

(Re)découverte de R... en 1h45

Ecole de Bioinformatique AVIESAN-IFB – Roscoff – Octobre 2020

Thomas Denecker – thomas.denecker@france-bioinformatique.fr

Jacques van Helden – jacques.van-helden@univ-amu.fr

Stevonn Volant - stevonn.volant@pasteur.fr

Slides d'Hugo Varet – hugo.varet@pasteur.fr

aviesan

alliance nationale
pour les sciences de la vie et de la santé



CNRS UPMC

**Station Biologique
Roscoff**

R en quelques mots

Langage de programmation qui permet de :

1. manipuler des données : importer, transformer, exporter
2. faire des analyses statistiques plus ou moins complexes : description, exploration, modélisation...
3. créer des (jolies) figures

Disponible sur [RCRAN](https://www.R-project.org/)



Historique :

- 1993 : début du projet R
- 2000 : sortie de R 1.0.0
- 2020 : R 4.0.2

Avantages et inconvénients

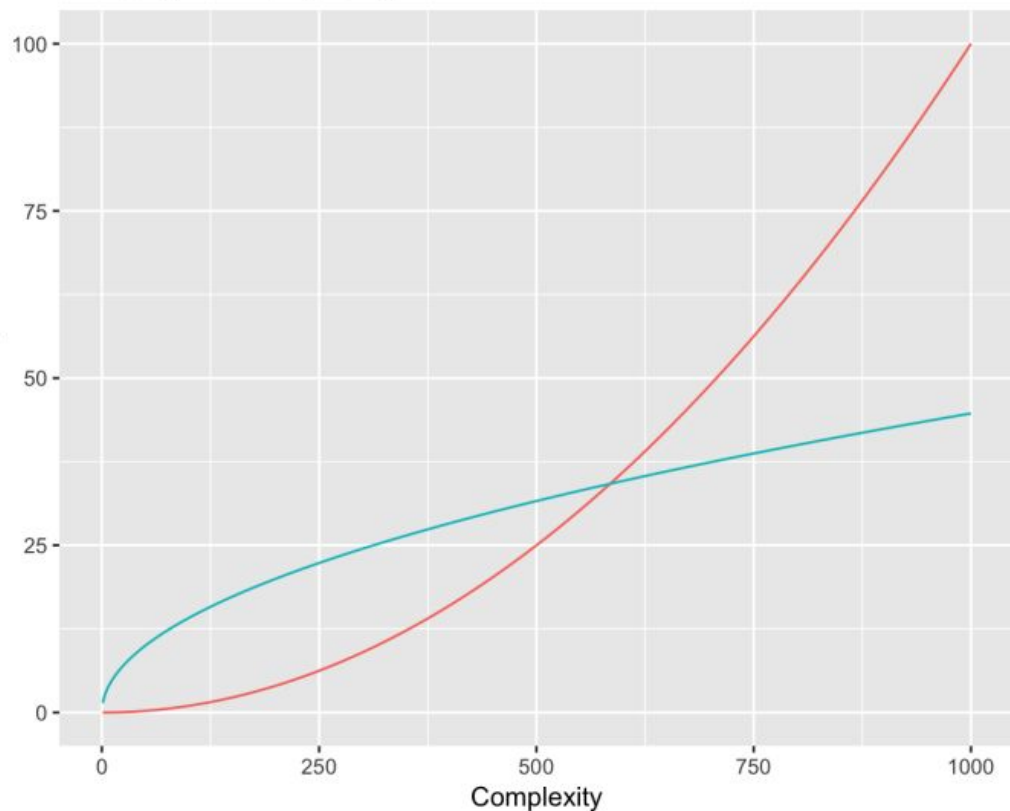
Avantages :

- 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

Inconvénients :

R vs Excel

Difficulty vs. Complexity



tool
— Excel
— R

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](#)

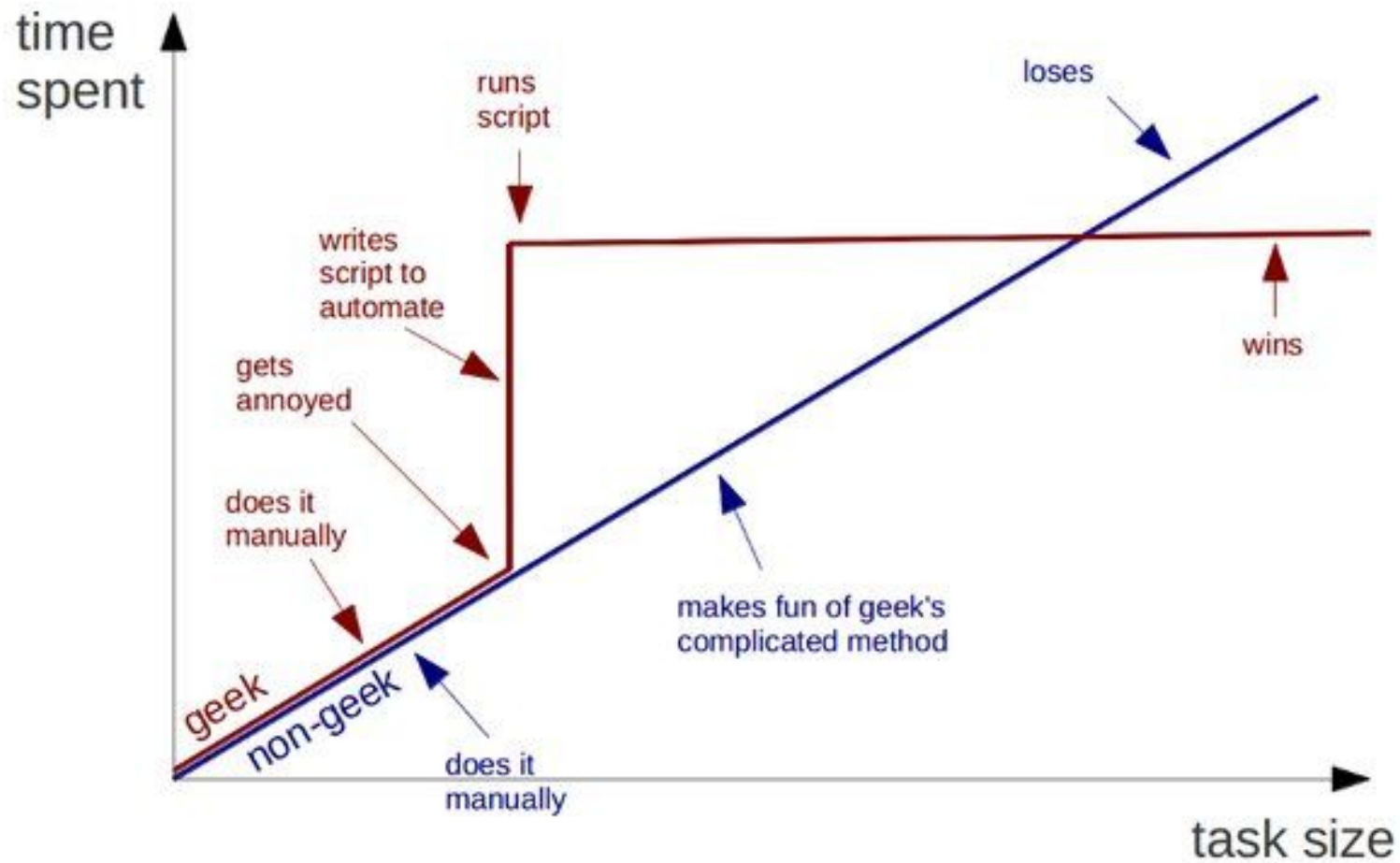


[Alexandre Counis, Les Echos, 5 oct. 2020](#)

Source: R-bloggers

Geeks and repetitive tasks

Geeks and repetitive tasks



R sait tout faire

Lire un tableau de données	<code>read.table()</code>
Fusionner deux tableaux	<code>merge()</code>
Filtrer des lignes	<code>data[data\$x > 10]</code>
Sélectionner des colonnes	<code>data[,c("x", "y")]</code>
Rechercher une chaîne de caractères	<code>grep()</code>
Calculer une moyenne	<code>mean()</code>
Réaliser une ACP	<code>prcomp()</code>
Additionner deux matrices	<code>mat1 + mat2</code>
Exporter un tableau de données	<code>write.table()</code>
Calculer une variance	<code>var()</code>
Régression linéaire	<code>lm()</code>
Tracer une courbe	<code>plot()</code>
Tester une hypothèse	<code>t.test()</code>
Dessiner un histogramme	<code>hist()</code>
Convertir des données	<code>as.matrix()</code>

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

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



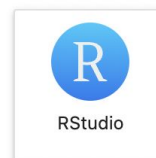
A screenshot of the RStudio sign-in interface. The title is "Sign in to RStudio". It features two input fields: "Username:" and "Password:". Below the password field is a checkbox labeled "Stay signed in". At the bottom center is a blue button labeled "Sign In".



3. Vous connecter via **Jupyter lab de l'IFB**

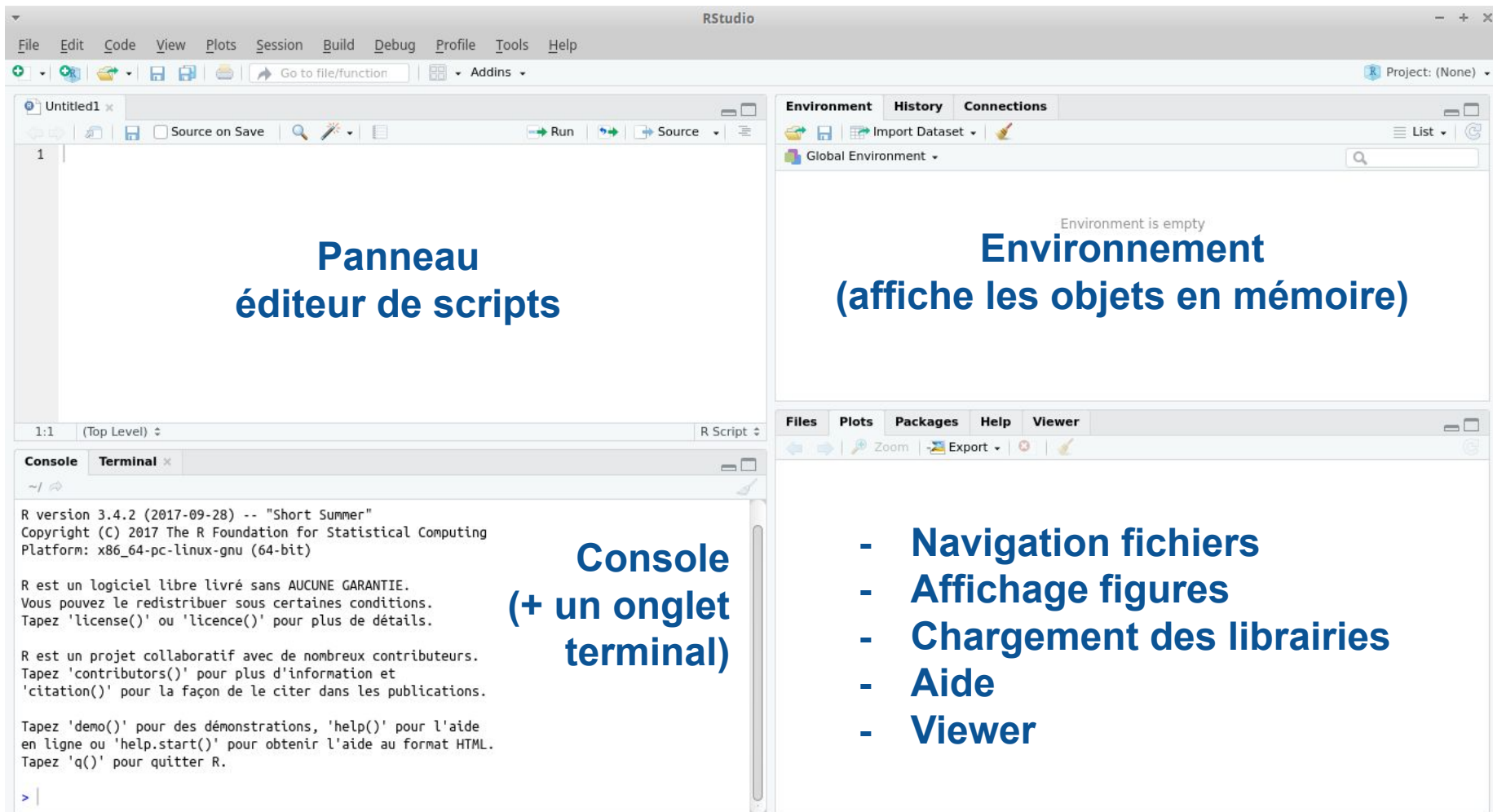
<https://jupyterhub.cluster.france-bioinformatique.fr>

puis cliquer sur l'icône RStudio



RStudio

- Disponible depuis 2011
- Logiciel facilitant l'utilisation de R via 4 panneaux
- Chaque panneau présente plusieurs onglets (fonctionnalités complémentaires)



R sait tout faire : il compte

Tapez les commandes suivantes dans le panneau Console de RStudio

```
2 + 3
```

```
4 * 5
```

```
6 / 4
```

```
1:10
```

```
8:-9
```

```
1,2
```

```
1.2
```

Notion de variable/objet

```
a <- 2      ## Créer une variable nommée a et lui assigner une valeur
print(a)   ## Afficher la valeur de la variable a
a          ## Même résultat: si on évoque le nom de variable, R l'imprime
```

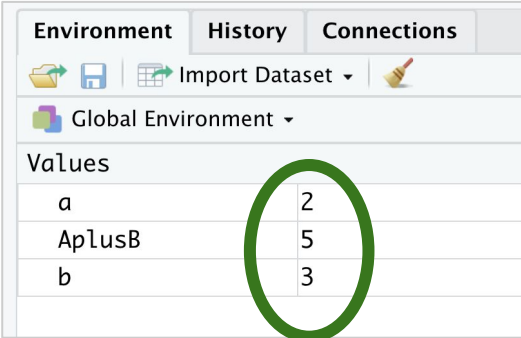
```
b <- 3      ## Assigner une valeur à une seconde variable
a_plus_b <- a + b  ## Effectuer un calcul avec 2 variables
print(a_plus_b)  ## Afficher le contenu de la variable a_plus_b
```

```
a <- 7      ## Changer la valeur de a
print(a_plus_b)  ## Note: le contenu de a_plus_b n'est pas modifié
```

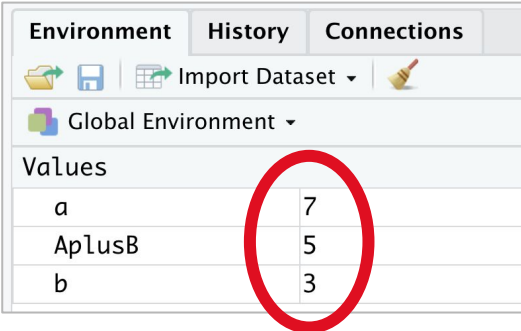
```
a_plus_b <- a + b  ## On recalcule a_plus_b
print(a_plus_b)   ## La nouvelle valeur tient compte de la modification de a
```

```
vec1 <- c(1,10)   ## Créer un vecteur
vec2 <- 1:10      ## Créer un vecteur contenant une séquence d'entiers de 1 à 10
vec2 + a         ## Somme d'un vecteur et d'un nombre
vec3 <- c("riri", "fifi", "loulou")  ## Vecteur de chaînes de caractères
vec2 / 2        ## Diviser un vecteur de nombres par un nombre
vec3 / 2        ## Diviser des chaînes de caractères par un nombre
```

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



Environment	History	Connections
Global Environment	Import Dataset	
Values		
a	2	
a_plus_b	5	
b	3	



Environment	History	Connections
Global Environment	Import Dataset	
Values		
a	7	
a_plus_b	5	
b	3	

Cas pratique : données d'expression

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

The screenshot shows the RStudio interface. The terminal window displays the R version 3.4.4 (2018-03-15) startup message. The Files pane shows the current directory structure, with the 'New Folder' button circled in red. A red arrow points from the 'New Folder' button to a dialog box.

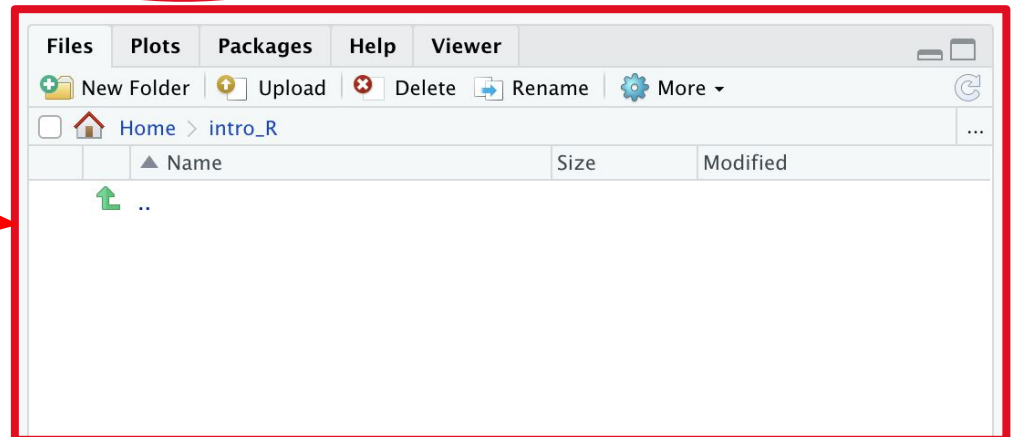
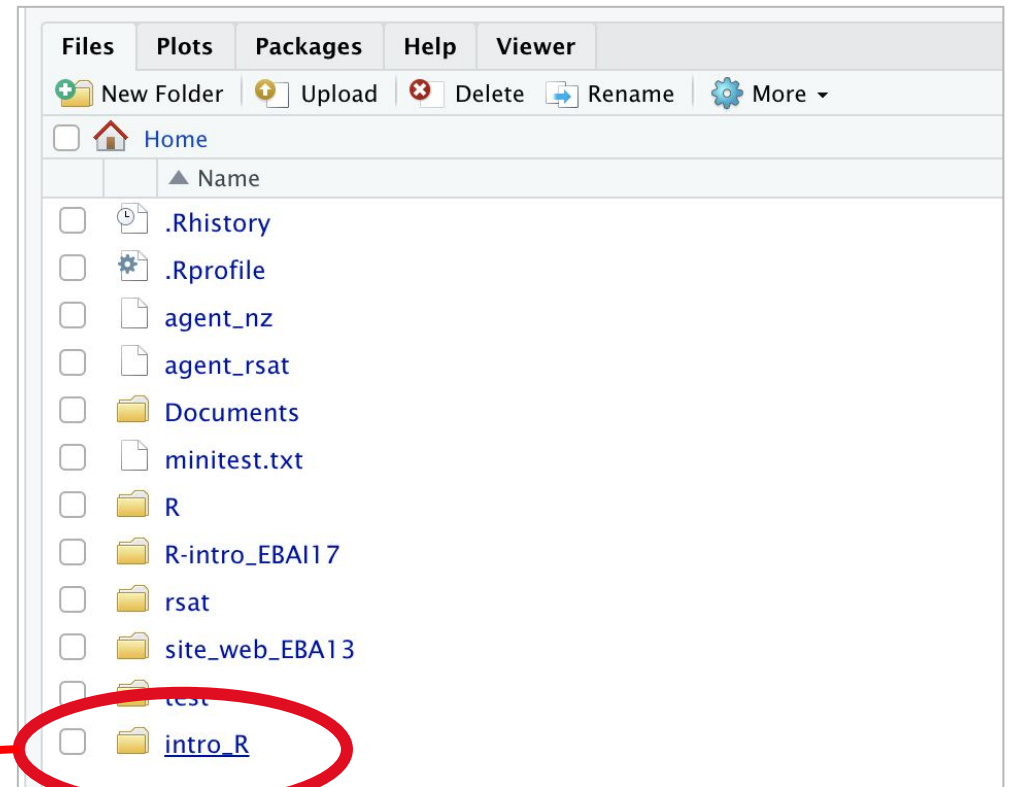
The 'New Folder' dialog box is shown, with the text 'Please enter the new folder name' and the input field containing 'intro_R'. The 'OK' and 'Cancel' buttons are visible at the bottom.

The screenshot shows the RStudio Files pane after the folder creation. The path is '/ > shared > ifbstor1 > home > jvanhelden'. The 'intro_R' folder is listed in the file list.

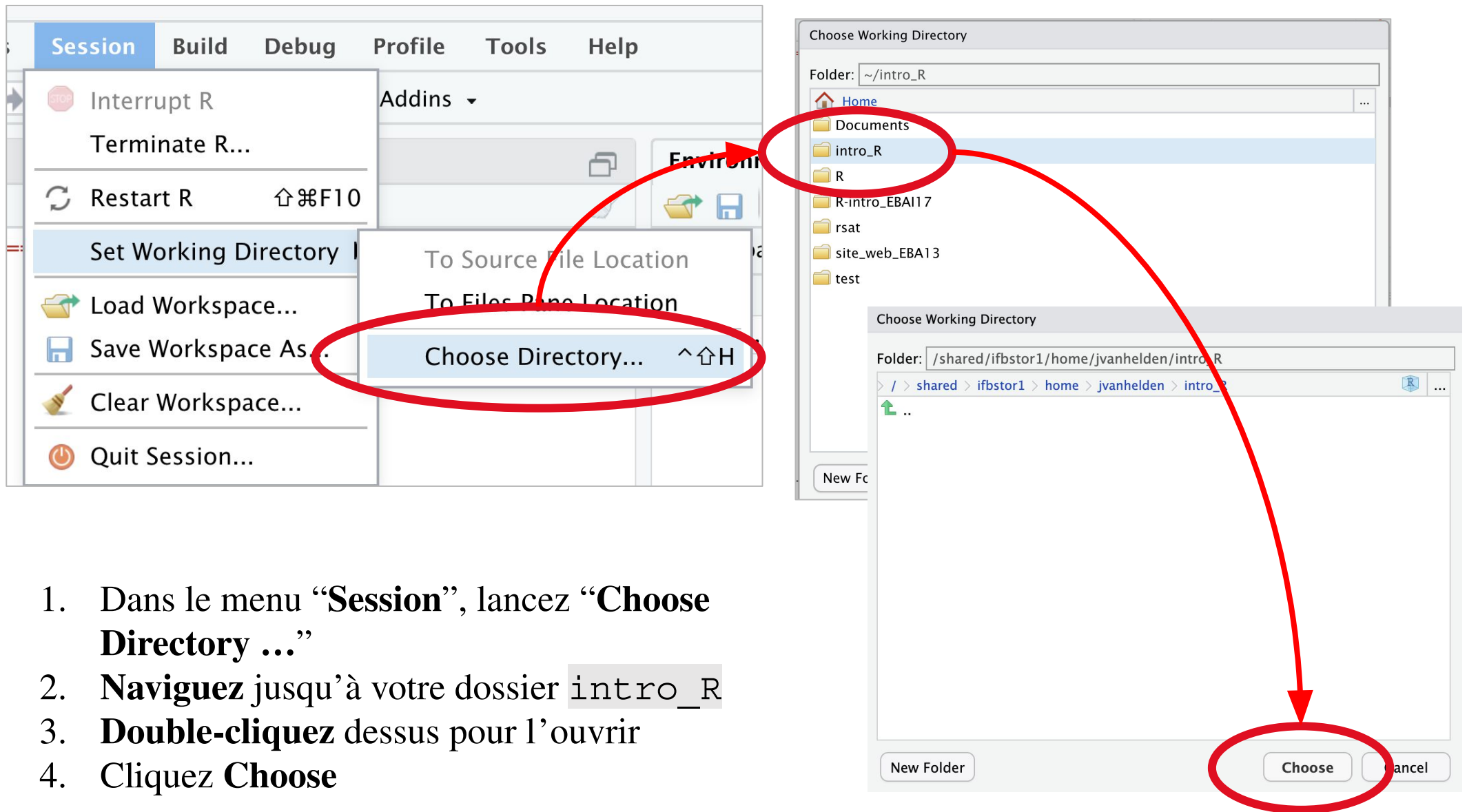
Déplacement dans le dossier “intro_R”

Double-cliquez sur le dossier “intro_R”, pour vous y déplacer.

Puisque vous venez de créer le dossier il est vide (image du bas).



Définissez votre dossier espace de travail (working directory)



The image illustrates the steps to set a working directory in RStudio. It shows the 'Session' menu with 'Choose Directory...' highlighted. Two dialog boxes are shown: the first shows the 'intro_R' folder selected, and the second shows the full path '/shared/ibfstor1/home/jvanhelden/intro_R' and the 'Choose' button highlighted.

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**

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)

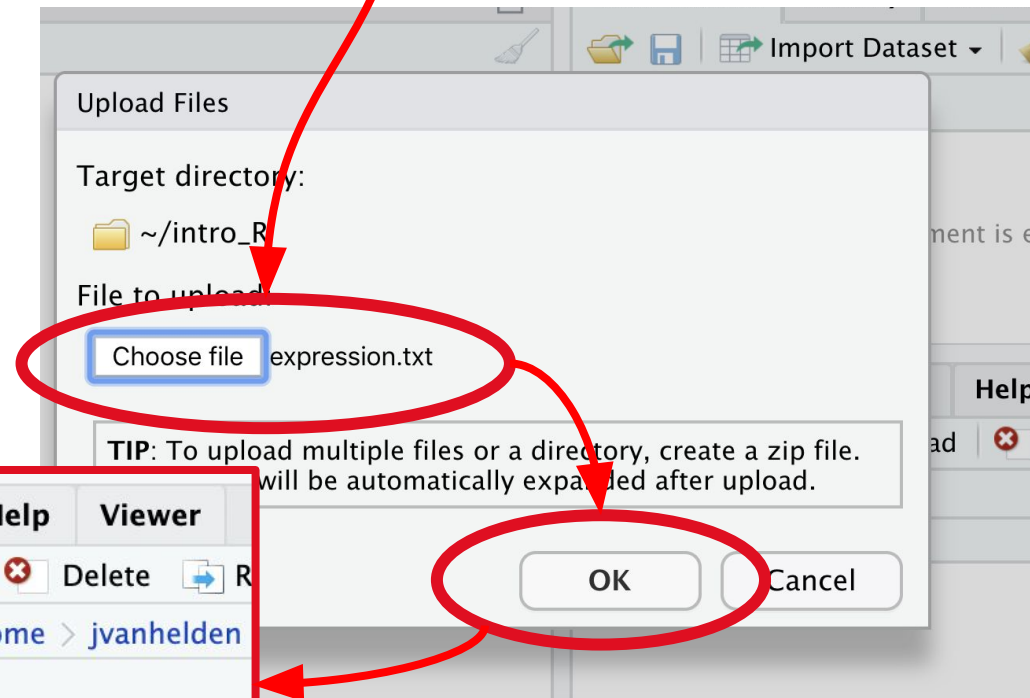
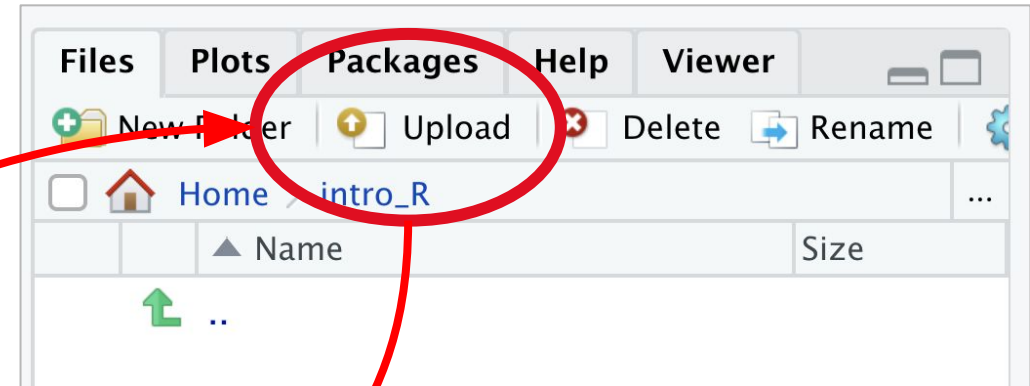
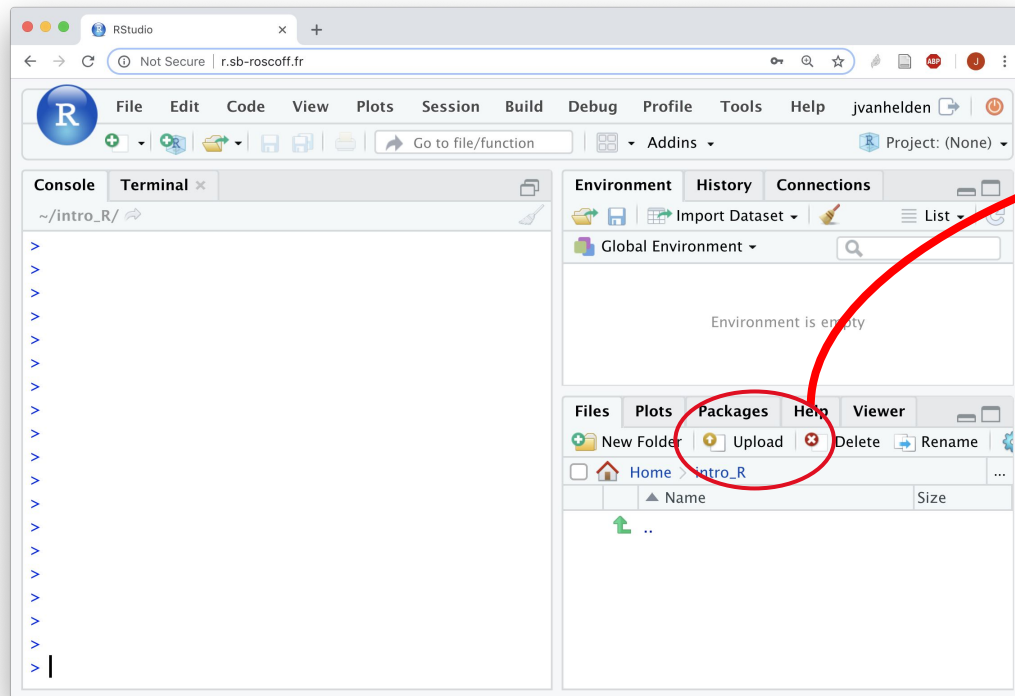
<https://tinyurl.com/ebaii-expression-txt>

<https://tinyurl.com/ebaii21-annotation-csv>

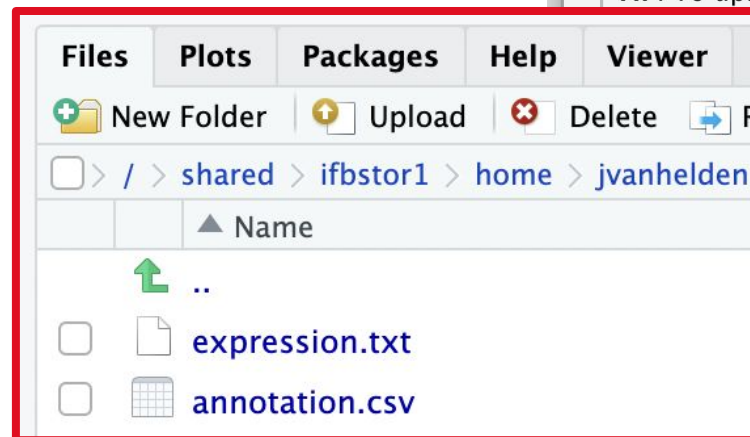
Attention: veillez à sauvegarder les fichiers

- 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)

Téléversement (“upload”) des données



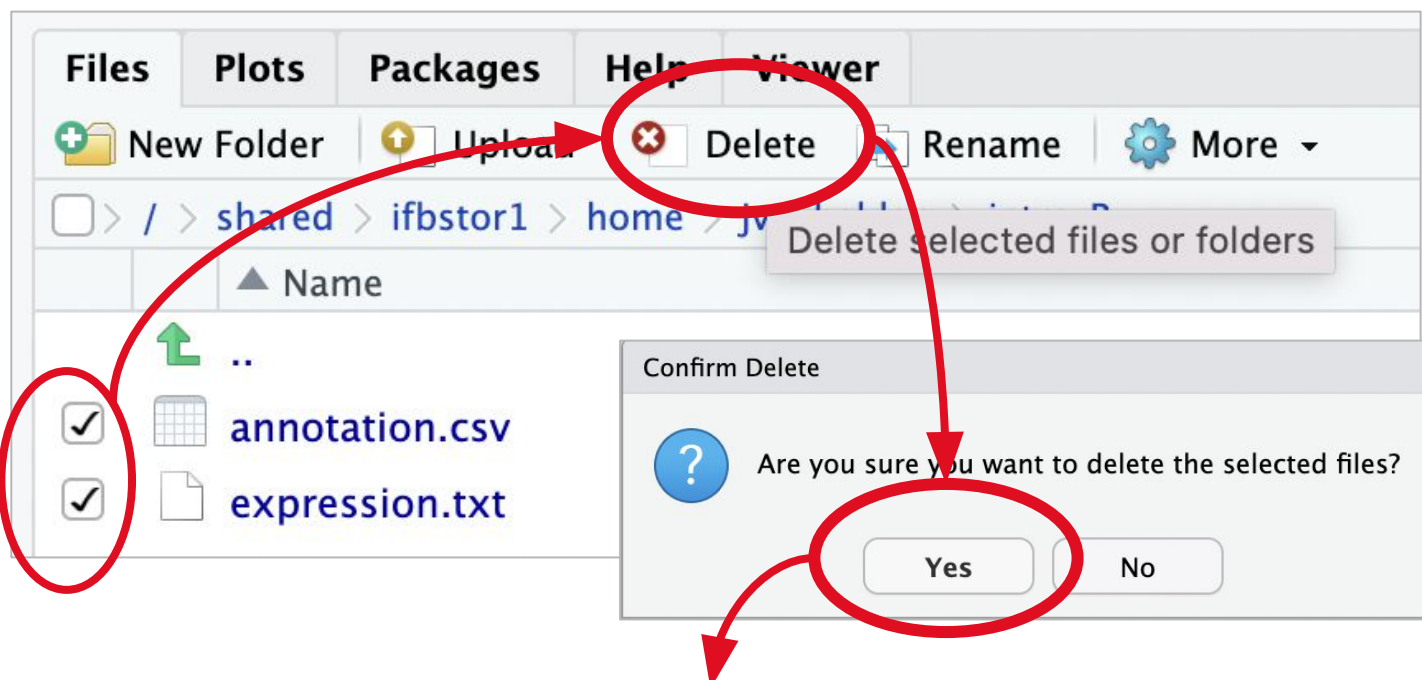
Au moyen du bouton **“Upload”**, téléversez les fichiers d’expression et d’annotation depuis votre ordinateur vers votre compte sur le serveur.



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)



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



Revenir à la maison (note: R interprète le caractère “~” comme le “HOME” de Linux; et cela marche aussi pour Windows!)

```
setwd("~/")
```

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

```
my_work_dir <- "~/intro_R"
```

S’il n’existe pas encore, créer le dossier de travail. (Commande Unix équivalente: "mkdir -p ~/intro_R")

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

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

```
getwd()
```

Aller dans ce dossier de travail (Commande Unix équivalente: "cd ~/intro_R")

```
setwd(my_work_dir)
```

Et maintenant, où suis-je ?

```
getwd()
```

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

```
list.files()
```

```
dir() ## Un autre nom pour la même commande
```

Télécharger un fichier : the “geek” way (V2)

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://tinyurl.com/ebaii-expression-txt", destfile = "expression.txt")
```

```
download.file(url = "https://tinyurl.com/ebaii21-annotation-csv", destfile = "annotation.csv")
```

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

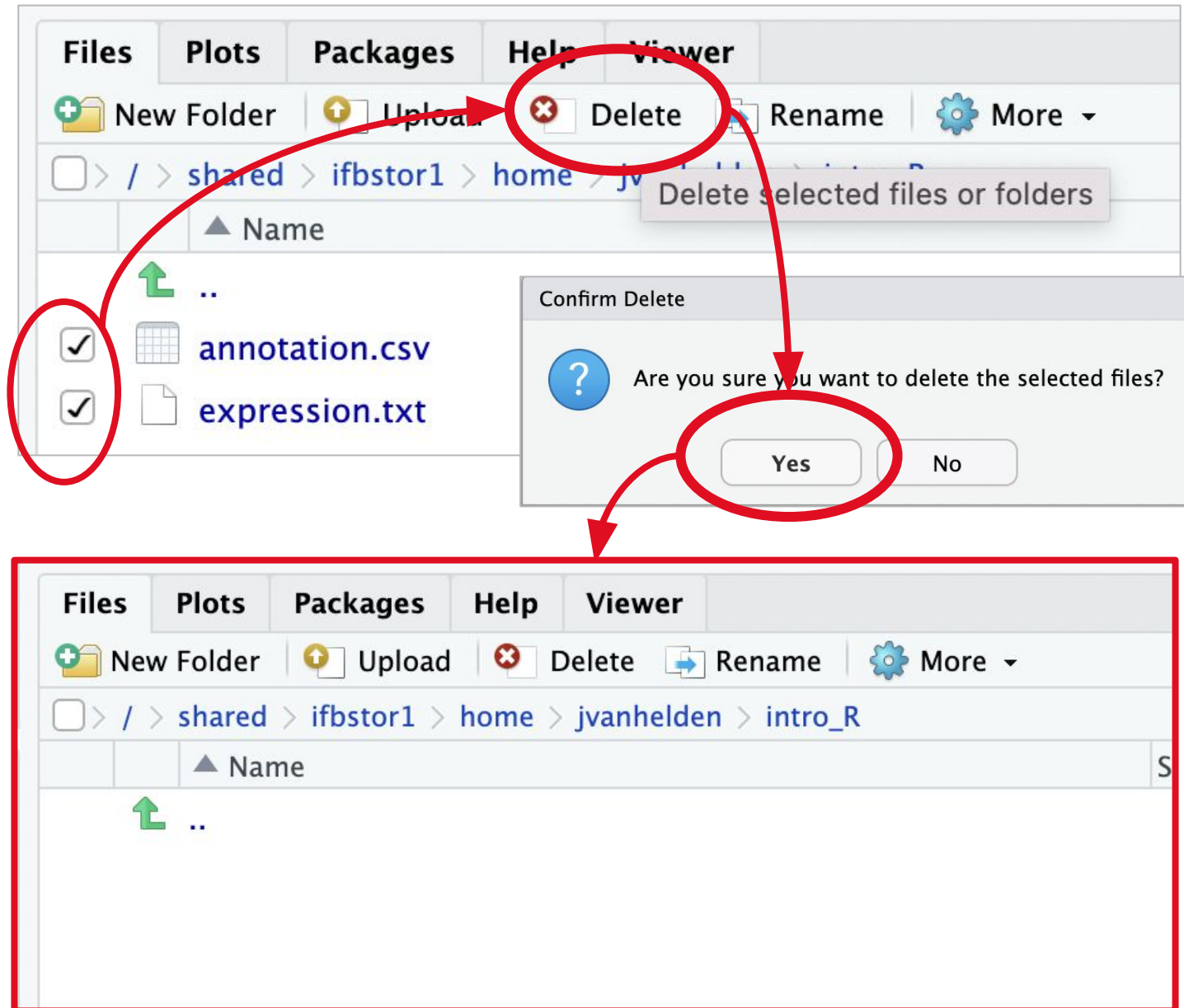
Qu'y a-t-il par ici ? (Commande Unix équivalente: "ls")

```
list.files()
```

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)



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

Alternative: 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
```

Créez un répertoire “intro_R”

```
mkdir -p ~/intro_R
```

Déplacez-vous dans votre dossier

```
cd ~/intro_R
```

Où suis-je maintenant ?

```
pwd
```

Téléchargez les données

```
wget https://tinyurl.com/ebaii-expression-txt --output-document=expression.txt
```

```
wget https://tinyurl.com/ebaii21-annotation-csv -O annotation.csv
```

Qu’y a-t-il ici ?

```
ls -l
```

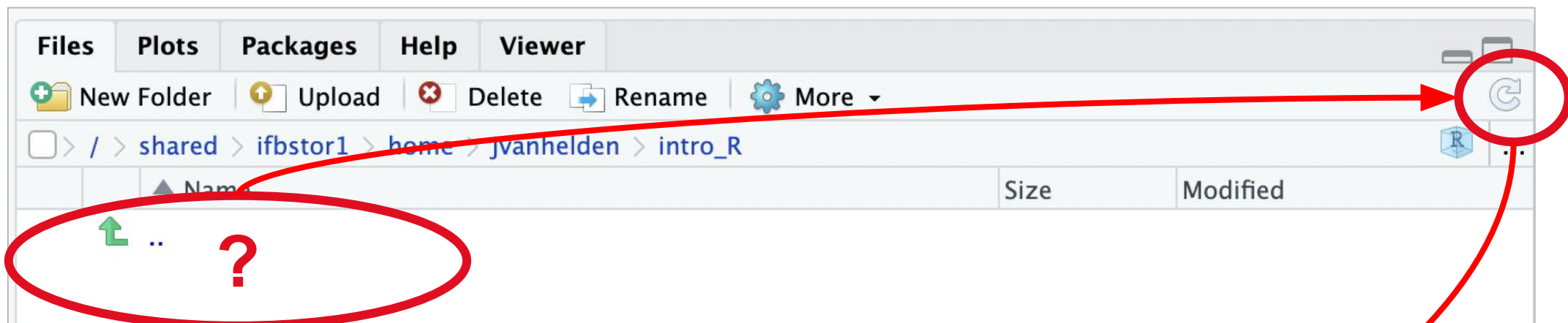
A quoi ressemblent ces fichiers ?

```
less expression.txt
```

```
less annotation.csv
```

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.



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

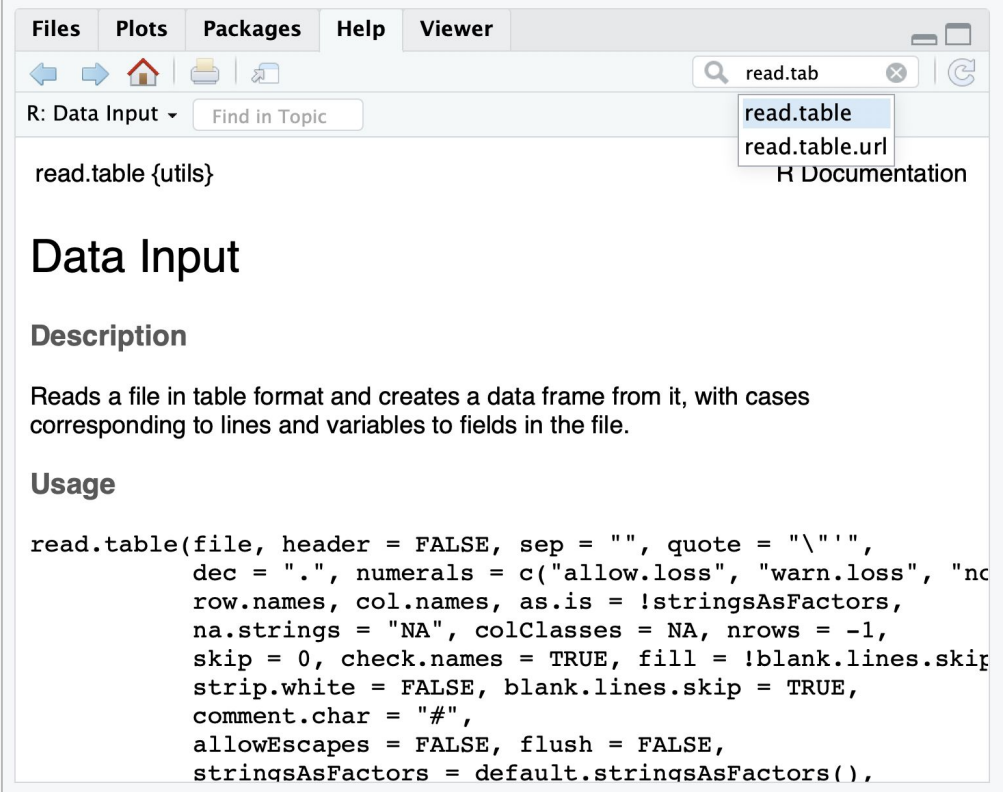
```
help(read.table)
```

Notation alternative

```
?read.table
```

Approche pratique :

1. demande à Google “*Comment lire une table en R ?*”
2. adapte l'exemple



The screenshot shows the RStudio Help viewer interface. The search bar at the top contains "read.tab" and a dropdown menu shows "read.table" and "read.table.url". The main content area displays the documentation for "read.table {utils}" under the "Data Input" section. It includes a "Description" and a "Usage" section with the following code snippet:

```
read.table(file, header = FALSE, sep = "", quote = "\"'",  
           dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),  
           row.names = NULL, as.is = !stringsAsFactors,  
           na.strings = "NA", colClasses = NA, nrows = -1,  
           skip = 0, check.names = TRUE, fill = !blank.lines.skip,  
           strip.white = FALSE, blank.lines.skip = TRUE,  
           comment.char = "#",  
           allowEscapes = FALSE, flush = FALSE,  
           stringsAsFactors = default.stringsAsFactors(),
```

Recherche interactive sous RStudio

- Sélectionner l'onglet "Help" du panneau inférieur droit.
- Taper "read.table" dans la boîte de recherche.

Affichage de l'objet "exprs"

Imprimer toutes les valeurs.

`print(exprs)`

Affichage des premières lignes de l'objet

`head(exprs)`

Affichage des dernières lignes de l'objet

`tail(exprs)`

Un peu plus de lignes

`head(exprs, n = 15)`

Explorer le tableau dans un panneau de visualisation

`View(exprs)`

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

The image displays two screenshots of the RStudio interface, illustrating the visualization of the 'exprs' data frame. The top screenshot shows the first five rows of the data frame, with the column headers 'id', 'WT1', 'WT2', 'KO1', and 'KO2' visible. The bottom screenshot shows the first 14 rows, with the same column headers. A red circle highlights the 'WT1' and 'WT2' column headers in both screenshots, and a red arrow points from the top circle to the bottom one, indicating a scroll action.

	id	WT1	WT2	KO1	KO2
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
21	ENSG00000148218	62324	33973	56862	37710
26	ENSG00000167005	26866	1111	24888	22661
19	ENSG00000147274	16013	17642	15055	18804
11	ENSG00000119235	15783	17359	18591	20077
46	ENSG00000262814	15470	11450	11656	13821
35	ENSG00000197081	14741	36309	14941	29645
14	ENSG00000129562	12089	7958	10708	7683

Caractéristiques d'un tableau de données

Dimensions

<code>ncol(exprs)</code>	<code>## Nombre de colonnes</code>
<code>nrow(exprs)</code>	<code>## Nombre de lignes</code>
<code>dim(exprs)</code>	<code>## Dimensions</code>

Noms des colonnes et des lignes

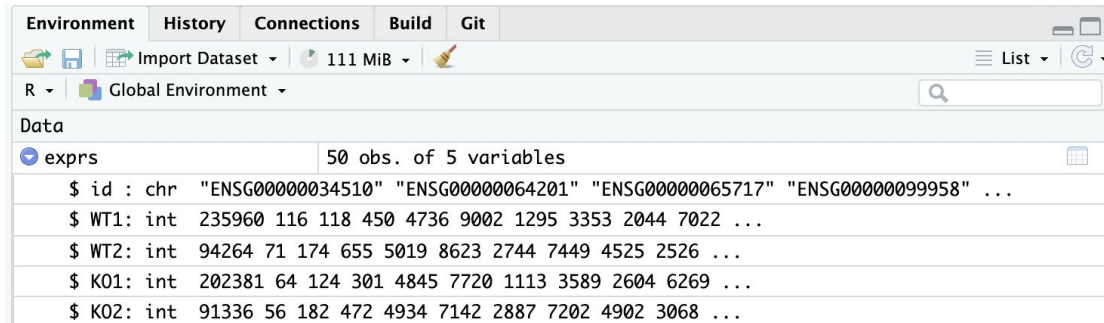
<code>colnames(exprs)</code>	<code>## Noms des colonnes</code>
<code>names(exprs)</code>	<code>## idem</code>
<code>rownames(exprs)</code>	<code>## noms des lignes</code>

Résumé rapide des données par colonne

`summary(exprs)` `## Statistiques par colonne`

`str(exprs)` `## Structure de la variable`

Même résultat que dans le panneau “Environment”



The screenshot shows the RStudio Environment pane. At the top, there are tabs for Environment, History, Connections, Build, and Git. Below the tabs, there are icons for Import Dataset, a memory usage indicator (111 MiB), and a search icon. The Environment pane shows a data frame named 'exprs' with 50 observations and 5 variables. The variables are: id (chr), WT1 (int), WT2 (int), K01 (int), and K02 (int). The first row of data is displayed as follows:

\$ 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	...

Sélection de colonnes d'un tableau

Afficher les noms des colonnes

```
colnames(exprs)
```

Valeurs stockées dans la colonne nommée "WT1"

```
exprs$WT1
```

Notation alternative

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

Sélection de plusieurs colonnes.

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

Sélection de colonnes par leur indice

```
exprs[, 2]
```

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

Histogrammes

Histogramme des valeurs d'expression pour l'échantillon WT1

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

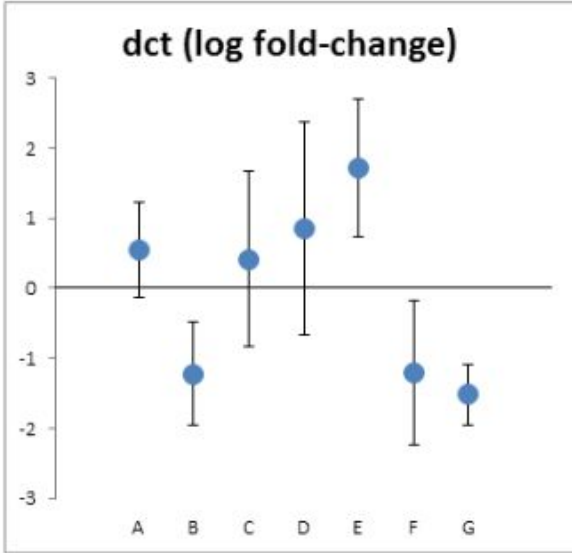
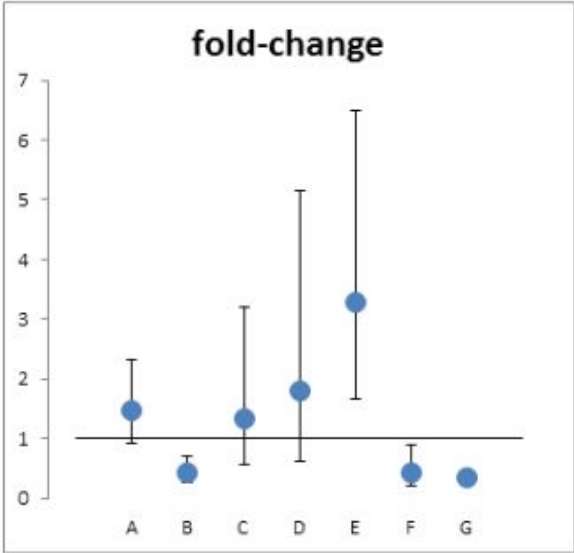
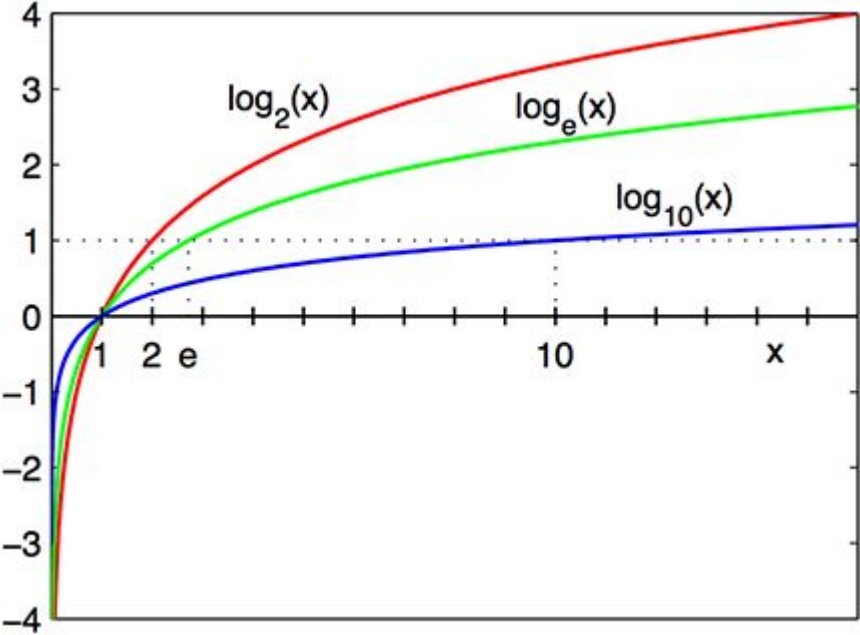
Histogramme du logarithme de ces valeurs

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

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

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

Logarithmes (pour rappel...)



Boîtes à moustaches (boxplots)

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)
```

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

```
## ignorer la première colonne
```

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

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

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

```
## afficher les étiquettes des axes horizontalement
```

Nuage de points

Expressions KO1 vs WT1

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

Personnalisation des paramètres graphiques

```
plot(x = log(exprs$WT1),  
     y = log(exprs$KO1),  
     main = "Expression KO1 vs WT1",  
     xlab = "WT1",  
     ylab = "KO1",  
     pch = 16,  
     las = 1,  
     col = "red")  
grid()  
abline(a = 0, b = 1)
```

```
## données pour l'abscisse  
## données pour l'ordonnée  
## Titre principal  
## légende de l'axe X  
## légende de l'axe Y  
## caractère pour marquer les points  
## écrire les échelles horizontalement  
## couleur des points  
## ajout d'une grille  
## ajouter la droite X = Y (intercept a = 0, pente b = 1).
```

Sélection de lignes d'un tableau

Sélection des lignes 4 et 11 du tableau des expressions

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

Sélection des identifiants de deux gènes d'intérêt

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

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

```
exprs$id %in% mygenes
```

Indices des lignes correspondant aux IDs des gènes d'intérêt

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

Afficher les lignes correspondantes

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

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

```
## formulation plus intuitive
```

Approche plus moderne, avec le package dplyr

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

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

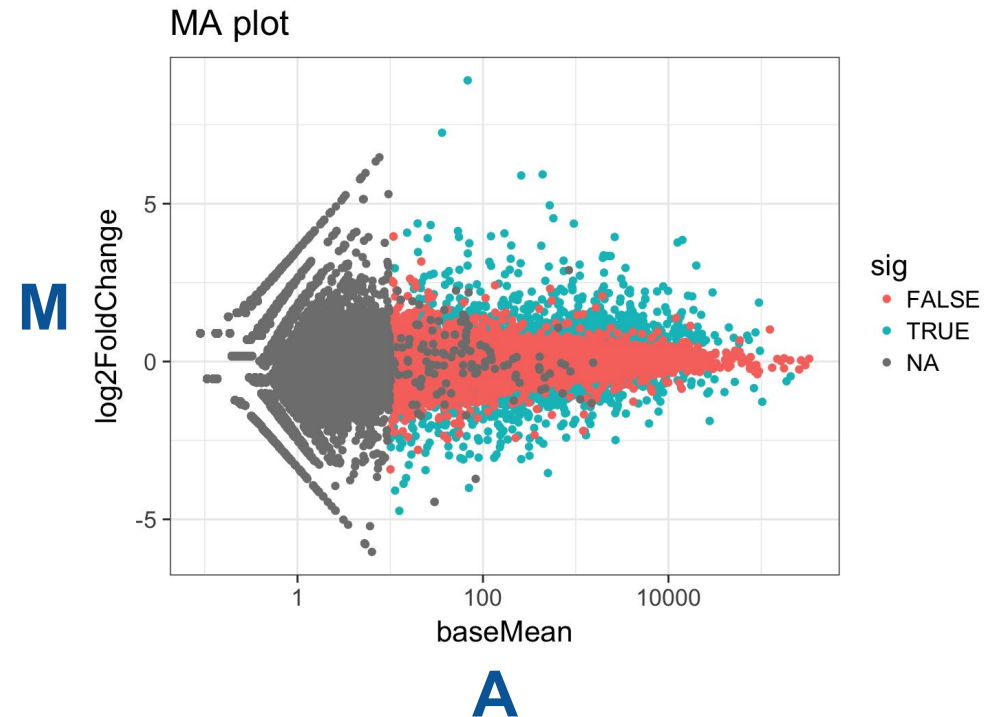
```
exprs %>% filter(id %in% mygenes) %>% mutate(mean_KO = (KO1 + KO2)/2)
```

```
## plus avancé : enchaîner plusieurs commandes
```


Analyse d'expression différentielle : MA-plot

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

	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



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.

$$M = \log_2(KO/WT) = \log_2(KO) - \log_2(WT) \quad \text{log2 fold change, “magnitude”}$$

$$A = \frac{1}{2} \log_2(KO \times WT) = \frac{1}{2} (\log_2(KO) + \log_2(WT)) \quad \text{average log2 value}$$

Calculs sur les colonnes

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

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

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

```
exprs$meanWT <- rowMeans(exprs[ , c("WT1","WT2")])  
exprs$meanKO <- rowMeans(exprs[ , c("KO1","KO2")])  
head(exprs) ## Check the result
```

Fold-change KO vs WT

```
exprs$FC <- exprs$meanKO / exprs$meanWT  
head(exprs) ## Check the result
```

Moyenne de tous les échantillons

```
exprs$mean <- rowMeans(exprs[ , c("WT1", "WT2", "KO1", "KO2")])  
head(exprs) ## Check the result
```

M = $\log_2(\text{KO}/\text{WT}) = \log_2(\text{KO}) - \log_2(\text{WT})$ \log_2 fold change, “magnitude”

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

MA-plot : \log_2FC vs intensité

Calcul de M et A

```
exprs$M <- log2(exprs$FC)
```

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

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)
```

M = $\log_2(KO/WT) = \log_2(KO) - \log_2(WT)$ \log_2 fold change, "magnitude"

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

Appliquer une fonction sur les lignes/colonnes

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)
```

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

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

```
mean_per_col <- apply(exprs[ , c("WT1", "WT2", "KO1", "KO2")], 2, mean)
```

```
var_per_col <- apply(exprs[ , c("WT1", "WT2", "KO1", "KO2")], 2, var)
```

Charger les annotations des gènes

```
read.table(file = "annotation.csv")           ## pas cool  
read.table(file = "annotation.csv", sep = ";") ## pas parfait  
read.table(file = "annotation.csv", sep = ";", header = TRUE) ## OK
```

```
annot <- read.table(file = "annotation.csv", sep = ";", header = TRUE)  
dim(annot)           ## Vérifier les dimensions  
head(annot)          ## Afficher quelques lignes
```

Combien de gènes par chromosome ?

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

Question : combien de gènes sur le chromosome 8 ? Et sur le X ?

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

Ma première bioinformatique intégrative

- 1ere étape : fusionner les tableaux d'expressions et d'annotations :

?merge

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

- 2eme étape : sous-ensemble des lignes pour lesquelles chr vaut 8 :

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

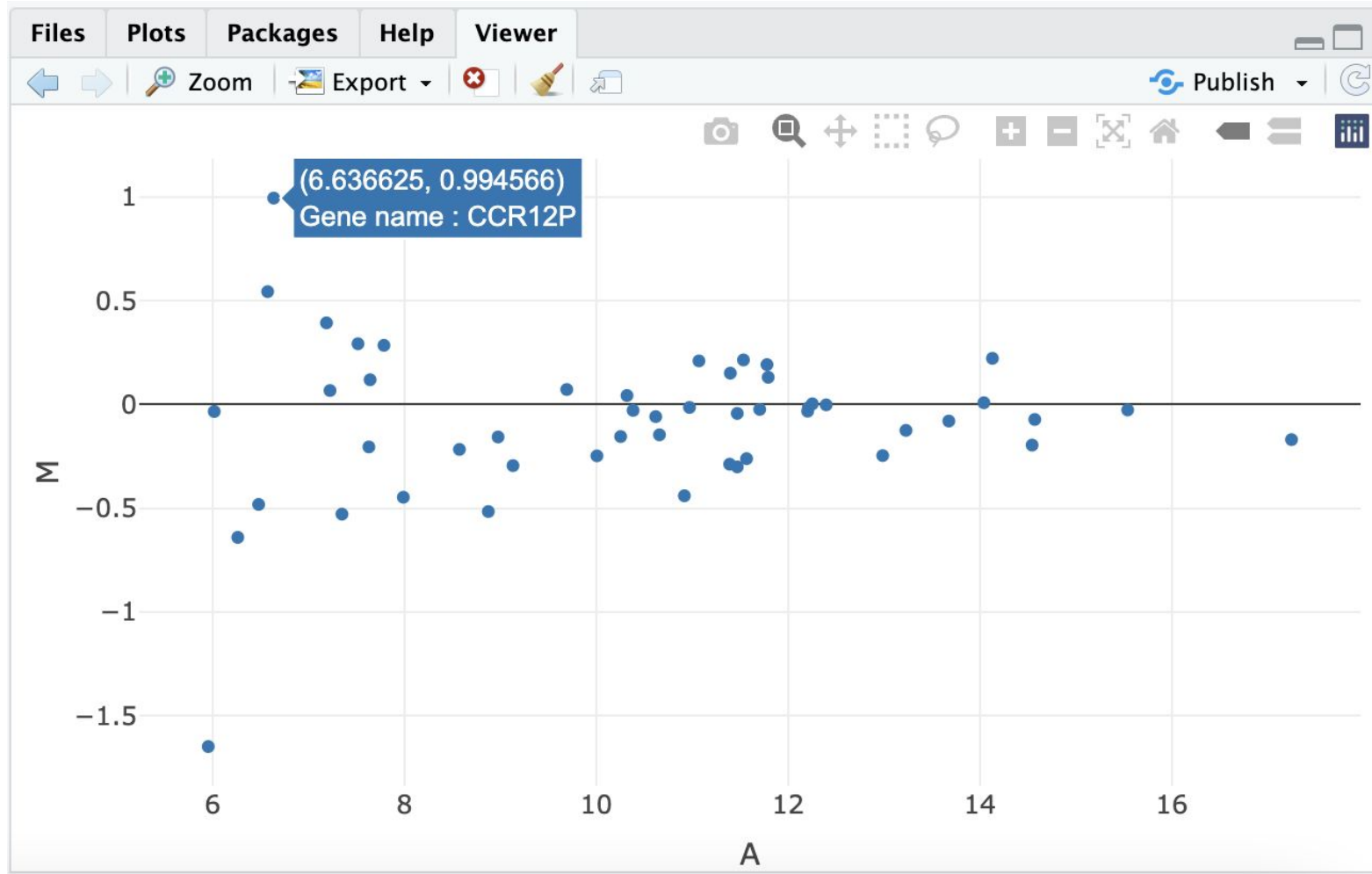
- Exporter exprs_chr8 dans un fichier :

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

Mon premier graphique dynamique

```
library(plotly)
```

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



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>

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/>



Ressources

Base R Cheat Sheet

Getting Help

Advanced R Cheat Sheet

RStudio IDE :: CHEAT SHEET

R Markdown :: CHEAT SHEET

What is R Markdown?

Workflow

render

Embed code with knitr syntax

Interactive Documents

GLOBAL OPTIONS

IMPORTANT CHUNK OPTIONS

Session

Environment

EM

ct

History

with a

shunt

age

Delete

from

Library

pane

code

data set

arch

value

of 2016-01

R Studio

RStudio® is a trademark of RStudio, Inc. • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at rmarkdown.rstudio.com • rmarkdown 1.6 • Updated: 2016-02

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/>

