

A photograph of a modern building facade. The left side shows a white brick wall, while the right side features a large glass curtain wall. A prominent dark blue diagonal beam crosses the upper part of the image. The sky is blue with a few white clouds. In the foreground, there are green plants and a glass-enclosed walkway.

Entrées et sorties en R

Vincent Guillemot
Amaury Vaysse

Institut Pasteur

MICS

Les formats de fichiers de données



peut lire différents formats de données. Les principaux formats rencontrés sont:

- Fichiers textes à valeurs séparées par un caractère spécial
- Fichiers Excel
- Fichier natif de R (.Rda)

Un utilisateur peut vouloir enregistrer principalement des:

- données transformées dans R
- résultats d'analyses sous forme de tables
- sorties textes plus complexes
- graphiques.

Les exemples...

... se trouvent dans un [dossier compressé](#) qui contient les fichiers suivants :

Nom du fichier	Taille
data_io/1.tabular_simple.tsv	4486
data_io/2.tabular_complex_header.tsv	4608
data_io/3.only_one_col	487
data_io/4.tabular_no_header.tsv	2012
data_io/5.data_in_excel.xlsx	17858

Tables dans un fichier texte

la plus basique: fichier texte avec un séparateur

exemple:

nom	groupe	Energie	Eau	Proteines	Glucides	Lipides		
Abricot	crus	194	87.1	0.81	9.01	0.25	6.7	1.3	1.7
Abricot	secs	1010	24.7	2.88	59.1	0.5	34.3	10.6	8.3
Ananas	exotique		304	81.3	0.94	15.1	0.25	14.9	2.8

lecture/écriture avec les fonctions de bases de R

```
fruits <- read.table(file = "data_io/1.tabular_simple.tsv", header = T, sep = "\t")
write.table(x = fruits, file = "1.simpletabular.tsv", quote = F, sep = "\t", row.names = F)
```

Attention à ne pas écraser un fichier

» Il existe plusieurs variantes selon les fichiers de données

avec plusieurs ligne d'en-tête

exemple:

```
# nom : nom du fruit
# groupe : fruit sec, exotique, mangé crus ou en compote
# autres colonnes : valeurs nutritionnelles
nom      groupe  Energie Eau      Proteines      Glucides      Lipides ....
Abricot  crus      194      87.1    0.81    9.01    0.25    6.7    1.3    1.7
Abricot  secs      1010     24.7    2.88    59.1    0.5     34.3   10.6   8.3
Ananas   exotique      304      81.3    0.94    15.1    0.25    14.9   2.8
```

```
fruits <- read.table(file = "2.tabular_complex_header.tsv", header = T, sep = "\t", skip=3)
```

avec une seule colonne

exemple:

```
Abricot  
Ananas  
Banane  
Canneberge  
Cerise  
Citron  
Clementine  
CompoteMultiFruits
```

```
fruit_names <- scan(file = "3.only_one_col", what=character())
```

sans en-tête

Abricot crus	194	87.1	0.81	9.01	0.25	
Abricot secs	1010	24.7	2.88	59.1	0.5	
Ananas exotique		304	81.3	0.94	15.1	0.25

```
fruits <- read.table(  
  file = "tabular_noheader.tsv",  
  col.names = c("nom", "groupe", "Energie", "Eau", "Proteines", "Glucides", "Lipides"),  
  sep = "\t"  
)
```

OU

```
fruits <- read.table(file = "4.tabular_no_header.tsv", , header = T, sep = "\t")  
names(fruits) <- c("nom", "groupe", "Energie", "Eau", "Proteines", "Glucides", "Lipides")
```

Fichier binaire de R

Les fichiers utilisés pour pouvoir sauvegarder et reprendre une session R plus tard.

```
load("data_io/fruits.RData")
```

cela crée dans la session un ou des objets tels qu'ils ont été sauvegardés.

pour sauvegarder l'objet fruits:

```
save(fruits, file = "mySave.Rdata")
```

Pour sauvegarder toute une session de travail:

```
save.image(file = "mySession.RData")
```

ou

```
q("yes")
```

Lecture de fichier excel

Les données peuvent aussi être transmises sous forme d'un classeur excel. Il faut alors extraire la table d'intérêt dans la feuille qui le contient.

On utilise pour cela le package `readxl` qui fournis (entre autres) les fonctions

- `read_excel`
- `read_xls`
- et `read_xlsx`

Par exemple le classeur `data_in_excel` contient 3 feuilles:

```
library(readxl)
read_xlsx(path = "data_io/5.data_in_excel.xlsx", sheet = "fruits", skip = 3)
```

Écriture de fichiers Excel

Pour écrire un fichier sous format “Excel”, on utilise pour cela le package `writexl`.

Par exemple, on enregistre le tableau `fruits_summary` dans un fichier `fruits_summary.xlsx` avec la suite de commandes suivantes :

```
library(dplyr)
fruits_summary <- fruits %>%
  group_by(groupe) %>%
  summarize(
    MoyenneSucres = mean(Sucres),
    MoyenneLipides = mean(Lipides))

library(writexl)
write_xlsx(fruits_summary, path = "fruits_summary.xlsx")
```

Sorties graphiques

Quand nous voulons sauvegarder un graphe en ligne de commande, nous avons 2 moyens principaux:

Si le graphe a été généré avec les fonctions de base de R:

R fournit des fonctions pour enregistrer sous différents formats de fichier image (BMP, JPEG, PNG & TIFF). Elles fonctionnent toutes dans le même scénario:

```
png(filename="monBeauGraph.png", width = 960, height = 960 ) # remplacer l'affichage
# Code pour generer le graphique
dev.off() # finaliser le fichier
```

Attention: suivant les options "width" et "height", les textes et les points ne prendront pas la même place dans le fichier qu'à l'écran.

Si le graphe a été généré avec les fonctions de ggplot:

ggplot fournit la fonction ggsave . Par exemple, pour sauvegarder un plot enregistré dans la variable "g":

```
ggsave( filename="monBeauGraph.png", plot=g, device = png)
```