

Introduction à R

Thomas Denecker & Steven Volant

2022-11-15

Contents

1	Présentation du cours	7
1.1	A propos de du livre	7
1.2	Demandez le programme	7
1.3	Intervenants	7
2	R en quelques mots	9
2.1	Pourquoi ?	9
2.2	Comment l’avoir ?	9
2.3	Sur quel OS ?	9
2.4	Historique	9
2.5	R vs Excel	10
2.6	Avantages et inconvénients	11
2.7	Geeks and repetitive tasks	12
2.8	R sait tout faire	12
3	Comment utiliser R ?	15
3.1	Modes d’utilisation (liste non exhaustive)	15
3.2	Ouverture ou connexion à RStudio	15
3.3	RStudio	16
4	Premiers pas	19
4.1	R sait tout faire : il compte !	19
4.2	Notion de variable/objet	20

5 Import de données	23
5.1 Version “Avec les boutons”	23
5.2 The “R geek” way (V2, directement depuis Rstudio)	27
5.3 The “bash geek” way (V3, directement de votre home du cluster)	30
5.4 Actualisation du dossier	33
6 Lecture des données	35
6.1 Chargement des données (dans la mémoire de R)	35
6.2 Affichage de l’objet “exprs”	36
6.3 Caractéristiques d’un tableau de données	39
7 Manipuler les données dans R	43
7.1 Sélection de colonnes d’un tableau	43
7.2 Sélection de lignes d’un tableau	46
7.3 formulation plus intuitive	47
8 Visualisation des données	49
8.1 Histogrammes	49
8.2 Boîtes à moustaches (boxplots)	52
8.3 Nuage de points	58
9 Analyse d’expression différentielle : MA-plot	61
9.1 C’est quoi un MA plot	61
9.2 Calculs sur les colonnes	62
9.3 MA-plot : log2FC vs intensité	64
9.4 Appliquer une fonction sur les lignes/colonnes	65
10 Intégration des données	71
10.1 Charger les annotations des gènes	71
10.2 Combien ?	75
10.3 Ma première bioinformatique intégrative	76
10.4 Visualisation	77

<i>CONTENTS</i>	5
11 Bonus	79
11.1 R de base	79
11.2 Plotly	81
11.3 echarts	82
12 Conclusion	83
12.1 Take home messages	83
12.2 Ressources IFB	83
12.3 Resource	84

Chapter 1

Présentation du cours

Bienvenues dans le cour Introduction à R de l'EBAIL ! Pour accompagner ce cours, Thomas Denecker et Steven Volant vous proposent ce livre. C'est une grande première alors n'hésitez pas à nous faire des retours.

1.1 A propos de du livre

L'objectif de ce livre est d'accompagner les apprenants de l'école EBAIL.

1.2 Demandez le programme

Debut	Fin	Durée	Lieu
8:30	10:15	01:45	HDF

1.3 Intervenants

- Thomas Denecker – thomas.denecker@france-bioinformatique.fr
- Steven Volant - steven.volant@pasteur.fr

La version “slides” a été créée initialement par Hugo Varet – hugo.varet@pasteur.fr

Chapter 2

R en quelques mots

2.1 Pourquoi ?

Langage de programmation qui permet de : - manipuler des données : importer, transformer, exporter faire des analyses statistiques plus ou moins complexes : description, exploration, modélisation... - créer des (jolies) figures

2.2 Comment l'avoir ?

Disponible sur RCRAN

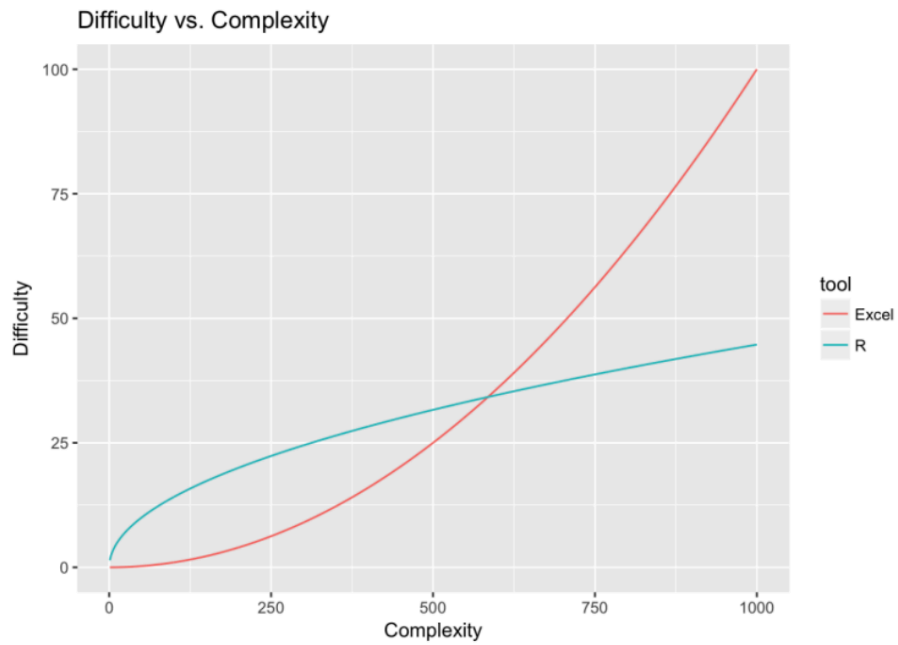
2.3 Sur quel OS ?

Tous !

2.4 Historique

- 1993 : Début du projet R
- 2000 : sortie de R 1.0.0
- 2022 : R 4.2.2

2.5 R vs Excel



Source: R-bloggers

2.5.1 Pourquoi plus Excel ?

Un exemple parmi tant d'autres !

Covid : le Royaume-Uni passe à côté de milliers de cas à cause... d'un fichier Excel arrivé à saturation

Les autorités sanitaires britanniques ont reconnu que près de 16.000 cas de coronavirus en Angleterre sont passés sous le radar au cours de la semaine écoulée à cause d'un problème dans le chargement des données.

[Lire plus tard](#) [Europe](#) [Partager](#) [Commenter](#)



Source Alexandre Counis, Les Echos, 5 oct. 2020

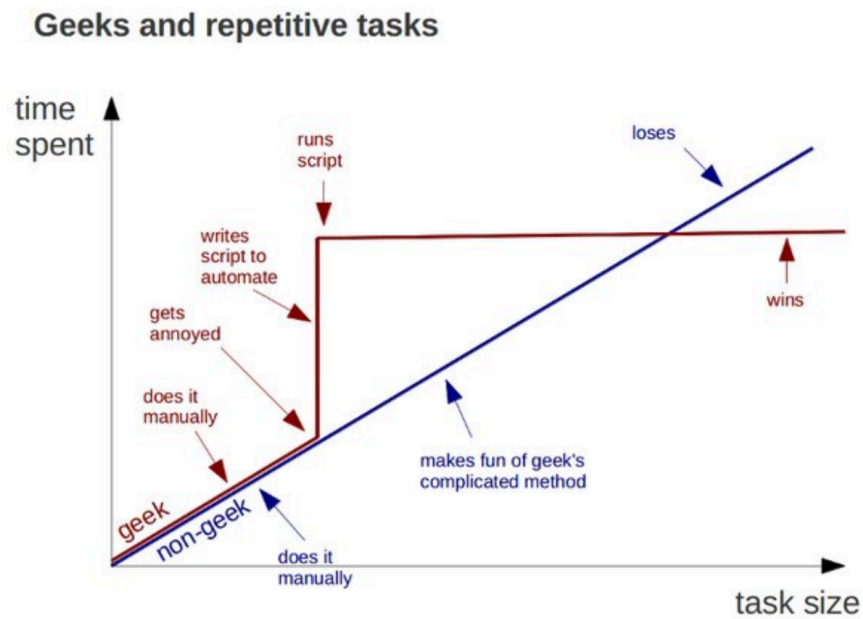
2.6 Avantages et inconvénients

2.6.1 Avantages

- Souplesse d'utilisation pour réaliser des analyses statistiques
- Libre et gratuit, même s'il existe maintenant des versions payantes de RStudio (shiny et/ou server)
- Reproductibilité des analyses en écrivant/sauvegardant les commandes R dans des scripts
- Large communauté d'utilisateurs/aide en ligne
- Grand nombre de packages spécifiques

2.6.2 Inconvénients

2.7 Geeks and repetitive tasks



2.8 R sait tout faire

Lire un tableau de données

```
read.table()
```

Fusionner deux tableaux

```
merge()
```

Filtrer des lignes

```
data[data$x > 10]
```

Sélectionner des colonnes

```
data[,c("x","y")]
```

Rechercher une chaîne de caractères

```
grep()
```

Réaliser une ACP

```
prcomp()
```

Calculer une moyenne

```
mean()
```

Additionner deux matrices

```
mat1 + mat2
```

Exporter un tableau de données

```
write.table()
```

Calculer une variance

```
var()
```

Régression linéaire

```
lm()
```

Tracer une courbe

```
plot()
```

Tester une hypothèse

```
t.test()
```

Dessiner un histogramme

```
hist()
```

Convertir des données

```
as.matrix()
```

Chapter 3

Comment utiliser R ?

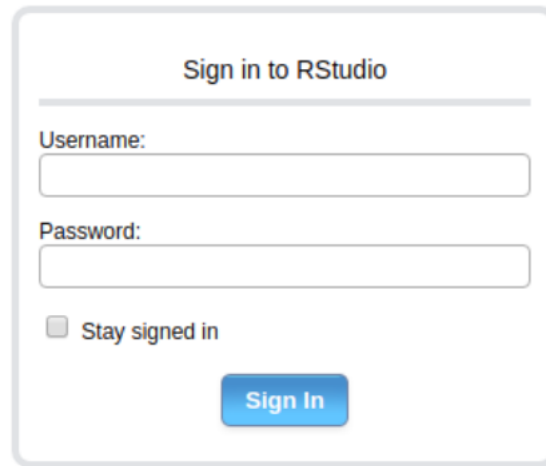
3.1 Modes d'utilisation (liste non exhaustive)

- Localement via le terminal
- Localement via RStudio (utilisation classique)
- Sur un serveur via le terminal et une connexion ssh
- Sur un serveur via un navigateur web pour accéder à RStudio server
- Sur un serveur via un navigateur web pour accéder à RStudio server par Jupyter

3.2 Ouverture ou connexion à RStudio

3 alternatives :

1. Ouvrir RStudio sur votre propre ordinateur (si installé)
2. Vous connecter au serveur Web RStudio de l'IFB <https://rstudio.cluster.france-bioinformatique.fr> puis vous identifier



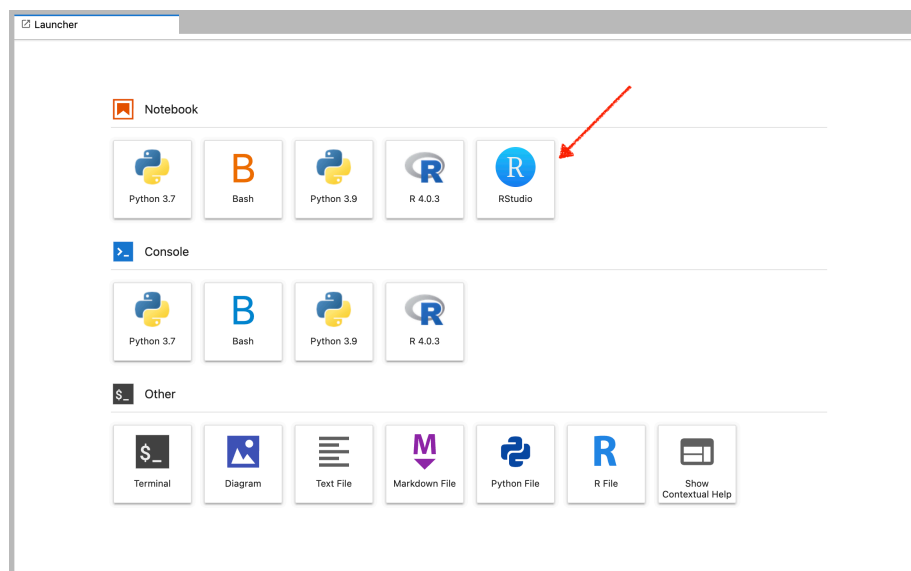
Sign in to RStudio

Username:

Password:

☐ Stay signed in

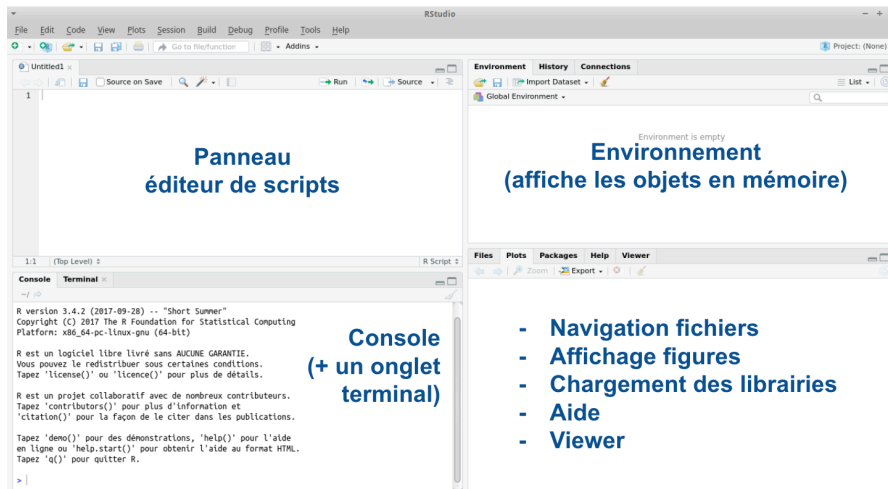
3. Vous connecter via Jupyter lab de l'IFB <https://jupyterhub.cluster.france-bioinformatique.fr> puis cliquer sur l'icône RStudio



3.3 RStudio

- Disponible depuis 2011
- Logiciel facilitant l'utilisation de R via 4 panneaux

- Chaque panneau présente plusieurs onglets (fonctionnalités complémentaires)



Chapter 4

Premiers pas

4.1 R sait tout faire : il compte !

Tapez les commandes suivantes dans le panneau Console de RStudio

```
2 + 3
```

```
## [1] 5
```

```
4 * 5
```

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

```
6 / 4
```

```
## [1] 1.5
```

```
1:10
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

```
8:-9
```

```
## [1] 8 7 6 5 4 3 2 1 0 -1 -2 -3 -4 -5 -6 -7 -8 -9
```

```
1,2
```

```
1.2
```

```
## [1] 1.2
```

4.2 Notion de variable/objet

Créer une variable nommée a et lui assigner une valeur

```
a <- 2
```

Afficher la valeur de la variable a

```
print(a)
```

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

Même résultat: si on évoque le nom de variable, R l'imprime

```
a
```

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

Assigner une valeur à une seconde variable

```
b <- 3
```

Effectuer un calcul avec 2 variables

```
a_plus_b <- a + b
```

Afficher le contenu de la variable a_plus_b

```
print(a_plus_b)
```

```
## [1] 5
```

Changer la valeur de a

```
a <- 7
```

Note: le contenu de a_plus_b n'est pas modifié

```
print(a_plus_b)
```

```
## [1] 5
```

On recalcule a_plus_b

```
a_plus_b <- a + b
```

La nouvelle valeur tient compte de la modification de a

```
print(a_plus_b)
```

```
## [1] 10
```

Créer un vecteur

```
vec1 <- c(1,10)
```

Créer un vecteur contenant une séquence d'entiers de 1 à 10

```
vec2 <- 1:10
```

Somme d'un vecteur et d'un nombre

```
vec2 + a
```

```
## [1] 8 9 10 11 12 13 14 15 16 17
```

Vecteur de chaînes de caractères

```
vec3 <- c("riri", "fifi", "loulou")
```

Diviser un vecteur de nombres par un nombre

```
vec2 / 2
```

```
## [1] 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
```

Diviser des chaînes de caractères par un nombre

vec3 / 2

Attention : Noms de variables interdits: TRUE, FALSE, T, F, c, t, pi, data, LETTERS, letters, ...

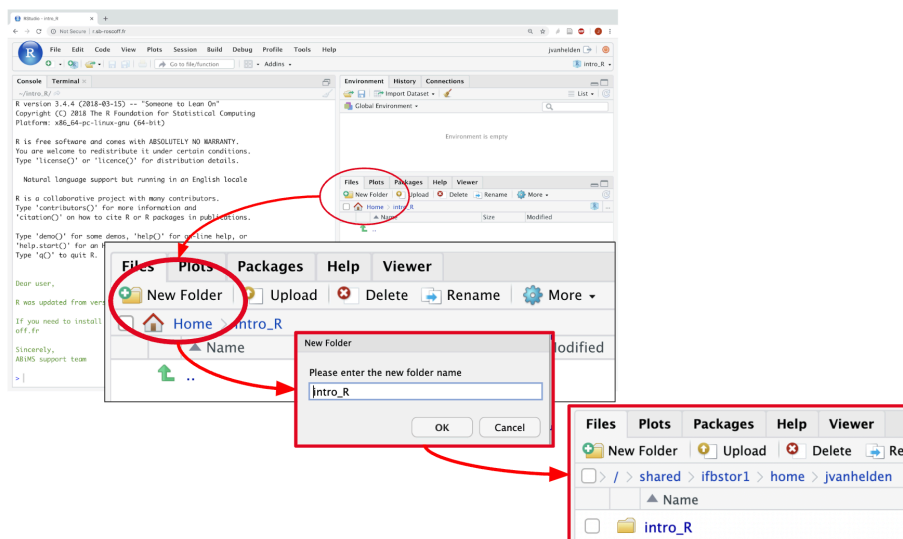
Chapter 5

Import de données

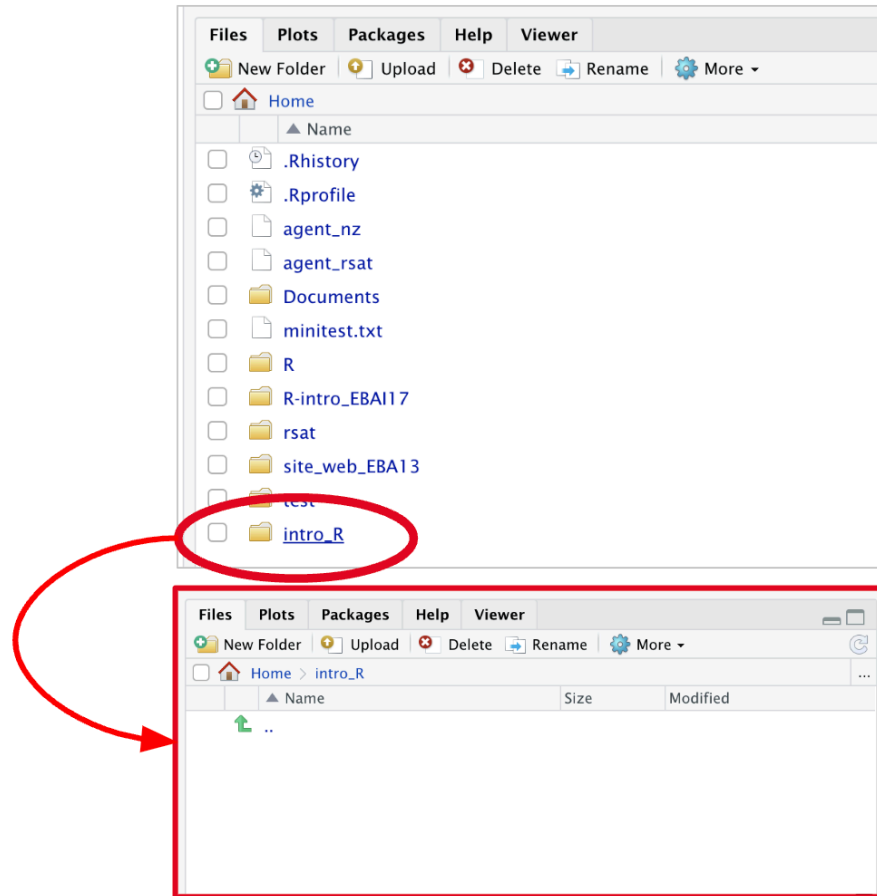
5.1 Version “Avec les boutons”

5.1.1 Création d'un dossier intro_R pour vos résultats de ce TP

Attention Dans votre espace projet ou votre home.

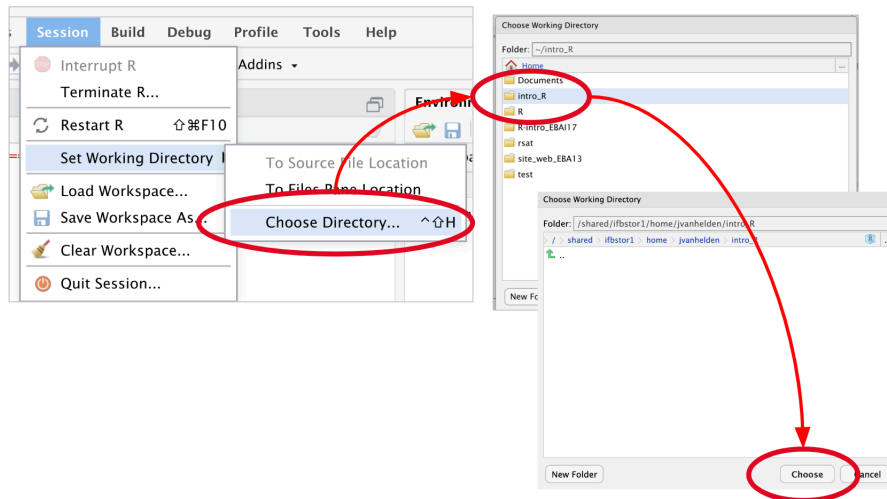


5.1.2 Déplacement dans le dossier “intro_R”



5.1.3 Définissez votre dossier espace de travail (working directory)

1. Dans le menu “Session”, lancez “Choose Directory ...”
2. Naviguez jusqu’à votre dossier `intro_R`
3. Double-cliquez dessus pour l’ouvrir
4. Cliquez Choose



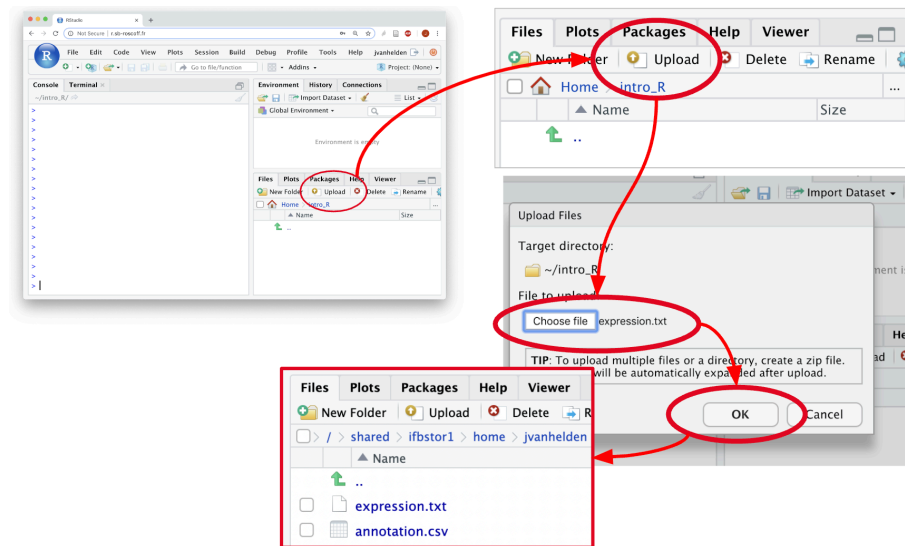
5.1.4 Téléchargez les fichiers sur votre machine

A partir d'un navigateur Web, téléchargez et enregistrez sur votre ordi les fichiers de données - expression.txt: données d'expressions pour 4 échantillons - annotation.csv: informations sur les gènes (id, name, chr, start, stop)

Attention: veillez à sauvegarder les fichiers - sous leur nom original, - avec les extensions .txt et .csv respectives (certains navigateurs omettent l'extension, ce qui poserait problème pour la suite du TP)

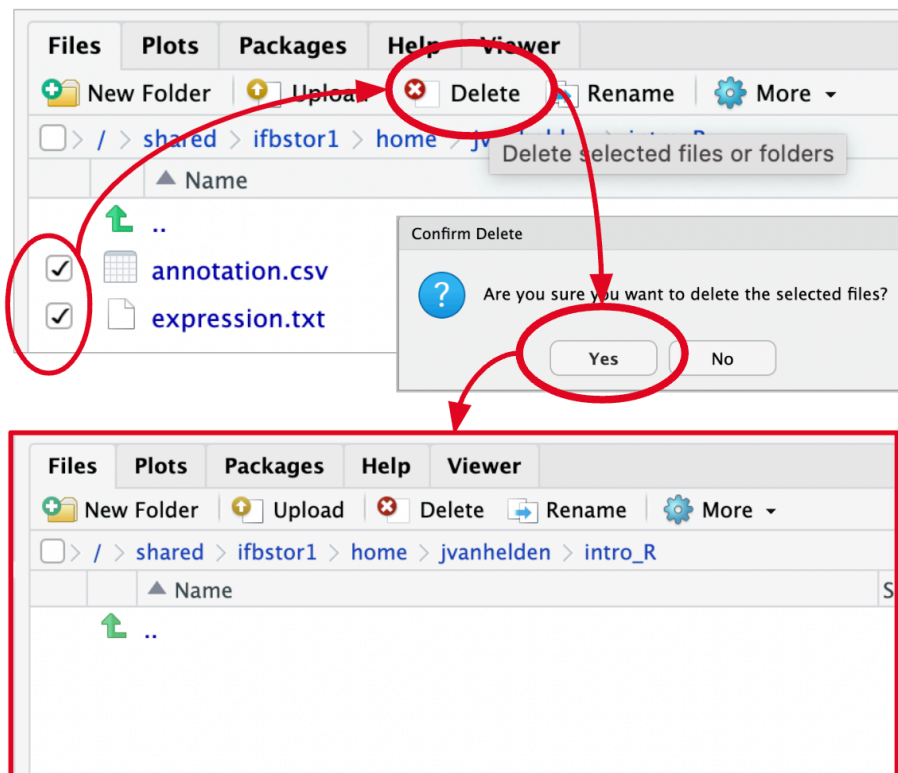
5.1.5 Téléversement (“upload”) des données

Au moyen du bouton “Upload”, téléversez les fichiers d'expression et d'annotation depuis votre ordinateur vers votre compte sur le serveur.



5.1.6 On efface tout et on recommence

1. Sélectionnez les deux fichiers
2. Effacez-les sans pitié



(nous allons vous montrer deux autres façons de les téléverser)

5.2 The “R geek” way (V2, directement depuis Rstudio)

Attention ! Dans votre espace projet !

5.2.1 Creation de l’arborescence

Aller dans **votre** espace projet !

Dans tous les commandes ci-dessous, remplacer toujours `form_2022_32/EBaII_IntroR` par votre nom d’espace projet

Note : Pour les personnes ne travaillant pas sur le cluster mais par exemple en local, vous pouvez sans soucis remplacer l’adresse par une adresse sur votre ordinateur.

```
setwd("/shared/ibfstor1/projects/form_2022_32/EBAII_IntroR")
```

Définir une variable qui indique le chemin du dossier de travail (working directory).

```
my_work_dir <- "/shared/ibfstor1/projects/form_2022_32/EBAII_IntroR/intro_R"
```

S'il n'existe pas encore, créer le dossier de travail. (Commande Unix équivalente: `mkdir -p /shared/ibfstor1/projects/form_2022_32/EBAII_IntroR/intro_R`)

```
dir.create(my_work_dir, recursive = TRUE, showWarnings = FALSE)
```

Où suis-je ? (Commande Unix équivalente: `pwd`)

```
getwd()
```

```
## [1] "/shared/ibfstor1/projects/form_2022_32/EBAII_IntroR"
```

Aller dans ce dossier de travail (Commande Unix équivalente: `cd /shared/ibfstor1/projects/form_2022_32/EBAII_IntroR`)

```
setwd(my_work_dir)
```

Et maintenant, où suis-je ?

```
getwd()
```

```
## [1] "/shared/ibfstor1/projects/form_2022_32/EBAII_IntroR"
```

Qu'y a-t-il par ici ? (Commande Unix équivalente: `ls`)

```
list.files()
```

```
## [1] "_bookdown_files"      "_bookdown.yml"        "_main_files"
## [4] "_main.log"            "_main.pdf"            "_main.Rmd"
## [7] "_main.tex"            "_output.yml"          "01-intro.Rmd"
## [10] "02-how.Rmd"           "03-firstSteps.Rmd"    "04-uploadData.Rmd"
## [13] "05-readData.Rmd"      "06-manipulate.Rmd"    "07-plots.Rmd"
## [16] "08-analyseDiff.Rmd"   "09-integration.Rmd"   "10-visu.Rmd"
## [19] "11-conclusion.Rmd"     "12-references.Rmd"     "annotation.csv"
## [22] "book.bib"             "docs"                  "EBAII_IntroR.Rproj"
## [25] "expression.txt"        "exprs_chr8.txt"        "images"
## [28] "index.Rmd"            "intro_R"               "LICENSE"
## [31] "packages.bib"         "preamble.tex"          "README.md"
## [34] "style.css"
```

Un autre nom pour la même commande

```
dir()
```

```
## [1] "_bookdown_files"    "_bookdown.yml"      "_main_files"
## [4] "_main.log"          "_main.pdf"          "_main.Rmd"
## [7] "_main.tex"          "_output.yml"         "01-intro.Rmd"
## [10] "02-how.Rmd"         "03-firstSteps.Rmd"  "04-uploadData.Rmd"
## [13] "05-readData.Rmd"    "06-manipulate.Rmd"  "07-plots.Rmd"
## [16] "08-analyseDiff.Rmd" "09-integration.Rmd" "10-visu.Rmd"
## [19] "11-conclusion.Rmd"  "12-references.Rmd"  "annotation.csv"
## [22] "book.bib"           "docs"               "EBaII_IntroR.Rproj"
## [25] "expression.txt"     "exprs_chr8.txt"     "images"
## [28] "index.Rmd"          "intro_R"            "LICENSE"
## [31] "packages.bib"       "preamble.tex"       "README.md"
## [34] "style.css"
```

5.2.2 Télécharger un fichier

Nous avons montré ci-dessus comment télécharger des fichiers en utilisant l’interface graphique de RStudio.

Alternativement, on peut télécharger des fichiers au moyen de la commande R `download.file`.

Les deux commandes suivantes permettent de télécharger les fichiers utilisés pour les exercices.

```
download.file(url = "https://raw.githubusercontent.com/IFB-ElixirFr/EBaII/master/2022/ebain1/intro.Rmd",
```

```
download.file(url = "https://raw.githubusercontent.com/IFB-ElixirFr/EBaII/master/2022/ebain1/intro.Rmd",
```

Note : équivalent de la commande `wget` sous Unix.

Qu’y a-t-il par ici ? (Commande Unix équivalente: `ls`)

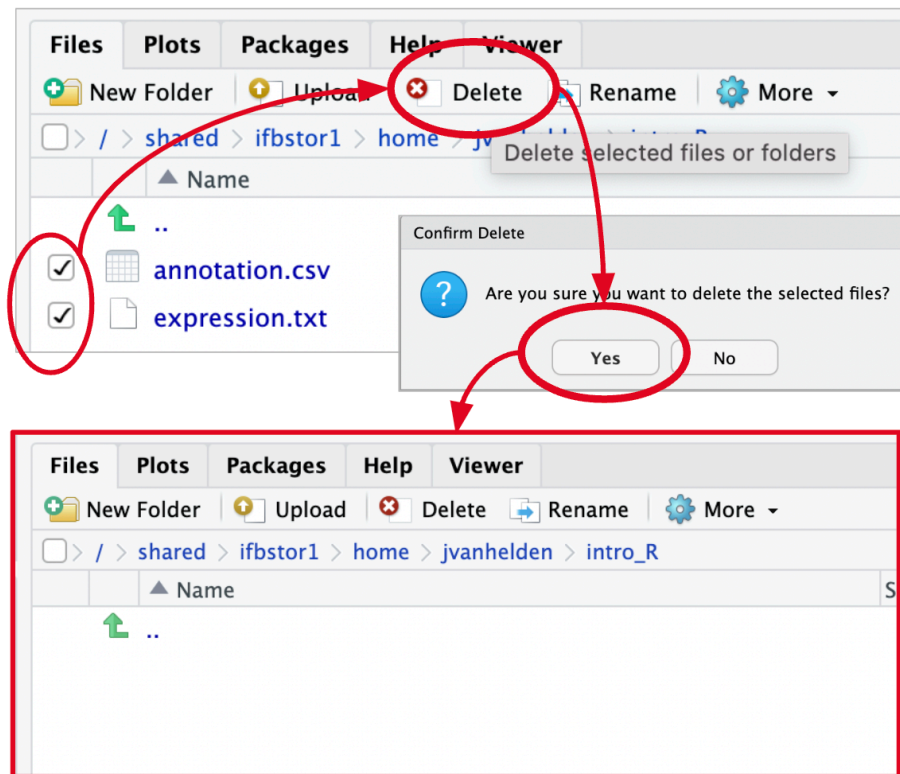
```
list.files()
```

```
## [1] "_bookdown_files"    "_bookdown.yml"      "_main_files"
## [4] "_main.log"          "_main.pdf"          "_main.Rmd"
## [7] "_main.tex"          "_output.yml"         "01-intro.Rmd"
## [10] "02-how.Rmd"         "03-firstSteps.Rmd"  "04-uploadData.Rmd"
## [13] "05-readData.Rmd"    "06-manipulate.Rmd"  "07-plots.Rmd"
## [16] "08-analyseDiff.Rmd" "09-integration.Rmd" "10-visu.Rmd"
```

```
## [19] "11-conclusion.Rmd" "12-references.Rmd" "annotation.csv"
## [22] "book.bib"         "docs"             "EBAIL_IntroR.Rproj"
## [25] "expression.txt"    "exprs_chr8.txt"   "images"
## [28] "index.Rmd"        "intro_R"          "LICENSE"
## [31] "packages.bib"     "preamble.tex"     "README.md"
## [34] "style.css"
```

5.2.3 On efface tout et on recommence

1. Sélectionnez les deux fichiers
2. Effacez-les sans pitié



Nous allons vous montrer une dernière façon de les téléverser.

5.3 The “bash geek” way (V3, directement de votre home du cluster)

Objectif

5.3. THE “BASH GEEK” WAY (V3, DIRECTEMENT DE VOTRE HOME DU CLUSTER)31

Dans le terminal du cluster, téléchargez et enregistrez dans votre home les fichiers de données: - expression.txt: données d’expressions pour 4 échantillons
- annotation.csv: informations sur les gènes (id, name, chr, start, stop)

Ouvrez un connection ssh

```
ssh [votre_login]@core.cluster.france-bioinformatique.fr
```

Où suis-je ?

```
pwd
```

```
## /shared/ifbstor1/projects/form_2022_32/EBAII_IntroR
```

Créez un répertoire “intro_R”

```
mkdir -p /shared/ifbstor1/projects/form_2022_32/EBAII_IntroR/intro_R
```

Déplacez-vous dans votre dossier

```
cd /shared/ifbstor1/projects/form_2022_32/EBAII_IntroR/intro_R
```

Où suis-je maintenant ?

```
pwd
```

```
## /shared/ifbstor1/projects/form_2022_32/EBAII_IntroR
```

Téléchargez les données

```
wget https://raw.githubusercontent.com/IFB-ElixirFr/EBAII/master/2022/ebain1/intro_R/expression.
```

```
## --2022-11-15 17:29:37-- https://raw.githubusercontent.com/IFB-ElixirFr/EBAII/master/2022/ebai
## Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.10
## Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... co
## HTTP request sent, awaiting response... 200 OK
## Length: 1747 (1.7K) [text/plain]
## Saving to: 'expression.txt'
##
##      OK .                               100% 18.3M=0s
##
## 2022-11-15 17:29:37 (18.3 MB/s) - 'expression.txt' saved [1747/1747]
```

```
wget https://raw.githubusercontent.com/IFB-ElixirFr/EBAIL/master/2022/ebain1/intro_R/

## --2022-11-15 17:29:37-- https://raw.githubusercontent.com/IFB-ElixirFr/EBAIL/master/2022/ebain1/intro_R/
## Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133,
## Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133:
## HTTP request sent, awaiting response... 200 OK
## Length: 2326 (2.3K) [text/plain]
## Saving to: 'annotation.csv'
##
##      OK ..                                     100% 26.2M=0s
##
## 2022-11-15 17:29:37 (26.2 MB/s) - 'annotation.csv' saved [2326/2326]
```

Qu'y a-t-il ici ?

```
ls -l
```

```
## total 316
## -rw-r--r--+ 1 tdenecker tdenecker 1843 Nov 15 09:19 01-intro.Rmd
## -rw-r--r--+ 1 tdenecker tdenecker 996 Nov 15 09:40 02-how.Rmd
## -rw-r--r--+ 1 tdenecker tdenecker 1478 Nov 15 09:48 03-firstSteps.Rmd
## -rw-r--r--+ 1 tdenecker tdenecker 5467 Nov 15 10:30 04-uploadData.Rmd
## -rw-r--r--+ 1 tdenecker tdenecker 1790 Nov 15 10:46 05-readData.Rmd
## -rw-r--r--+ 1 tdenecker tdenecker 1419 Nov 15 11:57 06-manipulate.Rmd
## -rw-r-----+ 1 tdenecker tdenecker 1882 Nov 15 12:20 07-plots.Rmd
## -rw-r-----+ 1 tdenecker tdenecker 2490 Nov 15 12:38 08-analyseDiff.Rmd
## -rw-r-----+ 1 tdenecker tdenecker 1490 Nov 15 15:51 09-integration.Rmd
## -rw-r-----+ 1 tdenecker tdenecker 1422 Nov 15 17:19 10-visu.Rmd
## -rw-r-----+ 1 tdenecker tdenecker 1128 Nov 15 12:35 11-conclusion.Rmd
## -rw-r--r--+ 1 tdenecker tdenecker 54 Nov 14 21:51 12-references.Rmd
## -rw-rw----+ 1 tdenecker tdenecker 2326 Nov 15 17:29 annotation.csv
## -rw-r--r--+ 1 tdenecker tdenecker 267 Nov 14 21:51 book.bib
## drwxrwx---+ 2 tdenecker tdenecker 4096 Nov 15 17:29 _bookdown_files
## -rw-r--r--+ 1 tdenecker tdenecker 113 Nov 15 16:00 _bookdown.yml
## drwxrwx---+ 5 tdenecker tdenecker 12288 Nov 15 17:29 docs
## -rw-rw----+ 1 tdenecker tdenecker 247 Nov 15 15:48 EBAIL_IntroR.Rproj
## -rw-rw----+ 1 tdenecker tdenecker 1747 Nov 15 17:29 expression.txt
## -rw-rw----+ 1 tdenecker tdenecker 244 Nov 15 17:29 exprs_chr8.txt
## drwxrwx---+ 2 tdenecker tdenecker 4096 Nov 15 12:35 images
## -rw-r--r--+ 1 tdenecker tdenecker 1460 Nov 15 17:29 index.Rmd
## drwxrwx---+ 2 tdenecker tdenecker 4096 Nov 15 10:25 intro_R
## -rw-rw----+ 1 tdenecker tdenecker 1551 Nov 14 21:50 LICENSE
## drwxrwx---+ 4 tdenecker tdenecker 4096 Nov 15 17:29 _main_files
## -rw-rw----+ 1 tdenecker tdenecker 37392 Nov 15 17:28 _main.log
```



```
## -rw-rw-----+ 1 tdenecker tdenecker 13209 Nov 15 17:28 _main.pdf
## -rw-r--r--+ 1 tdenecker tdenecker 23395 Nov 15 17:29 _main.Rmd
## -rw-rw-----+ 1 tdenecker tdenecker 102254 Nov 15 17:28 _main.tex
## -rw-r--r--+ 1 tdenecker tdenecker 500 Nov 14 21:52 _output.yml
## -rw-rw-----+ 1 tdenecker tdenecker 2655 Nov 15 17:29 packages.bib
## -rw-r--r--+ 1 tdenecker tdenecker 22 Nov 14 21:51 preamble.tex
## -rw-r--r--+ 1 tdenecker tdenecker 311 Nov 15 09:29 README.md
## -rw-r--r--+ 1 tdenecker tdenecker 172 Nov 14 21:51 style.css
```

A quoi ressemblent ces fichiers ?

```
head expression.txt
```

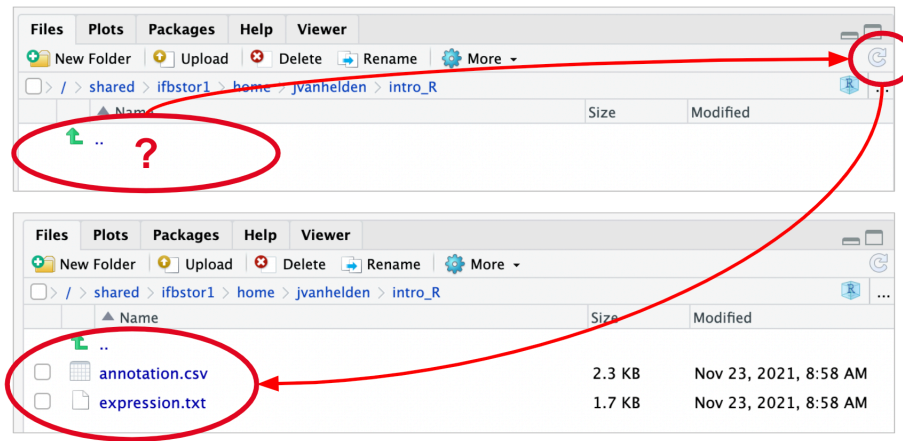
```
## id WT1 WT2 KO1 KO2
## ENSG00000034510 235960 94264 202381 91336
## ENSG00000064201 116 71 64 56
## ENSG00000065717 118 174 124 182
## ENSG00000099958 450 655 301 472
## ENSG00000104164 4736 5019 4845 4934
## ENSG00000104783 9002 8623 7720 7142
## ENSG00000105229 1295 2744 1113 2887
## ENSG00000105723 3353 7449 3589 7202
## ENSG00000116199 2044 4525 2604 4902
```

```
head annotation.csv
```

```
## id;name;chr;start;stop;strand
## ENSG00000225630;MTND2P28;1;629640;630683;+
## ENSG00000134198;TSPAN2;1;115048011;115089500;-
## ENSG00000116199;FAM20B;1;179025804;179076562;+
## ENSG00000119285;HEATR1;1;236549005;236604504;-
## ENSG00000034510;TMSB10;2;84905625;84906675;+
## ENSG00000198586;TLK1;2;170990823;171231314;-
## ENSG00000157036;EXOG;3;38496127;38542161;+
## ENSG00000157869;RAB28;4;13361354;13484365;-
## ENSG00000250202;RP11-397E7.2;4;86876338;86876652;+
```

5.4 Actualisation du dossier

Dans certains cas, il faut actualiser le contenu du dossier pour pouvoir voir le nouveau sous-dossier. Vérifiez ensuite si `intro_R` apparaît bien dans le contenu de votre dossier principal.



Chapter 6

Lecture des données

6.1 Chargement des données (dans la mémoire de R)

Charger le contenu du fichier “expression.txt” dans une variable nommée “exprs”.

```
exprs <- read.table(file = "expression.txt", header = TRUE, sep = "\t")
```

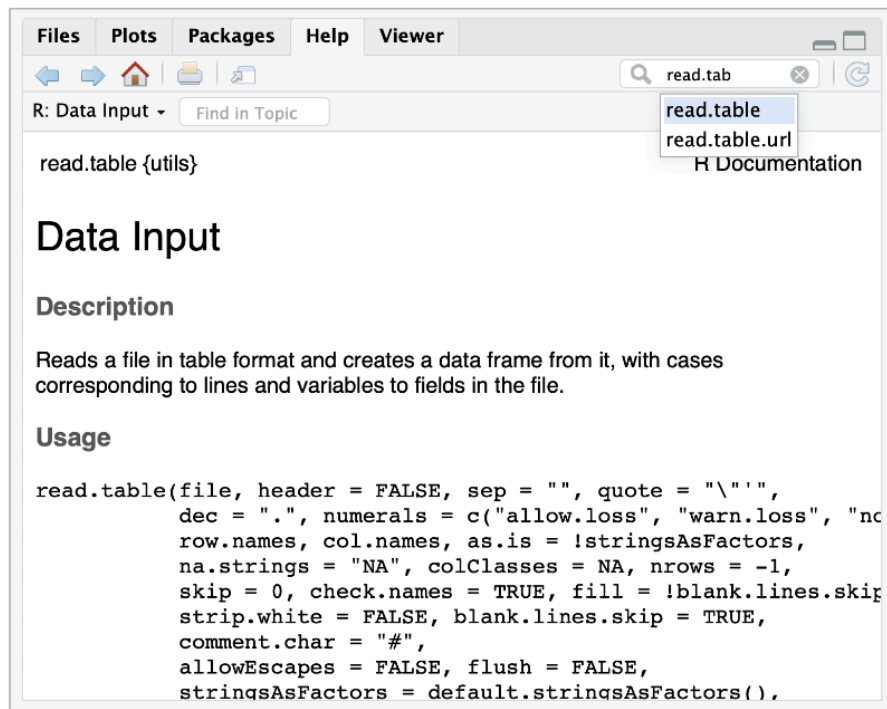
Accéder à l’aide d’une fonction

```
help(read.table)
```

Notation alternative

```
?read.table
```

Recherche interactive sous RStudio - Sélectionner l’onglet “Help” du panneau inférieur droit. - Taper “read.table” dans la boîte de recherche.



Sinon, une approche plus simple et plus pratique : - demande à Google “Comment lire une table en R ?” - adapte l’exemple

6.2 Affichage de l’objet “exprs”

Imprimer toutes les valeurs.

```
print(exprs)
```

```
##           id    WT1  WT2   K01  K02
## 1 ENSG00000034510 235960 94264 202381 91336
## 2 ENSG000000064201   116    71    64    56
## 3 ENSG000000065717   118   174   124   182
## 4 ENSG000000099958   450   655   301   472
## 5 ENSG000000104164  4736  5019  4845  4934
## 6 ENSG000000104783  9002  8623  7720  7142
## 7 ENSG000000105229  1295  2744  1113  2887
## 8 ENSG000000105723  3353  7449  3589  7202
## 9 ENSG000000116199  2044  4525  2604  4902
## 10 ENSG000000118939  7022  2526  6269  3068
```

```

## 11 ENSG00000119285 15783 17359 18591 20077
## 12 ENSG00000121680 3133 2775 2045 2796
## 13 ENSG00000125384 1380 3079 869 2419
## 14 ENSG00000129562 12089 7958 10708 7683
## 15 ENSG00000129932 1744 2247 1513 3104
## 16 ENSG00000134198 122 66 44 16
## 17 ENSG00000135452 635 427 662 291
## 18 ENSG00000140416 83 246 136 267
## 19 ENSG00000147274 16013 17642 15055 18804
## 20 ENSG00000148090 552 1062 615 1082
## 21 ENSG00000148248 62324 33973 56862 37710
## 22 ENSG00000157036 1225 1475 1275 1373
## 23 ENSG00000157869 1201 1034 1025 858
## 24 ENSG00000159433 31 788 30 675
## 25 ENSG00000161692 695 1825 746 1851
## 26 ENSG00000167005 26866 23111 24888 22661
## 27 ENSG00000168517 273 112 190 77
## 28 ENSG00000169570 202 181 207 209
## 29 ENSG00000172216 3515 1981 3204 3174
## 30 ENSG00000175221 1988 4788 2115 5306
## 31 ENSG00000183161 2238 974 2089 996
## 32 ENSG00000185324 1236 2163 1048 2024
## 33 ENSG00000188985 3415 1703 3587 2096
## 34 ENSG00000196867 209 189 293 192
## 35 ENSG00000197081 14741 36309 14941 29645
## 36 ENSG00000198586 1216 4545 1660 3932
## 37 ENSG00000214121 4044 2575 3019 2506
## 38 ENSG00000225630 1405 8135 1569 7866
## 39 ENSG00000226742 158 94 153 178
## 40 ENSG00000238241 90 43 122 143
## 41 ENSG00000248751 518 718 411 597
## 42 ENSG00000250202 261 163 177 191
## 43 ENSG00000251106 94 114 63 86
## 44 ENSG00000253991 77 78 134 92
## 45 ENSG00000254470 3025 3707 2558 4066
## 46 ENSG00000262814 15470 11450 11656 13821
## 47 ENSG00000267228 3801 2465 2787 2301
## 48 ENSG00000267699 1488 1086 1374 939
## 49 ENSG00000269293 424 162 310 120
## 50 ENSG00000279329 55 76 58 70

```

Affichage des premières lignes de l'objet

```
head(exprs)
```

```
##           id      WT1      WT2      K01      K02
```

```
## 1 ENSG00000034510 235960 94264 202381 91336
## 2 ENSG00000064201      116    71      64    56
## 3 ENSG00000065717      118    174     124    182
## 4 ENSG00000099958      450    655     301    472
## 5 ENSG00000104164     4736   5019    4845   4934
## 6 ENSG00000104783     9002   8623    7720   7142
```

Affichage des dernières lignes de l'objet

```
tail(exprs)
```

```
##           id    WT1    WT2    K01    K02
## 45 ENSG00000254470 3025  3707  2558  4066
## 46 ENSG00000262814 15470 11450 11656 13821
## 47 ENSG00000267228 3801  2465  2787  2301
## 48 ENSG00000267699 1488  1086  1374   939
## 49 ENSG00000269293  424   162   310   120
## 50 ENSG00000279329   55    76    58    70
```

Un peu plus de lignes

```
head(exprs, n = 15)
```

```
##           id    WT1    WT2    K01    K02
## 1 ENSG00000034510 235960 94264 202381 91336
## 2 ENSG00000064201      116    71      64    56
## 3 ENSG00000065717      118    174     124    182
## 4 ENSG00000099958      450    655     301    472
## 5 ENSG00000104164     4736   5019    4845   4934
## 6 ENSG00000104783     9002   8623    7720   7142
## 7 ENSG00000105229     1295   2744    1113   2887
## 8 ENSG00000105723     3353   7449    3589   7202
## 9 ENSG00000116199     2044   4525    2604   4902
## 10 ENSG00000118939     7022   2526    6269   3068
## 11 ENSG00000119285    15783  17359   18591  20077
## 12 ENSG00000121680     3133   2775     2045   2796
## 13 ENSG00000125384     1380   3079     869   2419
## 14 ENSG00000129562    12089   7958   10708   7683
## 15 ENSG00000129932     1744   2247     1513   3104
```

Explorer le tableau dans un panneau de visualisation

```
View(exprs)
```

Note: vous pouvez cliquer sur une en-tête de colonne pour trier les données

Explorer le tableau avec le package DT

```
library(DT)
datatable(exprs)
```

```
## PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is installed,
```

6.3 Caractéristiques d'un tableau de données

6.3.1 Dimensions

Nombre de colonnes

```
ncol(exprs)
```

```
## [1] 5
```

Nombre de lignes

```
nrow(exprs)
```

```
## [1] 50
```

Dimensions

```
dim(exprs)
```

```
## [1] 50 5
```

6.3.2 Noms des colonnes et des lignes

Noms des colonnes

```
colnames(exprs)
```

```
## [1] "id" "WT1" "WT2" "K01" "K02"
```

Idem

```
names(exprs)
```

```
## [1] "id" "WT1" "WT2" "K01" "K02"
```

Noms des lignes

```
rownames(exprs)
```

```
## [1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11" "12" "13" "14" "15"
## [16] "16" "17" "18" "19" "20" "21" "22" "23" "24" "25" "26" "27" "28" "29" "30"
## [31] "31" "32" "33" "34" "35" "36" "37" "38" "39" "40" "41" "42" "43" "44" "45"
## [46] "46" "47" "48" "49" "50"
```

6.3.3 Résumé rapide des données par colonne

Statistiques par colonne

```
summary(exprs)
```

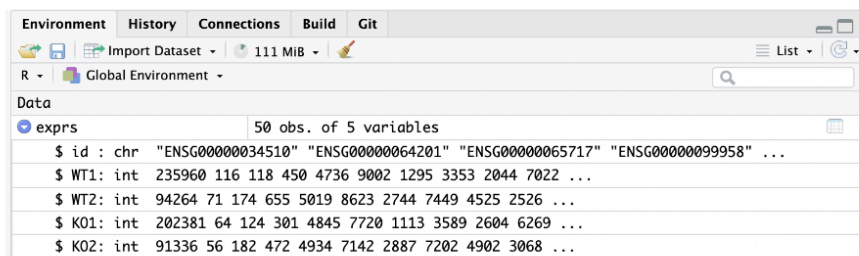
```
##          id                WT1                WT2                K01
## Length:50          Min.   :    31          Min.   :   43.0          Min.   :    30.0
## Class :character    1st Qu.:   264          1st Qu.:  203.2          1st Qu.:   228.5
## Mode  :character    Median :  1338          Median :  1903.0          Median :  1324.5
##                                     Mean   :   9358          Mean   :  6498.6          Mean   :   8356.0
##                                     3rd Qu.:   3730          3rd Qu.:  4727.2          3rd Qu.:   3491.2
##                                     Max.   :235960          Max.   :94264.0          Max.   :202381.0
##          K02
## Min.   :    16.0
## 1st Qu.:   223.5
## Median :  2060.0
## Mean   :  6489.5
## 3rd Qu.:  4926.0
## Max.   :91336.0
```

Structure de la variable


```
str(exprs)
```

```
## 'data.frame':   50 obs. of  5 variables:
## $ id : chr  "ENSG00000034510" "ENSG00000064201" "ENSG00000065717" "ENSG00000099958" ...
## $ WT1: int  235960 116 118 450 4736 9002 1295 3353 2044 7022 ...
## $ WT2: int  94264 71 174 655 5019 8623 2744 7449 4525 2526 ...
## $ K01: int  202381 64 124 301 4845 7720 1113 3589 2604 6269 ...
## $ K02: int  91336 56 182 472 4934 7142 2887 7202 4902 3068 ...
```

Même résultat que dans le panneau “Environment”



Chapter 7

Manipuler les données dans R

7.1 Sélection de colonnes d'un tableau

Afficher les noms des colonnes

```
colnames(exprs)
```

```
## [1] "id" "WT1" "WT2" "K01" "K02"
```

Valeurs stockées dans la colonne nommée “WT1”

```
exprs$WT1
```

```
## [1] 235960    116    118    450    4736    9002    1295    3353    2044    7022
## [11] 15783    3133    1380   12089    1744     122     635     83   16013    552
## [21] 62324    1225    1201     31     695   26866     273    202    3515   1988
## [31]  2238    1236    3415     209   14741    1216    4044    1405    158     90
## [41]   518     261     94      77    3025   15470    3801    1488     424     55
```

Notation alternative

```
exprs[, "WT1"]
```

```
## [1] 235960    116    118    450    4736    9002    1295    3353    2044    7022
## [11] 15783    3133    1380   12089    1744     122     635     83   16013    552
```

```
## [21] 62324 1225 1201 31 695 26866 273 202 3515 1988
## [31] 2238 1236 3415 209 14741 1216 4044 1405 158 90
## [41] 518 261 94 77 3025 15470 3801 1488 424 55
```

Sélection de plusieurs colonnes.

```
exprs[ , c("WT1", "WT2")]
```

```
##      WT1  WT2
## 1 235960 94264
## 2    116    71
## 3    118    174
## 4    450    655
## 5   4736   5019
## 6   9002   8623
## 7   1295   2744
## 8   3353   7449
## 9   2044   4525
## 10  7022   2526
## 11 15783 17359
## 12  3133   2775
## 13  1380   3079
## 14 12089   7958
## 15  1744   2247
## 16   122    66
## 17   635   427
## 18    83   246
## 19 16013 17642
## 20   552  1062
## 21 62324 33973
## 22  1225  1475
## 23  1201  1034
## 24    31   788
## 25   695  1825
## 26 26866 23111
## 27   273   112
## 28   202   181
## 29  3515 1981
## 30  1988 4788
## 31  2238   974
## 32  1236  2163
## 33  3415  1703
## 34   209   189
## 35 14741 36309
## 36  1216  4545
```

```
## 37  4044  2575
## 38  1405  8135
## 39   158   94
## 40   90   43
## 41   518  718
## 42   261  163
## 43   94  114
## 44   77   78
## 45  3025 3707
## 46 15470 11450
## 47  3801 2465
## 48  1488 1086
## 49   424  162
## 50    55   76
```

Sélection de colonnes par leur indice

```
exprs[ , 2]
```

```
## [1] 235960   116   118   450  4736  9002  1295  3353  2044  7022
## [11] 15783   3133  1380 12089  1744   122   635   83 16013   552
## [21] 62324  1225  1201   31   695 26866  273   202  3515  1988
## [31] 2238   1236  3415   209 14741  1216  4044  1405   158   90
## [41]   518   261    94    77  3025 15470  3801  1488   424   55
```

```
exprs[ , c( 3, 2)]
```

```
##      WT2   WT1
## 1  94264 235960
## 2    71   116
## 3   174   118
## 4   655   450
## 5   5019  4736
## 6   8623  9002
## 7   2744  1295
## 8   7449  3353
## 9   4525  2044
## 10  2526  7022
## 11 17359 15783
## 12  2775  3133
## 13  3079  1380
## 14  7958 12089
## 15  2247  1744
## 16    66   122
```

```
## 17 427 635
## 18 246 83
## 19 17642 16013
## 20 1062 552
## 21 33973 62324
## 22 1475 1225
## 23 1034 1201
## 24 788 31
## 25 1825 695
## 26 23111 26866
## 27 112 273
## 28 181 202
## 29 1981 3515
## 30 4788 1988
## 31 974 2238
## 32 2163 1236
## 33 1703 3415
## 34 189 209
## 35 36309 14741
## 36 4545 1216
## 37 2575 4044
## 38 8135 1405
## 39 94 158
## 40 43 90
## 41 718 518
## 42 163 261
## 43 114 94
## 44 78 77
## 45 3707 3025
## 46 11450 15470
## 47 2465 3801
## 48 1086 1488
## 49 162 424
## 50 76 55
```

7.2 Sélection de lignes d'un tableau

Sélection des lignes 4 et 11 du tableau des expressions

```
exprs[c(4, 11), ]
```

```
##           id   WT1   WT2   K01   K02
## 4  ENSG00000099958  450   655   301   472
## 11 ENSG00000119285 15783 17359 18591 20077
```

Sélection des identifiants de deux gènes d'intérêt

```
my_genes <- c("ENSG00000253991", "ENSG00000099958")
```

Vecteur booléen indiquant si chaque ID du tableau fait partie des gènes d'intérêt

```
exprs$id %in% my_genes
```

```
## [1] FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
## [49] FALSE FALSE
```

Indices des lignes correspondant aux IDs des gènes d'intérêt

```
which(exprs$id %in% my_genes)
```

```
## [1]  4 44
```

Afficher les lignes correspondantes

```
exprs[which(exprs$id %in% my_genes), ]
```

```
##           id WT1 WT2 K01 K02
## 4  ENSG00000099958 450 655 301 472
## 44 ENSG00000253991  77  78 134  92
```

7.3 formulation plus intuitive

```
subset(x = exprs, id %in% my_genes)
```

```
##           id WT1 WT2 K01 K02
## 4  ENSG00000099958 450 655 301 472
## 44 ENSG00000253991  77  78 134  92
```

Approche plus moderne, avec le package dplyr

```
## charger la librairie dplyr
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
## filter, lag
```

```
## The following objects are masked from 'package:base':
##
## intersect, setdiff, setequal, union
```

```
## envoyer le tableau exprs à la commande filter()
exprs %>% filter(id %in% my_genes)
```

```
##           id WT1 WT2 K01 K02
## 1 ENSG00000099958 450 655 301 472
## 2 ENSG00000253991  77  78 134  92
```

```
## plus avancé : enchaîner plusieurs commandes
exprs %>%
  filter(id %in% my_genes) %>%
  mutate(mean_KO = (K01 + K02)/2)
```

```
##           id WT1 WT2 K01 K02 mean_KO
## 1 ENSG00000099958 450 655 301 472   386.5
## 2 ENSG00000253991  77  78 134  92   113.0
```


Chapter 8

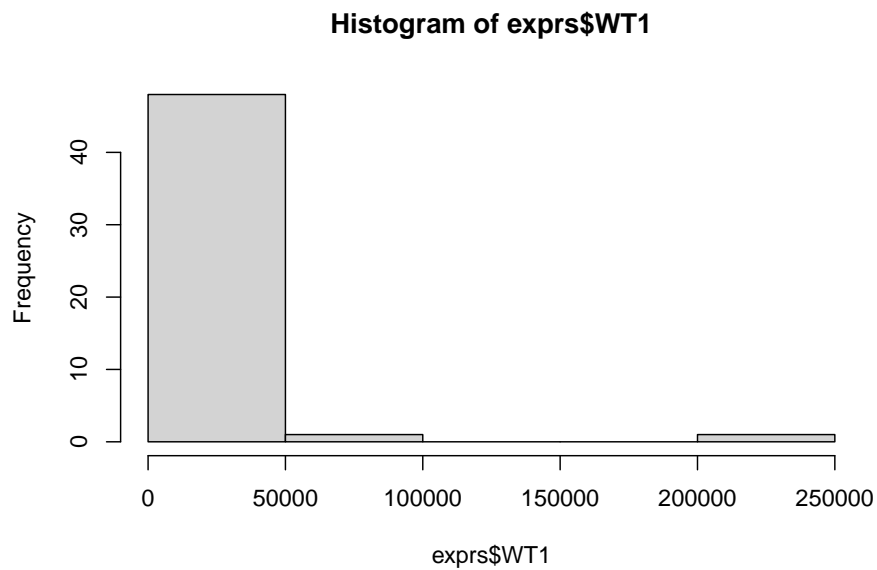
Visualisation des données

8.1 Histogrammes

8.1.1 Avec R de base

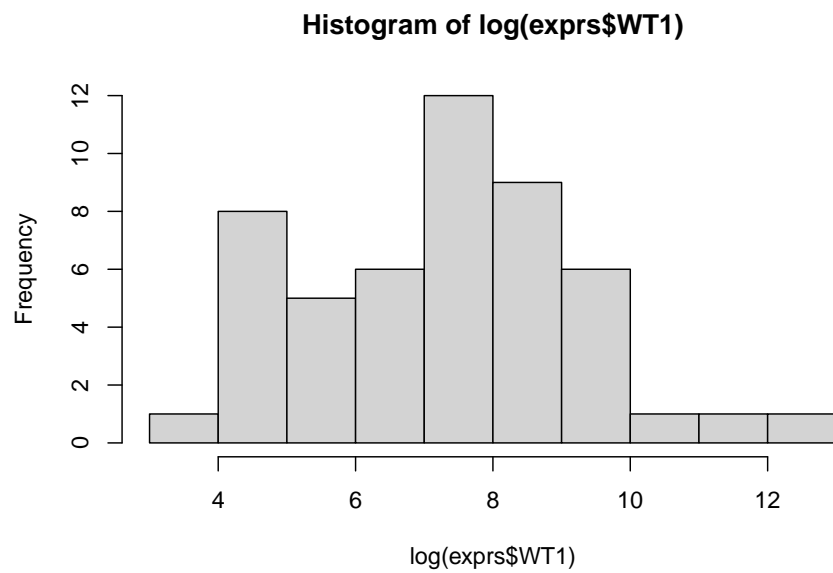
Histogramme des valeurs d'expression pour l'échantillon WT1

```
hist(exprs$WT1)
```

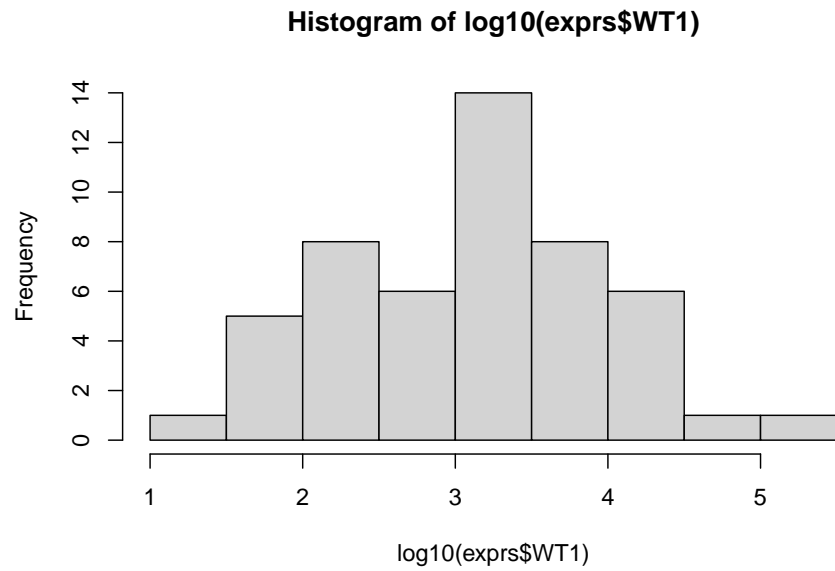


Histogramme du logarithme de ces valeurs

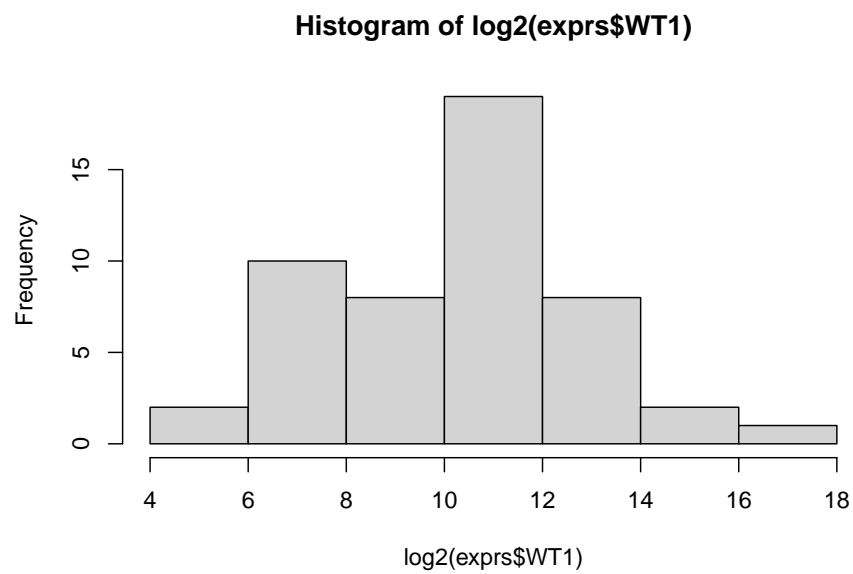
```
hist(log(exprs$WT1))
```



```
hist(log10(exprs$WT1))
```



```
hist(log2(exprs$WT1))
```



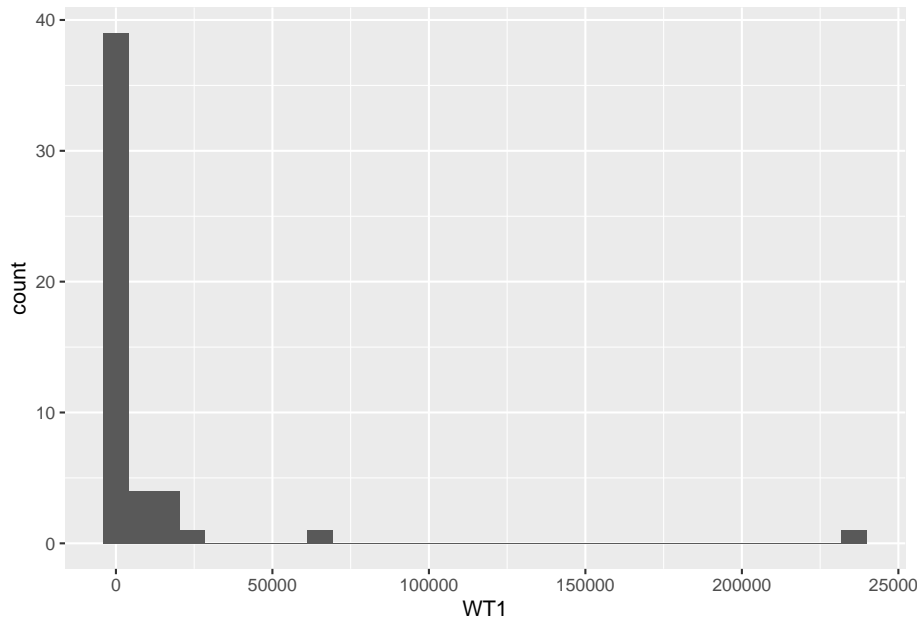
8.1.2 Avec ggplot2

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
ggplot(exprs, aes(x=WT1)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

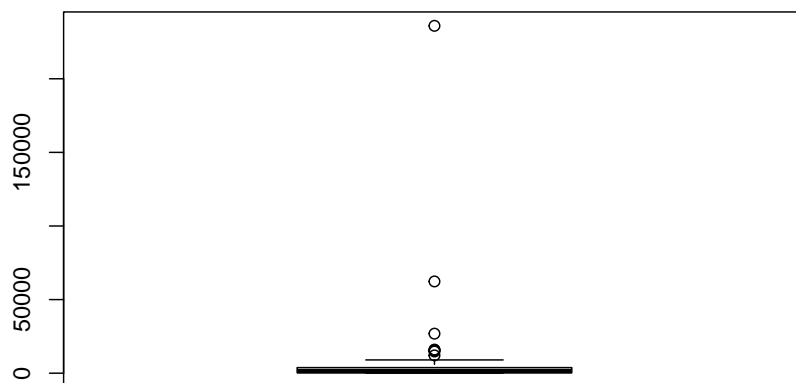


8.2 Boîtes à moustaches (boxplots)

8.2.1 Avec R de base

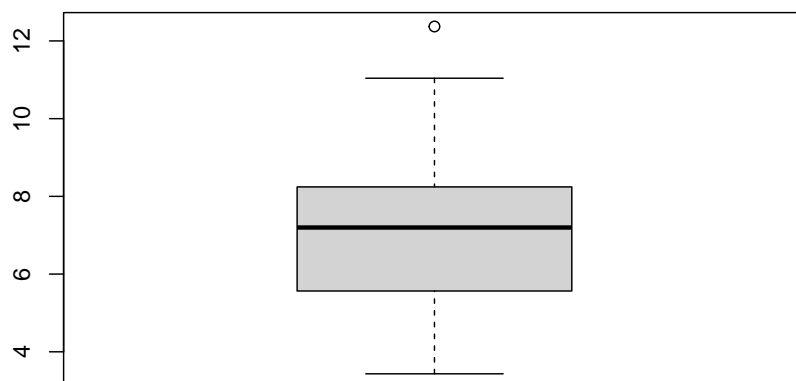
Boite à moustache des valeurs d'expression pour l'échantillon WT1

```
boxplot(exprs$WT1)
```

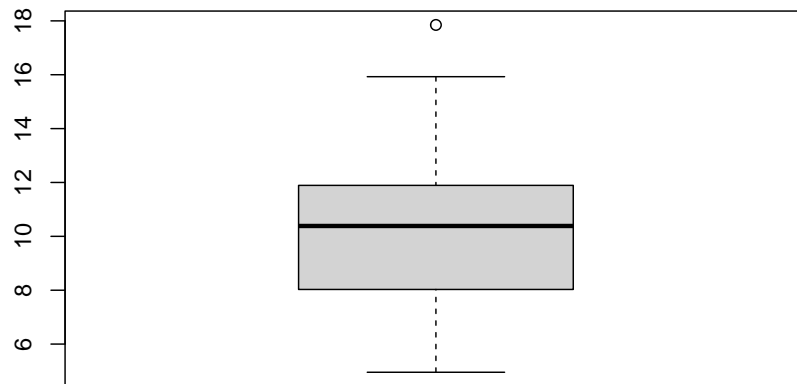


Boite à moustache du logarithme de ces valeurs

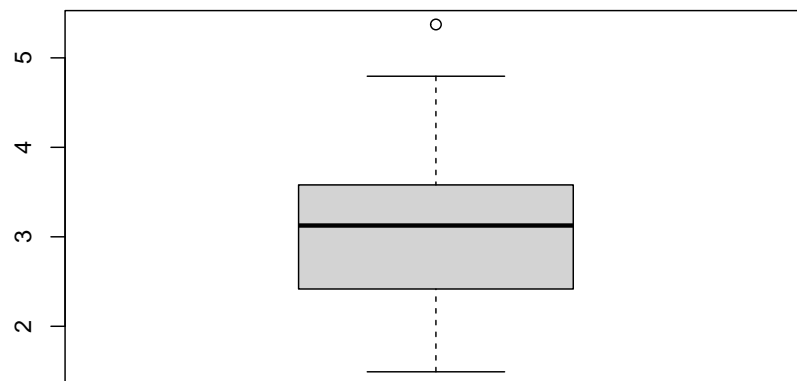
```
boxplot(log(exprs$WT1))
```



```
boxplot(log2(exprs$WT1))
```



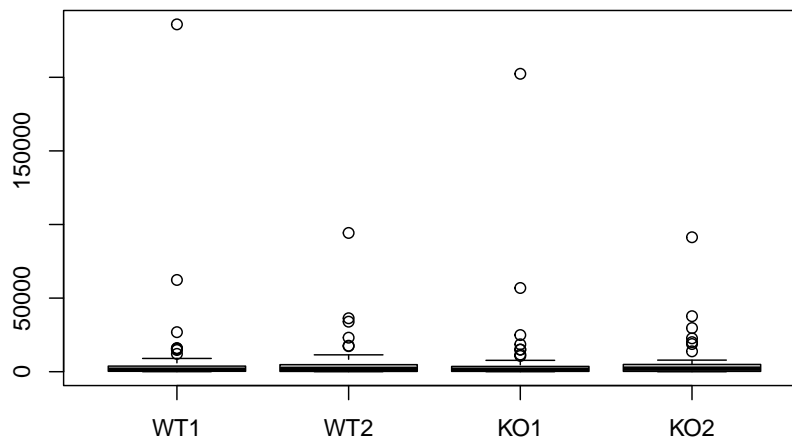
```
boxplot(log10(exprs$WT1))
```



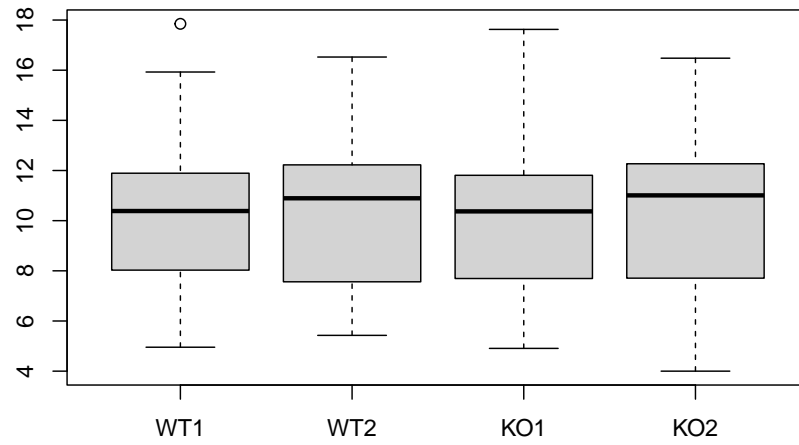
Boite à moustache des valeurs d'expression pour les 4 échantillons

```
boxplot(exprs)
```

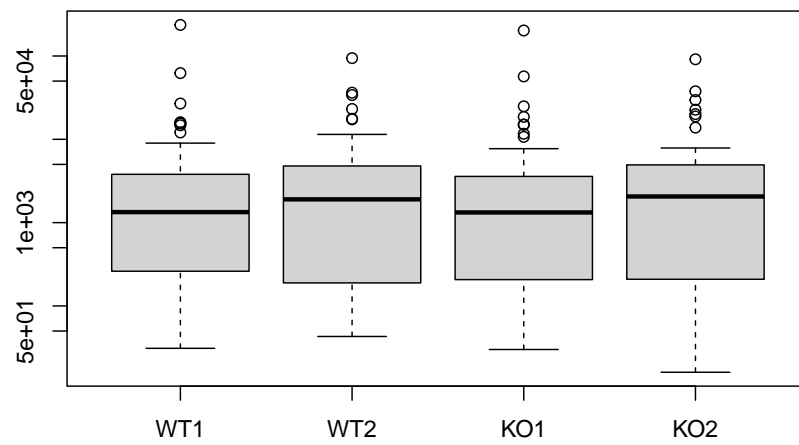
```
## ignorer la première colonne  
boxplot(exprs[, -1])
```



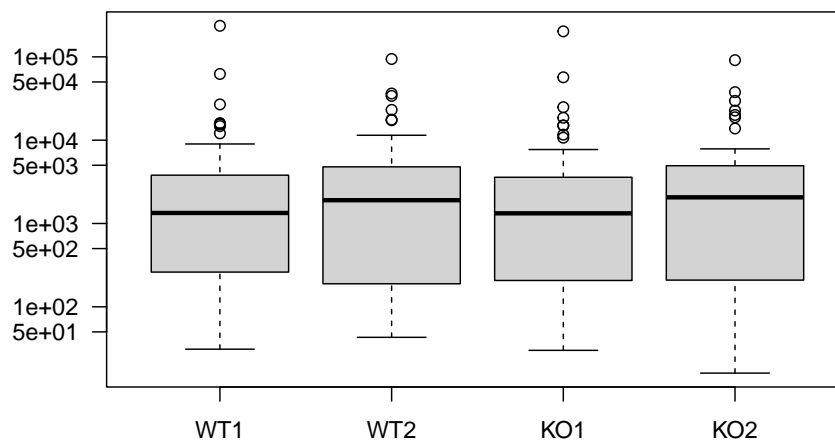
```
boxplot(log2(exprs[, -1]))
```



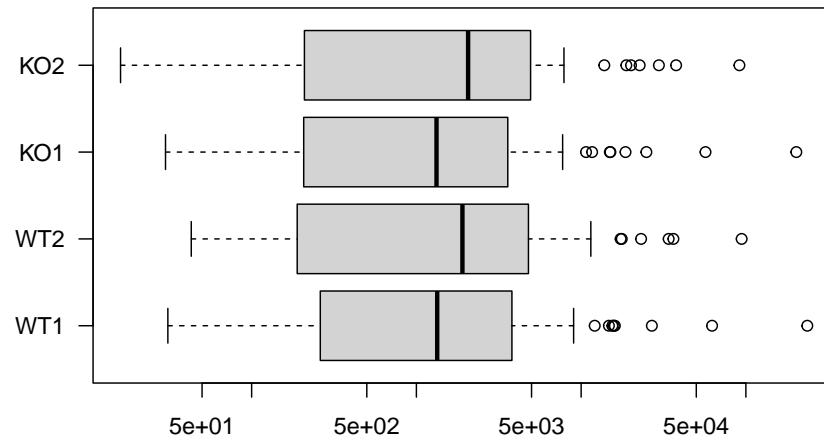
```
boxplot(exprs[, -1], log = "y")
```




```
## afficher les étiquettes des axes horizontalement  
boxplot(exprs[,-1], log = "y", las = 1)
```



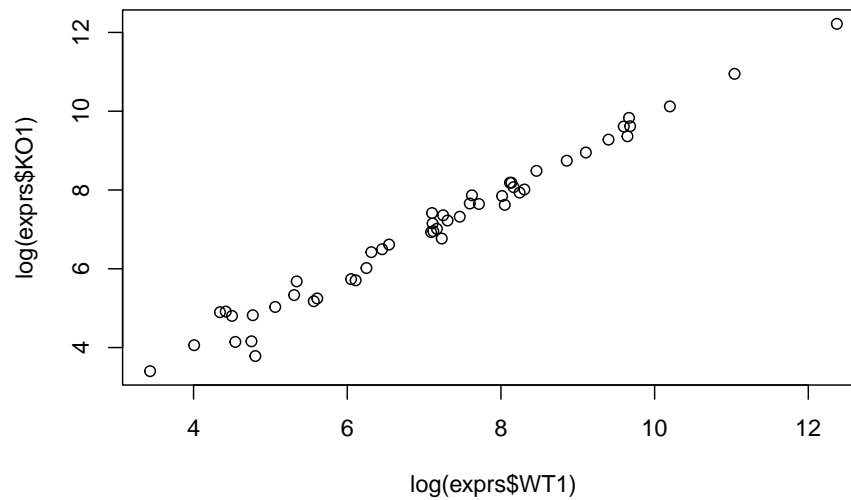
```
## Encore plus beau  
boxplot(exprs[,-1], log = "x", las = 1, horizontal = TRUE)
```



8.3 Nuage de points

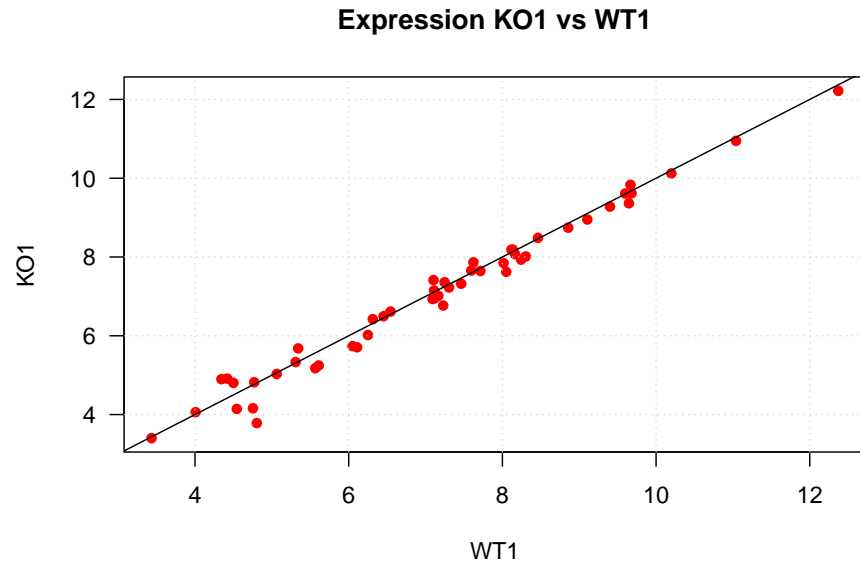
Expressions KO1 vs WT1

```
plot(x = log(exprs$WT1), y = log(exprs$KO1))
```



Personnalisation des paramètres graphiques

```
plot(x = log(exprs$WT1),          ## données pour l'abscisse
     y = log(exprs$KO1),          ## données pour l'ordonnée
     main = "Expression KO1 vs WT1", ## Titre principal
     xlab = "WT1",                ## légende de l'axe X
     ylab = "KO1",                ## légende de l'axe Y
     pch = 16,                   ## caractère pour marquer les points
     las = 1,                    ## écrire les échelles horizontalement
     col = "red")                ## couleur des points
grid()                           ## ajout d'une grille
abline(a = 0, b = 1)             ## ajouter la droite X = Y (intercept a = 0, pente b = 1)
```



Chapter 9

Analyse d'expression différentielle : MA-plot

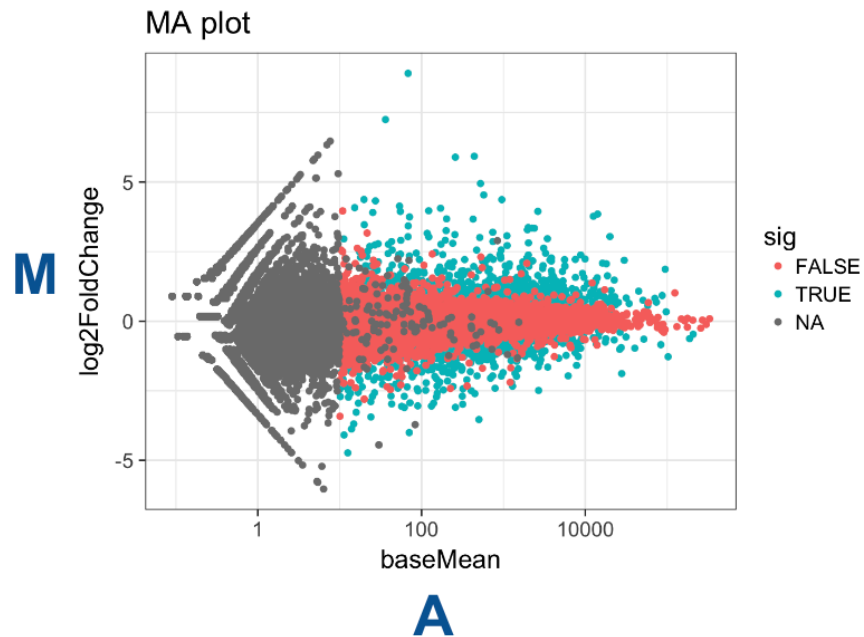
9.1 C'est quoi un MA plot

9.1.1 Nos data

```
head(exprs, 10)
```

##		id	WT1	WT2	K01	K02
## 1	ENSG000000034510	235960	94264	202381	91336	
## 2	ENSG000000064201	116	71	64	56	
## 3	ENSG000000065717	118	174	124	182	
## 4	ENSG000000099958	450	655	301	472	
## 5	ENSG00000104164	4736	5019	4845	4934	
## 6	ENSG00000104783	9002	8623	7720	7142	
## 7	ENSG00000105229	1295	2744	1113	2887	
## 8	ENSG00000105723	3353	7449	3589	7202	
## 9	ENSG00000116199	2044	4525	2604	4902	
## 10	ENSG00000118939	7022	2526	6269	3068	

9.1.2 La théorie



Le MA plot représente le lien entre différence d'expression et intensité moyenne.

- M (magnitude) est le logarithme en base 2 du rapport d'expression ("log2 fold-change")
- A (average intensity) est la moyenne des logarithmes des valeurs d'expression.

log2 fold change, "magnitude"

$$M = \log_2(KO/WT) = \log_2(KO) - \log_2(WT)$$

average log2 value

$$A = \frac{1}{2} \log_2(KO \times WT) = \frac{1}{2} (\log_2(KO) + \log_2(WT))$$

9.2 Calculs sur les colonnes

1. Calcul de moyennes par ligne (`rowMeans`) pour un sous-ensemble donné des colonnes (WT1 et WT2).

```
rowMeans(exprs[, c("WT1", "WT2")])
```

```
## [1] 165112.0    93.5    146.0    552.5    4877.5    8812.5    2019.5    5401.0
## [9]   3284.5   4774.0  16571.0   2954.0   2229.5   10023.5   1995.5     94.0
## [17]    531.0    164.5  16827.5    807.0  48148.5   1350.0   1117.5    409.5
## [25]   1260.0  24988.5    192.5    191.5   2748.0   3388.0   1606.0   1699.5
## [33]   2559.0   199.0  25525.0   2880.5   3309.5   4770.0    126.0    66.5
## [41]    618.0    212.0   104.0     77.5   3366.0  13460.0   3133.0   1287.0
## [49]    293.0     65.5
```

2. Ajout de colonnes avec les expressions moyennes des WT et des KO

```
exprs$meanWT <- rowMeans(exprs[, c("WT1", "WT2")])
exprs$meanKO <- rowMeans(exprs[, c("K01", "K02")])
```

3. Vérification des résultats

```
head(exprs)
```

```
##           id    WT1    WT2    K01    K02    meanWT    meanKO
## 1 ENSG00000034510 235960 94264 202381 91336 165112.0 146858.5
## 2 ENSG00000064201    116    71     64    56     93.5     60.0
## 3 ENSG00000065717    118   174    124   182    146.0    153.0
## 4 ENSG00000099958    450   655    301   472    552.5    386.5
## 5 ENSG00000104164   4736  5019   4845  4934   4877.5   4889.5
## 6 ENSG00000104783   9002  8623   7720  7142   8812.5   7431.0
```

4. Fold-change KO vs WT

```
exprs$FC <- exprs$meanKO / exprs$meanWT
```

5. Vérification des résultats

```
head(exprs)
```

```
##           id    WT1    WT2    K01    K02    meanWT    meanKO      FC
## 1 ENSG00000034510 235960 94264 202381 91336 165112.0 146858.5 0.8894478
## 2 ENSG00000064201    116    71     64    56     93.5     60.0 0.6417112
## 3 ENSG00000065717    118   174    124   182    146.0    153.0 1.0479452
## 4 ENSG00000099958    450   655    301   472    552.5    386.5 0.6995475
## 5 ENSG00000104164   4736  5019   4845  4934   4877.5   4889.5 1.0024603
## 6 ENSG00000104783   9002  8623   7720  7142   8812.5   7431.0 0.8432340
```

6. Moyenne de tous les échantillons

```
exprs$mean <- rowMeans(exprs[, c("WT1", "WT2", "K01", "K02")])
```

7. Vérification des résultats

```
head(exprs)
```

```
##           id    WT1   WT2   K01   K02   meanWT   meanKO      FC
## 1 ENSG00000034510 235960 94264 202381 91336 165112.0 146858.5 0.8894478
## 2 ENSG00000064201   116    71    64    56    93.5    60.0 0.6417112
## 3 ENSG00000065717   118   174   124   182   146.0   153.0 1.0479452
## 4 ENSG00000099958   450   655   301   472   552.5   386.5 0.6995475
## 5 ENSG00000104164  4736  5019  4845  4934  4877.5  4889.5 1.0024603
## 6 ENSG00000104783  9002  8623  7720  7142  8812.5  7431.0 0.8432340
##           mean
## 1 155985.25
## 2   76.75
## 3  149.50
## 4  469.50
## 5 4883.50
## 6 8121.75
```

9.3 MA-plot : log2FC vs intensité

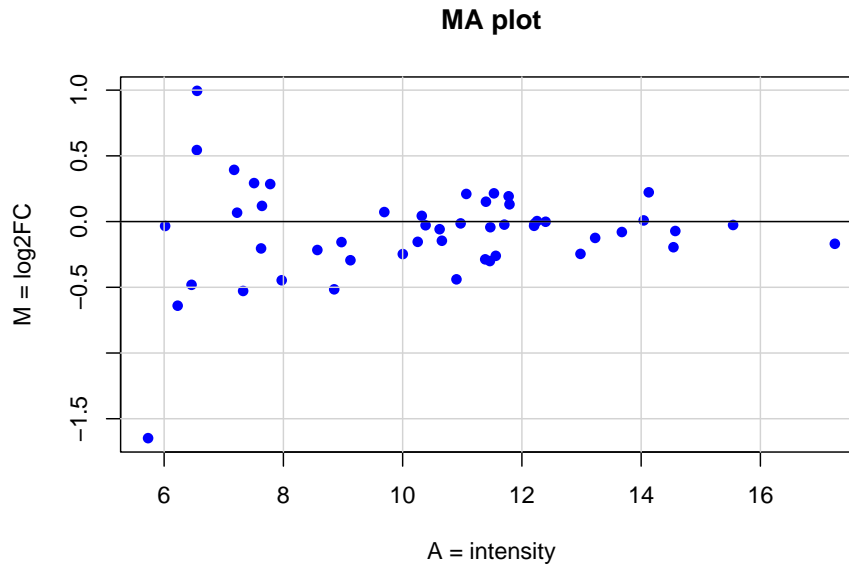
9.3.1 Calcul de M et A

```
exprs$M <- log2(exprs$FC)
exprs$A <- (log2(exprs$meanKO) + log2(exprs$meanWT)) / 2
```

9.3.2 Visualisation

On peut ensuite dessiner un nuage de points (en l'agrémentant un peu)

```
plot(x = exprs$A, y = exprs$M, main = "MA plot",
     col = "blue", pch = 16, xlab = "A = intensity", ylab = "M = log2FC")
grid(lty = "solid", col = "lightgray")
abline(h = 0)
```

9.4 Appliquer une fonction sur les lignes/colonnes

9.4.1 Appliquer une fonction (moyenne, variance, ...) sur chaque ligne d'un tableau

```
mean_per_row <- apply(exprs[ , c("WT1", "WT2", "KO1", "KO2")], 1, mean)

mean_per_row <- apply(exprs[ , c(2, 3, 4, 5)], 1, mean)

mean_per_row <- apply(exprs[ , -1 ], 1, mean)

mean_per_row <- apply(exprs[ , which(sapply(exprs, class) != "factor")], 1, mean)

## Warning in mean.default(newX[, i], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(newX[, i], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(newX[, i], ...): argument is not numeric or logical:
## returning NA
```

[illegible]


```
## Warning in mean.default(newX[, i], ...): argument is not numeric or logical:  
## returning NA
```

```
var_per_row <- apply(exprs[ , c("WT1", "WT2", "K01", "K02")], 1, var)
```

9.4.2 Appliquer une fonction (moyenne, variance, ...) sur chaque colonne d'un tableau

```
mean_per_col <- apply(exprs[ , c("WT1", "WT2", "K01", "K02")], 2, mean)  
var_per_col <- apply(exprs[ , c("WT1", "WT2", "K01", "K02")], 2, var)
```


Chapter 10

Intégration des données

10.1 Charger les annotations des gènes

```
read.table(file = "annotation.csv")
```

```
##                                     V1
## 1                                id;name;chr;start;stop;strand
## 2                ENSG00000225630;MTND2P28;1;629640;630683;+
## 3                ENSG00000134198;TSPAN2;1;115048011;115089500;-
## 4                ENSG00000116199;FAM20B;1;179025804;179076562;+
## 5                ENSG00000119285;HEATR1;1;236549005;236604504;-
## 6                ENSG00000034510;TMSB10;2;84905625;84906675;+
## 7                ENSG00000198586;TLK1;2;170990823;171231314;-
## 8                ENSG00000157036;EXOG;3;38496127;38542161;+
## 9                ENSG00000157869;RAB28;4;13361354;13484365;-
## 10                ENSG00000250202;RP11-397E7.2;4;86876338;86876652;+
## 11                ENSG00000169570;DTWD2;5;118837322;118988545;-
## 12                ENSG00000269293;ZSCAN16-AS1;6;28121795;28137293;-
## 13                ENSG00000197081;IGF2R;6;159969099;160113507;+
## 14                ENSG00000147274;RBMX;X;136848004;136880764;-
## 15                ENSG00000253991;KB-1562D12.2;8;101528723;101529569;-
## 16                ENSG00000214121;TDPX2;9;12972843;12973438;-
## 17                ENSG00000148090;AUH;9;91213815;91361913;-
## 18                ENSG00000148248;SURF4;9;133361449;133376166;-
## 19                ENSG00000064201;TSPAN32;11;2301997;2318200;+
## 20                ENSG00000183161;FANCF;11;22622519;22626787;-
## 21                ENSG00000121680;PEX16;11;45909669;45918812;-
## 22                ENSG00000254470;AP5B1;11;65775893;65780802;-
```

```

## 23      ENSG00000135452;TSPAN31;12;57738013;57750211;+
## 24      ENSG00000251106;FAM206BP;13;46270077;46270617;+
## 25      ENSG00000118939;UCHL3;13;75549480;75606020;+
## 26      ENSG00000238241;CCR12P;13;99407781;99409062;-
## 27      ENSG00000129562;DAD1;14;22564905;22589269;-
## 28      ENSG00000125384;PTGER2;14;52314305;52328606;+
## 29      ENSG00000159433;STARD9;15;42575659;42720981;+
## 30      ENSG00000104164;BLOC1S6;15;45587123;45615999;+
## 31      ENSG00000140416;TPM1;15;63042632;63071915;+
## 32      ENSG00000167005;NUDT21;16;56429133;56452199;-
## 33      ENSG00000185324;CDK10;16;89680737;89696364;+
## 34      ENSG00000161692;DBF4B;17;44708608;44752264;+
## 35      ENSG00000168517;HEXIM2;17;45160700;45170040;+
## 36      ENSG00000262814;MRPL12;17;81703357;81707526;+
## 37      ENSG00000188985;DHFRP1;18;26170726;26171284;-
## 38      ENSG00000267228;IER3IP1;18;47134656;47176281;-
## 39      ENSG00000267699;RP11-729L2.2;18;50968019;51058144;+
## 40      ENSG00000226742;HSBP1L1;18;79964561;79970822;+
## 41      ENSG00000172216;CEBPB;20;50190734;50192689;+
## 42      ENSG00000175221;MED16;19;867630;893218;-
## 43      ENSG00000065717;TLE2;19;2997638;3047635;-
## 44      ENSG00000129932;DOHH;19;3490821;3500940;-
## 45      ENSG00000105229;PIAS4;19;4007646;4039386;+
## 46      ENSG00000279329;CTD-2553L13.9;19;34675717;34677581;-
## 47      ENSG00000105723;GSK3A;19;42230186;42242625;-
## 48      ENSG00000104783;KCNN4;19;43766533;43781257;-
## 49      ENSG00000196867;ZFP28;19;56538948;56556810;+
## 50      ENSG00000099958;DERL3;22;23834503;23839128;-
## 51      ENSG00000248751;RP1-130H16.18;22;30285238;30299482;-

```

Pas cool...

```
read.table(file = "annotation.csv", sep = ";")
```

##	V1	V2	V3	V4	V5	V6
## 1	id	name	chr	start	stop	strand
## 2	ENSG00000225630	MTND2P28	1	629640	630683	+
## 3	ENSG00000134198	TSPAN2	1	115048011	115089500	-
## 4	ENSG00000116199	FAM20B	1	179025804	179076562	+
## 5	ENSG00000119285	HEATR1	1	236549005	236604504	-
## 6	ENSG00000034510	TMSB10	2	84905625	84906675	+
## 7	ENSG00000198586	TLK1	2	170990823	171231314	-
## 8	ENSG00000157036	EXOG	3	38496127	38542161	+
## 9	ENSG00000157869	RAB28	4	13361354	13484365	-
## 10	ENSG00000250202	RP11-397E7.2	4	86876338	86876652	+

## 11	ENSG00000169570	DTWD2	5	118837322	118988545	-
## 12	ENSG00000269293	ZSCAN16-AS1	6	28121795	28137293	-
## 13	ENSG00000197081	IGF2R	6	159969099	160113507	+
## 14	ENSG00000147274	RBMX	X	136848004	136880764	-
## 15	ENSG00000253991	KB-1562D12.2	8	101528723	101529569	-
## 16	ENSG00000214121	TDPX2	9	12972843	12973438	-
## 17	ENSG00000148090	AUH	9	91213815	91361913	-
## 18	ENSG00000148248	SURF4	9	133361449	133376166	-
## 19	ENSG00000064201	TSPAN32	11	2301997	2318200	+
## 20	ENSG00000183161	FANCF	11	22622519	22626787	-
## 21	ENSG00000121680	PEX16	11	45909669	45918812	-
## 22	ENSG00000254470	AP5B1	11	65775893	65780802	-
## 23	ENSG00000135452	TSPAN31	12	57738013	57750211	+
## 24	ENSG00000251106	FAM206BP	13	46270077	46270617	+
## 25	ENSG00000118939	UCHL3	13	75549480	75606020	+
## 26	ENSG00000238241	CCR12P	13	99407781	99409062	-
## 27	ENSG00000129562	DAD1	14	22564905	22589269	-
## 28	ENSG00000125384	PTGER2	14	52314305	52328606	+
## 29	ENSG00000159433	STARD9	15	42575659	42720981	+
## 30	ENSG00000104164	BLOC1S6	15	45587123	45615999	+
## 31	ENSG00000140416	TPM1	15	63042632	63071915	+
## 32	ENSG00000167005	NUDT21	16	56429133	56452199	-
## 33	ENSG00000185324	CDK10	16	89680737	89696364	+
## 34	ENSG00000161692	DBF4B	17	44708608	44752264	+
## 35	ENSG00000168517	HEXIM2	17	45160700	45170040	+
## 36	ENSG00000262814	MRPL12	17	81703357	81707526	+
## 37	ENSG00000188985	DHFRP1	18	26170726	26171284	-
## 38	ENSG00000267228	IER3IP1	18	47134656	47176281	-
## 39	ENSG00000267699	RP11-729L2.2	18	50968019	51058144	+
## 40	ENSG00000226742	HSBP1L1	18	79964561	79970822	+
## 41	ENSG00000172216	CEBPB	20	50190734	50192689	+
## 42	ENSG00000175221	MED16	19	867630	893218	-
## 43	ENSG00000065717	TLE2	19	2997638	3047635	-
## 44	ENSG00000129932	DOHH	19	3490821	3500940	-
## 45	ENSG00000105229	PIAS4	19	4007646	4039386	+
## 46	ENSG00000279329	CTD-2553L13.9	19	34675717	34677581	-
## 47	ENSG00000105723	GSK3A	19	42230186	42242625	-
## 48	ENSG00000104783	KCNN4	19	43766533	43781257	-
## 49	ENSG00000196867	ZFP28	19	56538948	56556810	+
## 50	ENSG00000099958	DERL3	22	23834503	23839128	-
## 51	ENSG00000248751	RP1-130H16.18	22	30285238	30299482	-

Pas encore parfait.

```
read.table(file = "annotation.csv", sep = ";", header = TRUE)
```

##		id	name	chr	start	stop	strand
## 1	ENSG00000225630	MTND2P28	1	629640	630683	+	
## 2	ENSG00000134198	TSPAN2	1	115048011	115089500	-	
## 3	ENSG00000116199	FAM20B	1	179025804	179076562	+	
## 4	ENSG00000119285	HEATR1	1	236549005	236604504	-	
## 5	ENSG00000034510	TMSB10	2	84905625	84906675	+	
## 6	ENSG00000198586	TLK1	2	170990823	171231314	-	
## 7	ENSG00000157036	EXOG	3	38496127	38542161	+	
## 8	ENSG00000157869	RAB28	4	13361354	13484365	-	
## 9	ENSG00000250202	RP11-397E7.2	4	86876338	86876652	+	
## 10	ENSG00000169570	DTWD2	5	118837322	118988545	-	
## 11	ENSG00000269293	ZSCAN16-AS1	6	28121795	28137293	-	
## 12	ENSG00000197081	IGF2R	6	159969099	160113507	+	
## 13	ENSG00000147274	RBMX	X	136848004	136880764	-	
## 14	ENSG00000253991	KB-1562D12.2	8	101528723	101529569	-	
## 15	ENSG00000214121	TDPX2	9	12972843	12973438	-	
## 16	ENSG00000148090	AUH	9	91213815	91361913	-	
## 17	ENSG00000148248	SURF4	9	133361449	133376166	-	
## 18	ENSG00000064201	TSPAN32	11	2301997	2318200	+	
## 19	ENSG00000183161	FANCF	11	22622519	22626787	-	
## 20	ENSG00000121680	PEX16	11	45909669	45918812	-	
## 21	ENSG00000254470	AP5B1	11	65775893	65780802	-	
## 22	ENSG00000135452	TSPAN31	12	57738013	57750211	+	
## 23	ENSG00000251106	FAM206BP	13	46270077	46270617	+	
## 24	ENSG00000118939	UCHL3	13	75549480	75606020	+	
## 25	ENSG00000238241	CCR12P	13	99407781	99409062	-	
## 26	ENSG00000129562	DAD1	14	22564905	22589269	-	
## 27	ENSG00000125384	PTGER2	14	52314305	52328606	+	
## 28	ENSG00000159433	STARD9	15	42575659	42720981	+	
## 29	ENSG00000104164	BLOC1S6	15	45587123	45615999	+	
## 30	ENSG00000140416	TPM1	15	63042632	63071915	+	
## 31	ENSG00000167005	NUDT21	16	56429133	56452199	-	
## 32	ENSG00000185324	CDK10	16	89680737	89696364	+	
## 33	ENSG00000161692	DBF4B	17	44708608	44752264	+	
## 34	ENSG00000168517	HEXIM2	17	45160700	45170040	+	
## 35	ENSG00000262814	MRPL12	17	81703357	81707526	+	
## 36	ENSG00000188985	DHFRP1	18	26170726	26171284	-	
## 37	ENSG00000267228	IER3IP1	18	47134656	47176281	-	
## 38	ENSG00000267699	RP11-729L2.2	18	50968019	51058144	+	
## 39	ENSG00000226742	HSBP1L1	18	79964561	79970822	+	
## 40	ENSG00000172216	CEBPB	20	50190734	50192689	+	
## 41	ENSG00000175221	MED16	19	867630	893218	-	
## 42	ENSG00000065717	TLE2	19	2997638	3047635	-	

```
## 43 ENSG00000129932      DOHH 19  3490821  3500940  -
## 44 ENSG00000105229      PIAS4 19  4007646  4039386  +
## 45 ENSG00000279329 CTD-2553L13.9 19  34675717  34677581  -
## 46 ENSG00000105723      GSK3A 19  42230186  42242625  -
## 47 ENSG00000104783      KCNN4 19  43766533  43781257  -
## 48 ENSG00000196867      ZFP28 19  56538948  56556810  +
## 49 ENSG00000099958      DERL3 22  23834503  23839128  -
## 50 ENSG00000248751 RP1-130H16.18 22  30285238  30299482  -
```

Parfait !

```
annot <- read.table(file = "annotation.csv", sep = ";", header = TRUE)
```

Vérifier les dimensions

```
dim(annot)
```

```
## [1] 50  6
```

Afficher quelques lignes

```
head(annot)
```

```
##           id      name chr      start      stop strand
## 1 ENSG00000225630 MTND2P28   1    629640    630683      +
## 2 ENSG00000134198  TSPAN2   1 115048011 115089500      -
## 3 ENSG00000116199  FAM20B   1 179025804 179076562      +
## 4 ENSG00000119285  HEATR1   1 236549005 236604504      -
## 5 ENSG00000034510  TMSB10   2  84905625  84906675      +
## 6 ENSG00000198586   TLK1    2 170990823 171231314      -
```

10.2 Combien ?

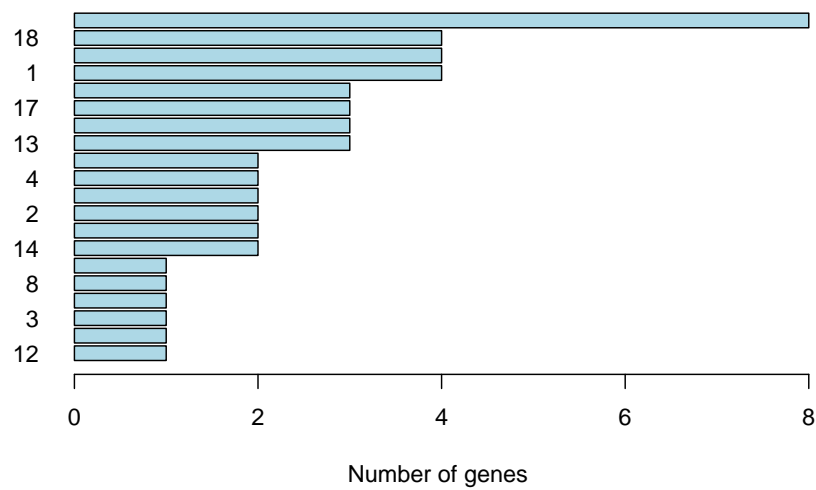
- Combien de gènes sur le chromosome 18 ?
- Combien de gènes sur le chromosome X ?

10.2.1 Solution pour y répondre

```
table(annot$chr)
```

```
##
##  1 11 12 13 14 15 16 17 18 19  2 20 22  3  4  5  6  8  9  X
##  4  4  1  3  2  3  2  3  4  8  2  1  2  1  2  1  2  1  3  1
```

```
barplot(sort(table(annot$chr)), horiz = TRUE, las = 1,
        col = "lightblue", xlab = "Number of genes")
```



10.3 Ma première bioinformatique intégrative

10.3.1 Étape 1 - Fusionner les tableaux d'expressions et d'annotations :

```
?merge
```

```
exprs_annot <- merge(exprs, annot, by = "id")
head(exprs_annot)
```

```
##           id    WT1  WT2    K01  K02  meanWT  meanKO      FC
## 1 ENSG00000034510 235960 94264 202381 91336 165112.0 146858.5 0.8894478
## 2 ENSG00000064201   116   71    64   56    93.5    60.0 0.6417112
```

```
## 3 ENSG000000065717    118    174    124    182    146.0    153.0 1.0479452
## 4 ENSG000000099958    450    655    301    472    552.5    386.5 0.6995475
## 5 ENSG000000104164   4736   5019   4845   4934   4877.5    4889.5 1.0024603
## 6 ENSG000000104783   9002   8623   7720   7142   8812.5    7431.0 0.8432340
##          mean          M          A      name chr      start      stop strand
## 1 155985.25 -0.16901821 17.248576  TMSB10  2  84905625  84906675      +
## 2    76.75 -0.64000386  6.226893  TSPAN32 11  2301997  2318200      +
## 3   149.50  0.06756328  7.223606   TLE2   19  2997638  3047635      -
## 4   469.50 -0.51550605  8.852078   DERL3   22  23834503  23839128      -
## 5  4883.50  0.00354507 12.253699  BLOC1S6 15  45587123  45615999      +
## 6   8121.75 -0.24599498 12.982338   KCNN4 19  43766533  43781257      -
```

10.3.2 Étape 2 - Sous-ensemble des lignes pour lesquelles chr vaut 8 :

```
exprs_chr8 <- exprs_annot[which(exprs_annot$chr == 8), ]
print(exprs_chr8)
```

```
##          id WT1 WT2 K01 K02 meanWT meanK0      FC mean      M
## 44 ENSG000000253991  77  78 134  92   77.5   113 1.458065 95.25 0.5440546
##          A      name chr      start      stop strand
## 44 6.548152 KB-1562D12.2   8 101528723 101529569      -
```

10.3.3 Exporter exprs_chr8 dans un fichier

```
write.table(x = exprs_chr8, file = "exprs_chr8.txt",
  sep = "\t",
  row.names = FALSE,
  col.names = TRUE)
```

10.4 Visualisation

```
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
```

```
##
```

```
##      last_plot
```

```
## The following object is masked from 'package:stats':
```

```
##
```

```
##      filter
```

```
## The following object is masked from 'package:graphics':
```

```
##
```

```
##      layout
```

```
plot_ly(data = exprs_annot, x = ~A, y = ~M, text = paste("Gene name :", exprs_annot$na
```

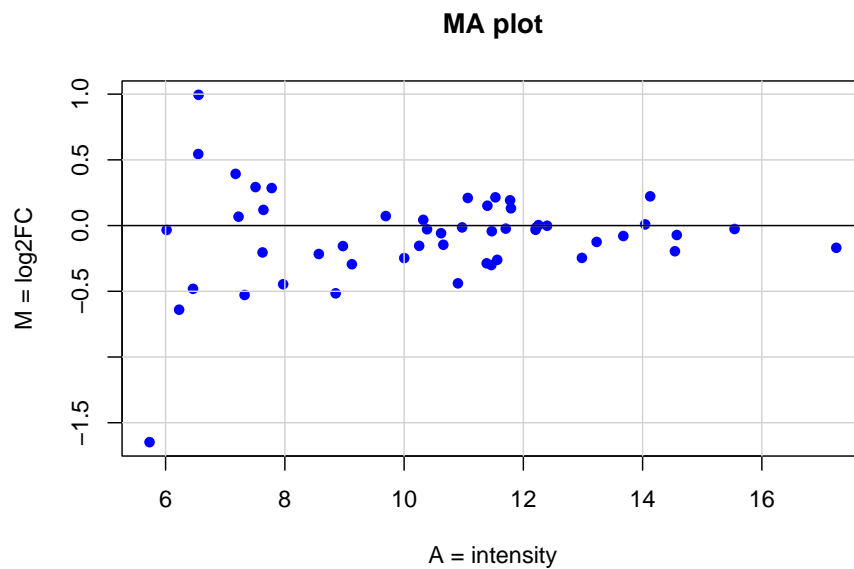
Chapter 11

Bonus

Dans cette partie, nous allons produire un même graphe avec différentes approches.

11.1 R de base

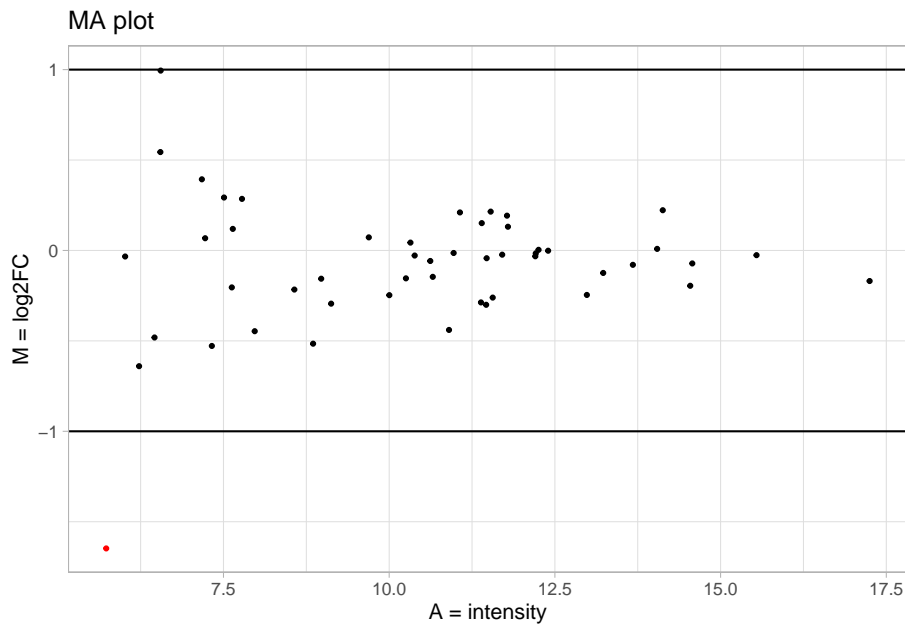
```
plot(x = exprs$A, y = exprs$M, main = "MA plot",  
     col = "blue", pch = 16, xlab = "A = intensity", ylab = "M = log2FC")  
grid(lty = "solid", col = "lightgray")  
abline(h = 0)
```



```
## ggplot2
```

```
library(ggplot2)

g <- ggplot(data = exprs, aes(x = A, y = M)) +
  geom_point(aes(A, M, colour = factor(ifelse(abs(M) <= 1, 1, 2))), size = 0.8) +
  geom_hline(yintercept = c(-1, 1)) +
  scale_color_manual(values = c("black", "red")) +
  ggtitle("MA plot") +
  labs(y = "M = log2FC", x = "A = intensity") +
  theme_light() + theme(legend.position = "none")
g
```

11.2 Plotly

11.2.1 A partir de ggplot2

```
library(plotly)
ggplotly(g)
```

```
## Warning: `gather_()` was deprecated in tidyr 1.2.0.
## i Please use `gather()` instead.
## i The deprecated feature was likely used in the plotly package.
## Please report the issue at <https://github.com/plotly/plotly.R/issues>.
```

11.2.2 En plotly pur

```
plot_ly(data = exprs_annot, x = ~A, y = ~M, text = paste("Gene name :", exprs_annot$name), type =
```

11.3 echarts

```
library(echarts4r)
library(dplyr)

exprs %>%
  mutate(interst = ifelse(abs(M) <= 1, 1, 2))|>
  group_by(interst)|>
  e_charts(A) |>
  e_scatter(M, symbol_size=10) |>
  e_legend(FALSE) |>
  e_tooltip() |>
  e_color(
    c("black", "red")
  ) |>
  e_title("MA plot") |>
  e_axis_labels(y = "M = log2FC", x = "A = intensity") |>
  e_toolbox_feature(feature = "saveAsImage") |>
  e_toolbox_feature(feature = "dataZoom") |>
  e_toolbox_feature(feature = "dataView")
```

Chapter 12

Conclusion

12.1 Take home messages

- Tout est faisable avec R
- **Définir et comprendre l'opération mathématique/statistique** avant de chercher la fonction R correspondante
- R est un langage :
 - plusieurs types et structures de données (out of scope)
 - énormément de commandes à découvrir (out of scope)
 - Google est votre ami
- Une infinité de :
 - ressources en ligne
 - tutoriels pour des analyses spécifiques (e.g. DESeq2 pour le RNA-Seq)

Bonnes pratiques : <https://style.tidyverse.org/syntax.html>

12.2 Ressources IFB

- Serveur RStudio: <https://rstudio.cluster.france-bioinformatique.fr/>
- Jupyter lab (inclut un serveur RStudio et plein d'autres choses : <http://jupyterhub.cluster.france-bioinformatique.fr/>)
- Une question ? Un besoin ? Un problème ? Contactez la communauté IFB : <https://community.france-bioinformatique.fr/>

12.3 Resource

- R : <https://www.r-project.org/>
- RStudio : <https://rstudio.com/>
- R bloggers : <https://www.r-bloggers.com/>
- Thinkr : <https://thinkr.fr/>
- Rstudio Cheatsheets (un tas de thèmes): <https://rstudio.com/resources/cheatsheets/>

