.    :2395.00
## Passage en 2ème année d'études de santé dans la filière Pharmacie en 1 ou 2 ans
## Min.    : 0.00
## 1st Qu.: 0.00
## Median  : 0.00
## Mean    : 18.83
## 3rd Qu.: 1.00
## Max.    :628.00
## Passage en 2ème année d'études de santé dans la filière Chirurgie dentaire en 1 ou 2 ans
## Min.    : 0.000
## 1st Qu.: 0.000
## Median  : 0.000

```

```
## Mean : 7.253
## 3rd Qu.: 0.000
## Max. :224.000
## Passage en 2ème année d'études de santé dans la filière Maïeutique en 1 ou 2 ans
## Min. : 0.000
## 1st Qu.: 0.000
## Median : 0.000
## Mean : 5.787
## 3rd Qu.: 0.000
## Max. :308.000
## Redoublement Passage dans d'autres diplômes de santé
## Min. : 0.0 Min. : 0.000
## 1st Qu.: 1.0 1st Qu.: 0.000
## Median : 3.0 Median : 0.000
## Mean : 111.1 Mean : 5.013
## 3rd Qu.: 13.5 3rd Qu.: 0.750
## Max. :3295.0 Max. :167.000
```

Un souci vient du fait que les colonnes comportant des chaînes de caractères (ou des codes) ont été (logiquement) importées au format `character`. Or, il semblerait préférable d'avoir des colonnes de `factor`. On va reprendre l'importation en spécifiant cela au moyen du paramètre `col_types`.

Importation des données : le retour

Utilisons donc l'argument `col_types`. L'aide en ligne de la fonction `read_delim` ou de la fonction `read_csv2` indique plusieurs façons de procéder. Nous utilisons ici l'option compacte en donnant une chaîne de caractères indiquant que les 9 premières colonnes contiennent des données de type `factor` et les 10 suivantes, des données de type `double`.

```
fr_esr_parcours_des_bacheliers_en_paces <- read_csv2("fr-esr-parcours-des-bacheliers-en-paces.csv",
  col_types = "fffffffddddd")
```

```
## Using ',' as decimal and '.' as grouping mark. Use read_delim() for more control.
```

Premier aperçu : le retour

```
fr_esr_parcours_des_bacheliers_en_paces

## # A tibble: 150 x 19
##   `Id Série ou ty~` `Série ou type ~` `Id Âge au bac` `Âge au bac` `Id Sexe`
##   <fct>           <fct>           <fct>           <fct>           <fct>
## 1 2              BAC ES             R1              En retard d~ 1
## 2 2              BAC ES             R2              En retard d~ 2
## 3 3              BAC S              R1              En retard d~ 1
## 4 5              BAC technologi~ R1              En retard d~ 1
## 5 1              BAC L              R1              En retard d~ 1
## 6 2              BAC ES             R0              A l'heure o~ 1
## 7 3              BAC S              R1              En retard d~ 1
## 8 3              BAC S              R1              En retard d~ 2
## 9 6              BAC professionn~ R4              Non pris en~ 2
## 10 1             BAC L              R0              A l'heure o~ 2
## # ... with 140 more rows, and 14 more variables: Sexe <fct>, `Id Mention au
## #   Bac` <fct>, `Mention au Bac` <fct>, `Année de cohorte (Année de première
## #   inscription)` <fct>, `Effectif de néobacheliers de la cohorte` <dbl>,
## #   `Passage en 2ème année d'études de santé en 1 an` <dbl>, `Passage en 2ème
```

```
## # année d'études de santé en 2 ans` <dbl>, `Passage en 2ème année d'études de
## # santé en 1 ou 2 ans` <dbl>, `Passage en 2ème année d'études de santé dans
## # la filière Médecine en 1 ou 2 ans` <dbl>, `Passage en 2ème année d'études
## # de santé dans la filière Pharmacie en 1 ou 2 ans` <dbl>, `Passage en 2ème
## # année d'études de santé dans la filière Chirurgie dentaire en 1 ou 2
## # ans` <dbl>, `Passage en 2ème année d'études de santé dans la filière
## # Maïeutique en 1 ou 2 ans` <dbl>, Redoublement <dbl>, `Passage dans d'autres
## # diplômes de santé` <dbl>
```

```
summary(fr_esr_parcours_des_bacheliers_en_paces)
```

```
## Id Série ou type de Bac          Série ou type de Bac Id Âge au bac
## 2:27          BAC ES                :27          R1:47
## 3:36          BAC S                  :36          R2:41
## 5:35          BAC technologique hors STMG:35          R0:51
## 1:17          BAC L                  :17          R4:11
## 6:11          BAC professionnel      :11
## 4:24          BAC STMG               :24
##                Âge au bac Id Sexe    Sexe    Id Mention au Bac
## En retard d'un an          :47    1:67    Homme:67    C:28
## En retard de plus d'un an:41    2:83    Femme:83    P:29
## A l'heure ou en avance    :51
## Non pris en compte        :11
##                               B:21
##                               A:18
##                               Q:22
##                Mention au Bac
## Assez bien                  :28
## Passable deuxième groupe:29
## Passable premier groupe :32
## Bien                        :21
## Très bien                   :18
## Inconnue                    :22
## Année de cohorte (Année de première inscription)
## 2014:150
##
##
##
## Effectif de néobacheliers de la cohorte
## Min.   : 1.00
## 1st Qu.: 3.25
## Median : 10.00
## Mean   : 235.83
## 3rd Qu.: 44.50
## Max.   :5470.00
## Passage en 2ème année d'études de santé en 1 an
## Min.   : 0.00
## 1st Qu.: 0.00
## Median : 0.00
## Mean   : 27.08
## 3rd Qu.: 0.00
## Max.   :1643.00
## Passage en 2ème année d'études de santé en 2 ans
## Min.   : 0.00
```

```

## 1st Qu.: 0.00
## Median : 0.00
## Mean   : 52.37
## 3rd Qu.: 2.00
## Max.   :1912.00
## Passage en 2ème année d'études de santé en 1 ou 2 ans
## Min.   : 0.00
## 1st Qu.: 0.00
## Median : 0.00
## Mean   : 79.35
## 3rd Qu.: 2.00
## Max.   :3146.00
## Passage en 2ème année d'études de santé dans la filière Médecine en 1 ou 2 ans
## Min.   : 0.00
## 1st Qu.: 0.00
## Median : 0.00
## Mean   : 47.47
## 3rd Qu.: 1.00
## Max.   :2395.00
## Passage en 2ème année d'études de santé dans la filière Pharmacie en 1 ou 2 ans
## Min.   : 0.00
## 1st Qu.: 0.00
## Median : 0.00
## Mean   : 18.83
## 3rd Qu.: 1.00
## Max.   :628.00
## Passage en 2ème année d'études de santé dans la filière Chirurgie dentaire en 1 ou 2 ans
## Min.   : 0.000
## 1st Qu.: 0.000
## Median : 0.000
## Mean   : 7.253
## 3rd Qu.: 0.000
## Max.   :224.000
## Passage en 2ème année d'études de santé dans la filière Maïeutique en 1 ou 2 ans
## Min.   : 0.000
## 1st Qu.: 0.000
## Median : 0.000
## Mean   : 5.787
## 3rd Qu.: 0.000
## Max.   :308.000
## Redoublement      Passage dans d'autres diplômes de santé
## Min.   : 0.0      Min.   : 0.000
## 1st Qu.: 1.0      1st Qu.: 0.000
## Median : 3.0      Median : 0.000
## Mean   : 111.1    Mean   : 5.013
## 3rd Qu.: 13.5     3rd Qu.: 0.750
## Max.   :3295.0    Max.   :167.000

```

Noter que les résultats de la fonction `summary` sont maintenant beaucoup plus informatifs notamment concernant les variables catégorielles.

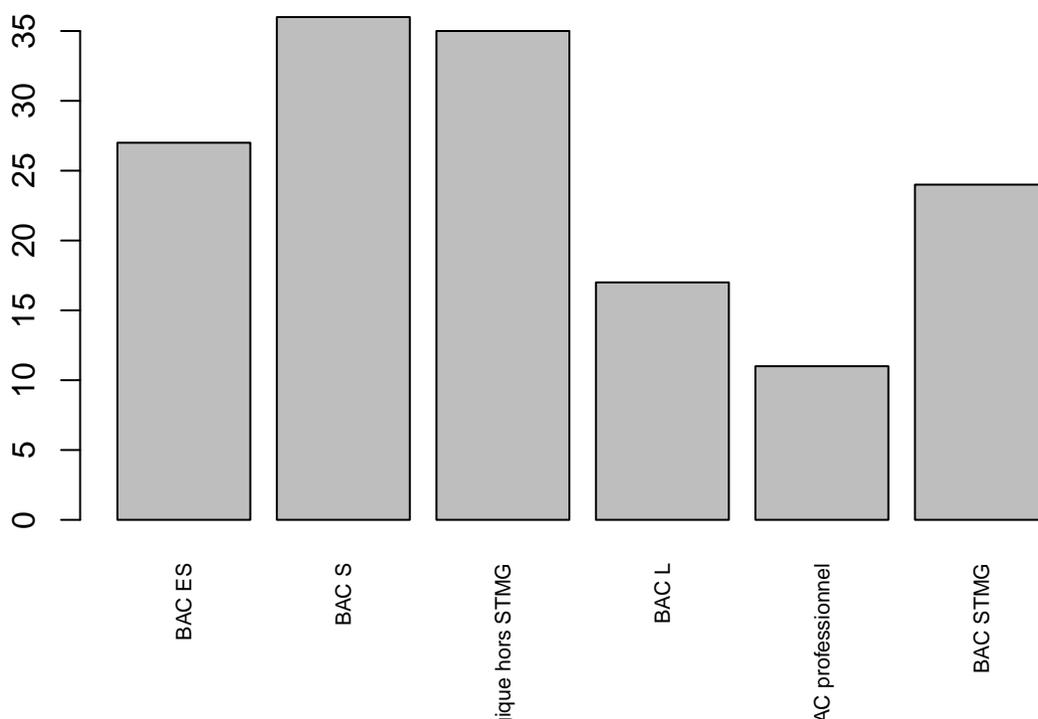
Quelques analyses descriptives

Représentations graphiques

```
Effectif_Serie <- table(fr_esr_parours_des_bacheliers_en_paces$`Série ou type de Bac`)  
Effectif_Serie
```

```
##  
##                BAC ES                BAC S  
##                27                36  
## BAC technologique hors STMG                BAC L  
##                35                17  
##                BAC professionnel                BAC STMG  
##                11                24
```

```
barplot(Effectif_Serie, las=3, cex.names=0.65)
```



```
pie(Effectif_Serie, las=3, cex.names=0.65)
```

```
## Warning in text.default(1.1 * P$x, 1.1 * P$y, labels[i], xpd = TRUE, adj =  
## ifelse(P$x < : "cex.names" n'est pas un paramètre graphique
```

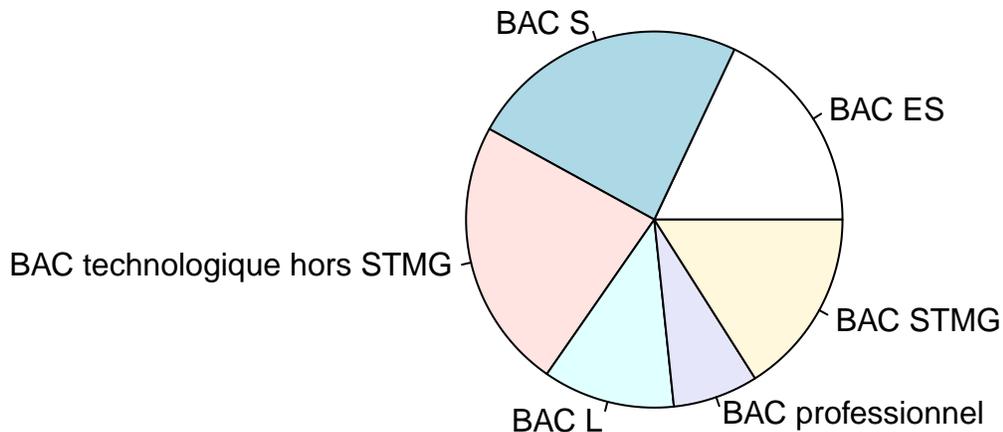
```
## Warning in text.default(1.1 * P$x, 1.1 * P$y, labels[i], xpd = TRUE, adj =  
## ifelse(P$x < : "cex.names" n'est pas un paramètre graphique
```

```
## Warning in text.default(1.1 * P$x, 1.1 * P$y, labels[i], xpd = TRUE, adj =  
## ifelse(P$x < : "cex.names" n'est pas un paramètre graphique
```

```
## Warning in text.default(1.1 * P$x, 1.1 * P$y, labels[i], xpd = TRUE, adj =  
## ifelse(P$x < : "cex.names" n'est pas un paramètre graphique
```

```
## Warning in text.default(1.1 * P$x, 1.1 * P$y, labels[i], xpd = TRUE, adj =  
## ifelse(P$x < : "cex.names" n'est pas un paramètre graphique
```

```
## Warning in text.default(1.1 * P$x, 1.1 * P$y, labels[i], xpd = TRUE, adj =
## ifelse(P$x < : "cex.names" n'est pas un paramètre graphique
## Warning in title(main = main, ...): "cex.names" n'est pas un paramètre graphique
```



Tests statistiques

On peut par exemple tester l'égalité des proportions dans les séries selon le sexe.

```
Effectif_Serie_Sexe <- table(fr_esr_parcours_des_bacheliers_en_paces$`Série ou type de Bac`, fr_esr_parcours_des_bacheliers_en_paces$Sexe)
Effectif_Serie_Sexe
```

```
##
##
##      Homme Femme
## BAC ES      11   16
## BAC S       18   18
## BAC technologique hors STMG 17   18
## BAC L        5   12
## BAC professionnel    5    6
## BAC STMG     11   13
```

```
chisq.test(Effectif_Serie_Sexe)
```

```
## Warning in chisq.test(Effectif_Serie_Sexe): Chi-squared approximation may be
## incorrect
```

```
##
## Pearson's Chi-squared test
##
## data:  Effectif_Serie_Sexe
## X-squared = 2.4152, df = 5, p-value = 0.7892
```

```
fisher.test(Effectif_Serie_Sexe)
```

```
##
## Fisher's Exact Test for Count Data
##
## data:  Effectif_Serie_Sexe
## p-value = 0.797
## alternative hypothesis: two.sided
```

Conclusion

Il va de soi que le présent document ne présente qu'une infime partie des analyses statistiques qu'il est possible de réaliser avec R. Mais les principes de base étant maintenant présentés, il n'y a plus qu'à se lancer...