



# Programmer avec le logiciel R – Cours 2

Gaëlle LELANDAIS



# Importation et exportation de données

# Rappel : gérer son répertoire de travail

3

- ◉ Dans quel répertoire je travaille ?
  - > `getwd()`
- ◉ Changer de répertoire
  - > `setwd()`

```
> getwd()
[1] "/Users/gaellelelandais/Enseignements/Seance2"

> setwd("../Seance3")

> getwd()
[1] "/Users/gaellelelandais/Enseignements/Seance3"
```

# Lecture d'un fichier texte

4

- Objectif

- > Importer dans R un ensemble de données écrites dans un fichier texte

- Fonctions disponibles

- > `scan()`
  - > **`read.table()`**, **`read.csv()`**
  - > `readLines()`

Attention au caractère de séparation



	Col 1	Col 2	...	Col n
Ligne 1	Val 1	Val 2	...	Val n
Ligne 2	Val 1	Val 2		Val n
...				
Ligne n	Val 1	Val 2		Val n

# Fonction « read.table() »

5

Le nom du fichier texte  
est « FichierALire.txt »

Les colonnes sont  
séparées par des  
tabulations

	col1	col2	col3
Line1	2	8	5
Line2	1	6	3

```
> read.table("FichierALire.txt")
      col1 col2 col3
Line1    2    8    5
Line2    1    6    3
```

# Ecriture d'un fichier texte

6

## Objectif

- > Ecrire un ensemble de données obtenues avec le logiciel R dans un fichier texte
- > Ce fichier pourra être lu par un autre logiciel (Excel ou OpenOffice par exemple)

## Fonctions disponibles

- > `cat()`
- > `write()`
- > **`write.table()`**

Caractère de séparation ↓

	Col 1	Col 2	...	Col n
Ligne 1	Val 1	Val 2	...	Val n
Ligne 2	Val 1	Val 2		Val n
...				
Ligne n	Val 1	Val 2		Val n

# La fonction « write.table() »

7

## ◉ Exemple

- Choisir au hasard un ensemble de 10 valeurs numériques selon distribution normale, puis sauver le résultat dans un fichier

```
> myData = rnorm(10)
> write.table(myData, file = "FichierDeSortie.txt")
> write.table(myData, file = "FichierDeSortie2.txt", row.names
= F, col.names = F)
```

```
"x"
"1" 1.25043233021441
"2" 1.4820228504417
"3" -1.54421011445287
"4" 1.05130132225392
"5" 0.119163137729908
"6" 0.931434307605138
"7" -1.48901345258875
"8" 0.769482947989051
"9" 0.327828096338627
"10" -0.582002255213116
```

```
1.25043233021441
1.4820228504417
-1.54421011445287
1.05130132225392
0.119163137729908
0.931434307605138
-1.48901345258875
0.769482947989051
0.327828096338627
-0.582002255213116
```

# Manipulation des objets



# Vecteurs

9

## ⊙ Définition

- > Succession d'éléments (ou informations) de même type (nombres entiers ou décimaux, lettres de l'alphabet, mots, etc.)

```
> Vect = c(1, 4, 5, 6, 57)
> Vect
[1] 1 4 5 6 57
> Vect2 = c('a', 'k', 'm', 'p')
> Vect2
[1] 'a' 'k' 'm' 'p'
```

## ⊙ Valeurs particulières

- > NA : Valeur manquante (*Not Available*)
- > NaN : Pas de nombre (*Not a Number*)
- > -Inf/Inf : Symbole infini (+ ou -)

# Vecteurs

10

## ◉ Exemple

```
> vect = 12:28  
[1] 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
```

## ◉ Accéder à un élément d'un vecteur

- > Symbole « [ ] »

```
> vect[2]      # 2ème élément du vecteur  
[1] 13
```

# Vecteurs

11

- Accéder à plusieurs éléments d'un vecteur

- > Consécutifs

```
> vect[c(5, 6, 7, 8, 9)]  
[1] 16 17 18 19 20
```

```
> vect[5:9]  
[1] 16 17 18 19 20
```

- > Non consécutifs

```
> vect[c(5, 10, 13)]  
[1] 16 21 24
```

- Supprimer un ou plusieurs éléments d'un vecteur

```
> vect[-1]  
[1] 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28  
> vect[-5:-9]  
[1] 12 13 14 15 21 22 23 24 25 26 27 28
```

# Etiquetage des éléments d'un vecteur

12

## ⊙ Principe

- > Donner un nom explicite aux éléments d'un vecteur
- > Possibilité de suivre les éléments au fur et à mesure des manipulations du vecteur

## ⊙ Fonction

- > *names()*

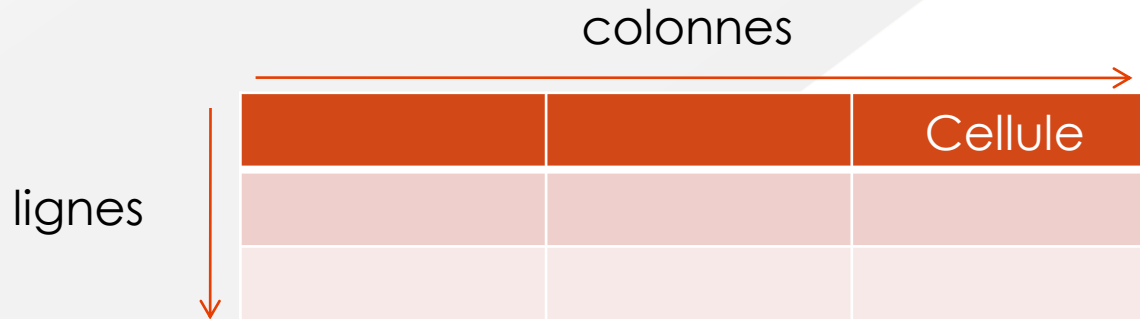
```
> notes = c(12,15,8,9,11,15,5,20,17)
> names(notes) = c("Villon", "Polin", "Exfi", "Rotaf", "Zerif",
"Gared", "Neyres", "Ropert", "Saïdil")
> notes
Villon Polin Exfi Rotaf Zerif Gared Neyres Ropert Saïdil
    12    15    8     9    11    15     5    20    17
> notes[c("Exfi", "Gared")]
Exfi Gared
   8   15
```

# Tableaux : matrices et *data frame*

13

- Définition

- > Ensemble d'éléments regroupés en deux dimensions



- Matrice

- > Un seul type d'élément pour toutes les cellules

- *Data frame*

- > Les colonnes peuvent être de types différents (valeurs numériques, chaînes de caractères, etc.)

# Utilisation des tableaux

14

- ◉ Fonctions de création d'un tableau
  - > *matrix()*, *data.frame()*, *cbind()*, *rbind()*, etc.
- ◉ Fonctions de manipulation d'un tableau
  - > *data.class()*
  - > *dim()*
  - > etc.

```
> matrix(0,nrow=2,ncol=2)
      [,1] [,2]
[1,]    0    0
[2,]    0    0
> dim(matrix(0,nrow=2,ncol=2))
[1] 2 2
```

```
> cbind(c(1,2),c(3,4))
      [,1] [,2]
[1,]    1    3
[2,]    2    4
> rbind(c(1,2),c(3,4))
      [,1] [,2]
[1,]    1    2
[2,]    3    4
```

# Tableaux

15

- Exemple :

```
> Mat = cbind(1:4, 5:8)
  [,1] [,2]
[1,]  1  5
[2,]  2  6
[3,]  3  7
[4,]  4  8
```

- Accéder aux éléments du tableau

- Symbol « [ligne, colonne] »

```
> Mat[3, 2]           #élément de la 3ème ligne et 2ème colonne
[1] 7
> Mat[1, ]            #éléments de la première ligne
[1] 1 5
> Mat[c(1,3),]        #éléments des lignes 1 et 3
  [,1] [,2]
[1,]  1  5
[2,]  3  7
> Mat[,1]             #éléments de la première colonne
```

# Etiquetage des éléments d'un tableau

16

## ⦿ Principe

- Donner un nom explicite aux lignes et aux colonnes d'un tableau

## ⦿ Fonctions

- `row.names()`, `colnames()`

```
> Mat1
      [,1]      [,2]
[1,] -0.1177814 -0.7376553
[2,] -1.1422671 -0.4758635
> row.names(Mat1) = c("Ligne1", "Ligne2")
> colnames(Mat1) = c("Colonne1", "Colonne2")
> Mat1
      Colonne1  Colonne2
Ligne1 -0.1177814 -0.7376553
Ligne2 -1.1422671 -0.4758635
```



# Test sur un vecteur ou un tableau

17

- Identifier les éléments d'un vecteur qui vérifient une condition
- Les positions peuvent également être obtenues
  - > `which(cond)`

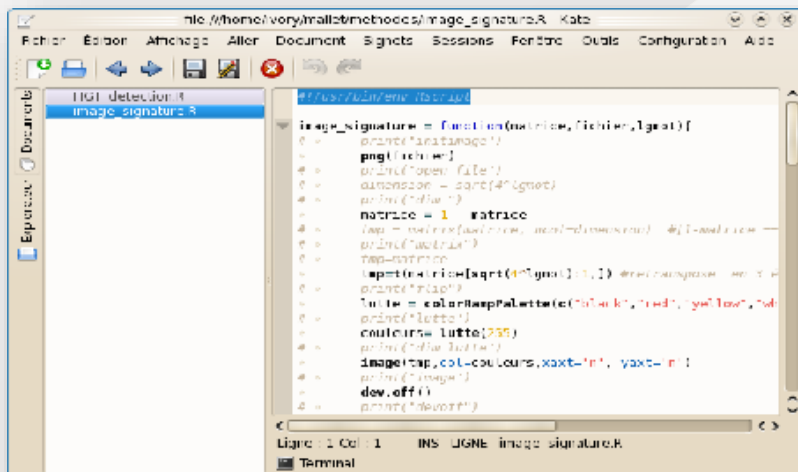
```
> vect = c(12,15,8,15,9,5,11,17,19,5,15,12,8)
> vect <= 10
[1] FALSE FALSE TRUE FALSE TRUE TRUE FALSE FALSE FALSE
TRUE FALSE FALSE TRUE

> vect = 100:110
> vect
[1] 100 101 102 103 104 105 106 107 108 109 110
> which(vect >= 105)
[1] 6 7 8 9 10 11
```

# Sauvegarde des commandes

18

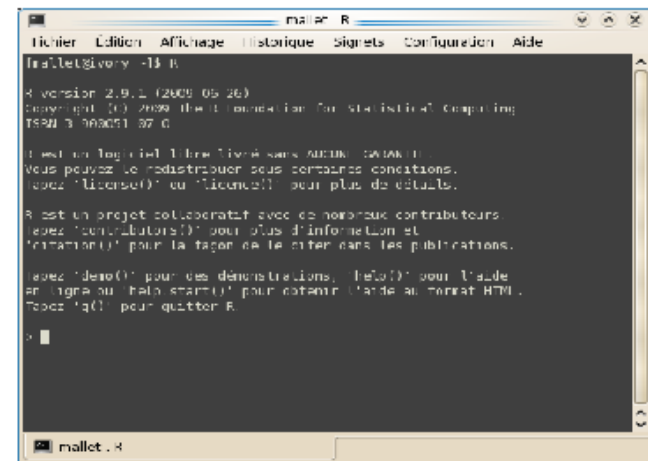
- Choisir un logiciel d'édition de texte
  - Par exemple : *kate*, *kwrite*, ***gedit***, *nedit*, *vim*, *emacs*, etc.
- Copier les commandes écrites dans l'éditeur de texte puis les coller dans la console R
  - Ne pas oublier de sauvegarder le fichier texte



The screenshot shows the Kate text editor with the file `image_signature.R` open. The script defines a function `image_signature` that takes a matrix and a file name as input. It generates a color palette, creates a plot, and saves it as a PNG file. The script is as follows:

```
#Fichier image_signature.R

image_signature = function(matrice,fichier,lgret){
  #
  png(fichier)
  plot(matrice)
  dimensions = sort(dim(matrice))
  print(dimensions)
  matrice = 1:matrice
  cap = matrix(matrice, nrow=dimensions[1], ncol=dimensions[2])
  print(dim(cap))
  row=matrix(1:lgret, nrow=lgret, ncol=1)
  print(dim(row))
  matrice = colorRampPalette(c("black","red","yellow","blue"))(255)
  couleurs = matrice
  image(cap,col=couleurs,xaxt="n", yaxt="n")
  print(fichier)
  dev.off()
  print("devoff")
}
```



The screenshot shows the R console window with the R startup message. The message includes the R version (2.9.1), the date (2009-05-26), and the location (Lyon). It also mentions the R project and the R Foundation. The message is as follows:

```
R version 2.9.1 (2009-05-26)
Copyright (C) 2009 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R est un logiciel libre livré sans AUCUNE GARANTIE.
Vous pouvez le redistribuer sous certaines conditions.
Voyez 'licence()' ou 'licence()' pour plus de détails.

R est un projet collaboratif avec de nombreux contributeurs.
Voyez 'contributeurs()' pour plus d'information et
'citation()' pour la façon de le citer dans les publications.

Voyez 'demo()' pour des démonstrations, 'help()' pour l'aide
en ligne ou 'help.start()' pour obtenir l'aide au format HTML.
Touchez 'q()' pour quitter R.
```

# Exécuter les commandes d'un fichier

19

- Les commandes R écrites dans un fichier texte peuvent être exécutées successivement
  - `source(« MonFichier.R »)`
- Certaines lignes peuvent être ignorées
  - Ajout du symbole « # » en début de ligne

Fichier texte (nommé « ScriptR.R ») :

```
# Exemple de script R
print("c'est un test")

# calcul de la valeur du
# cosinus de 90
a = cos(90)
print(a)
```

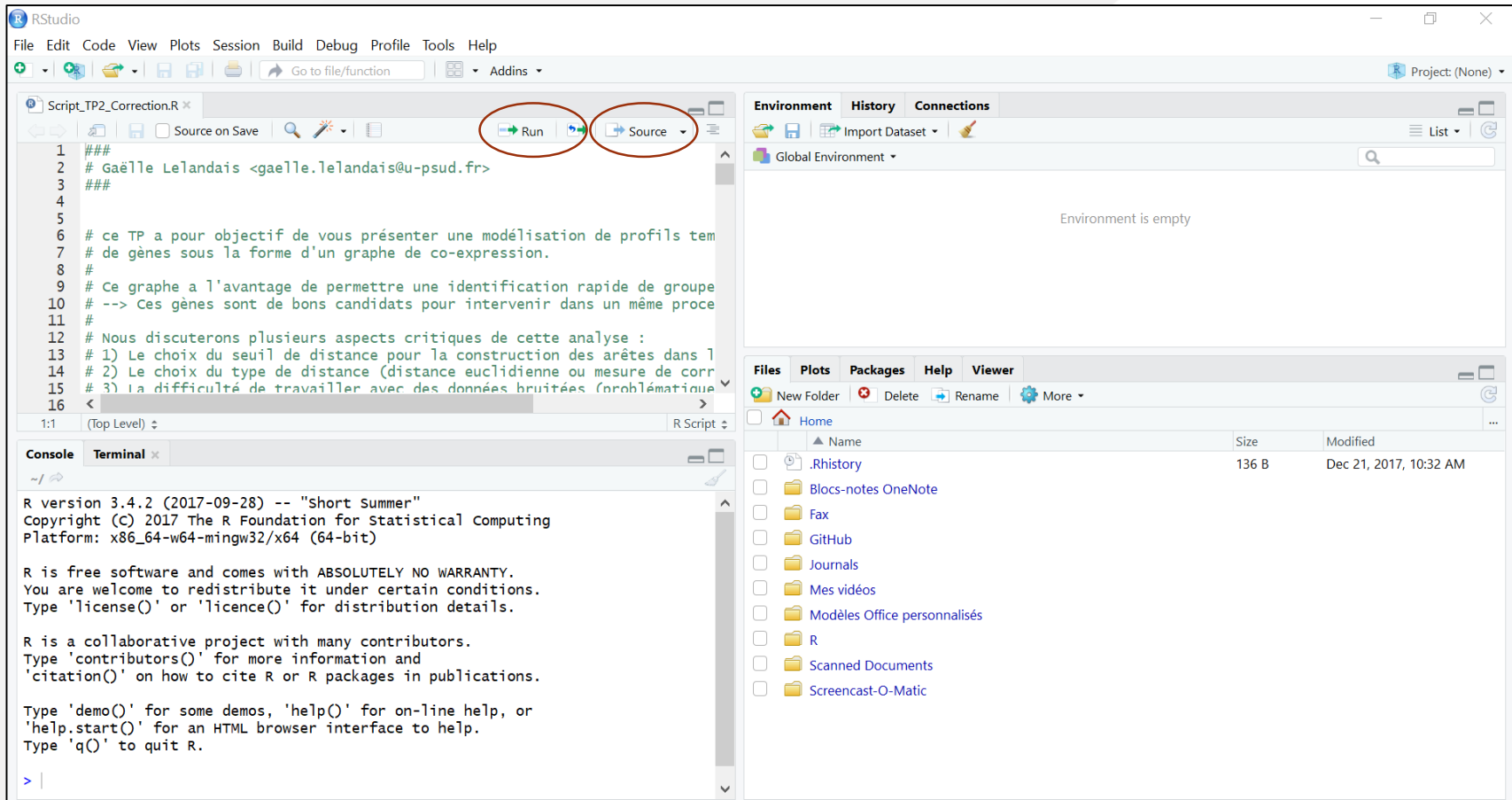
Logiciel R :

```
> source("ScriptR.R")
[1] "c'est un test"
[1] -0.4480736
```

# Le logiciel RStudio

20

<https://rstudio.com/>



# ⊙ Séance d'exercices