

1 L'environnement RStudio

1.1 Données du TP

Récupérer les fichiers pour le TP en suivant les indications suivantes :

- Créer un nouveau projet
 - Choisir version de contrôle
 - Choisir Git
 - écrire l'adresse suivante : <https://github.com/eHirchaud/formationR>
- Un dossier nommé formationR sera créé dans votre dossier personnel

1.2 Éléments de RStudio

Présentation des différentes fenêtres de RStudio et personnalisation.
Par défaut La fenêtre de RStudio est composée de 4 panneaux principaux
De gauche à droite puis de haut en bas

Le panneau à droite ... (images)

2 Premiers Pas

2.1 R est une calculatrice

```
1 ###Calcul###
2
3 # Une opération simple
4 10 + 3
5 # Une Opération plus complexe
6 10 / (3+8) * 78
7 # les différents opérateurs sont :
8 # multiplication *
9 # addition +
10 # division /
11 # soustraction -
12
13
14 ###Stockage###
15
16 #stockage d'une valeur dans une variable
17 nombreX <- 50.8
18 #Accès à la valeur stockée dans la variable nombreX
19 #Utiliser l'auto-complétion. Commencer par écrire nomb puis appuyer sur la
    touche de tabulation.
20 nombreX
21
22
23
24 # Utilisation de plusieurs variables
25 # stockage de la variable nombreY (le symbole <- est identique à =)
26 nombreY <- 7.4
27 # Utiliser l'auto-complétion, si plusieurs solutions existent vous pouvez
28 # continuer de taper le nom de la variable pour que le choix soit plus
29 # restreint puis utiliser les flèches du clavier pour choisir la bonne
30 # variable. Une fois sélectionnée, appuyer soit sur entrée.
31 nombreX + nombreY
32 # Stockage du résultat de l'opération dans la variable sommeXY
33 # Conseil : utiliser l'historique en appuyant sur la 'flèche haut'
34 # on peut retrouver des commandes déjà écrites.
35 sommeXY <- nombreX + nombreY
36 # Opération avec variable et constante déjà existante.
```

2.2 Créer des objets et s'informer

2.2.1 Vecteurs

```

1 #####Création###
2
3 #création d'un vecteur numérique
4 monVecteur1 <- c(20,45,78,12)
5
6 # Une suite de nombre de 1 à 30
7 maSuite <- seq(from = 1, to = 30)
8
9 #Une suite de nombre paire
10 maSuitePaire <- seq(from = 2, to = 20, by = 2)
11
12
13 # Nous avons utiliser la fonction seq()
14 # Pour trouver à quoi serve les arguments taper la commande
15 ?seq
16
17 # La documentation de cette fonction s'affiche sur la panneau en bas à
   droite de RStudio
18
19
20 #Création d'un vecteur avec chaine de caractère.
21 monVecteurA <- c("Mut_1", "Mut_2", "Mut_3")
22
23
24 ###S'informer sur les objets####
25
26 ##Taille de l'objet##
27
28 #Connaitre la longueur d'un vecteur au moyen de la fonction length
29 # Indiquer la longueur de tous les vecteur créer
30 # Pour connaitre toutes les variables créer il faut regarder l'onglet
   Workspace
31 # dans le panneau en haut à droite de R studio.
32 # Il est également possible d'utiliser la fonction ls() dans la console R.
33
34 #Longueur des vecteurs :
35 #
36 #
37 #
38 #
39
40
41 ##Accéder au élément##
42
43 #Accès à l'élément d'indice 1
44 monVecteur1[1]
45 #Accéder aux élément d'indice 3 , 4 puis 5.
46 # Remarques.
47
48
49 #Accès aux éléments d'indice de 1 à 3
50 monVecteur1[1:3]
51
52 ##Calcul##

```

```

53
54 #opération sur le vecteur
55 monVecteur1 + 5.5
56
57 #addition de deux vecteurs
58 monVecteur2 <- c(10, 100, 5, 2)
59 monVecteur1 + monVecteur2

```

2.2.2 Matrices et data frame

```

1 ###Creation###
2
3 ##Matrice avec des nombres aléatoire##
4 maMatrice <- matrix(rnorm(100), ncol = 10)
5
6 ##Expliquer cette ligne de commande##
7 #A quoi corresponde les différents mots composant cette ligne de commande.
8
9 ##S'informer sur la matrice
10 #dimension de la matrice
11 dim(maMatrice)
12 #que renvoie la fonction dim()
13
14 #La fonction ncol() permet de nous renseigner sur le nombre de colone
15 #Trouver une fonction similaire pour trouver le nombre de ligne
16
17 #Avec la fonction dim() afficher uniquement le nombre de lignes

```

3 Notions pratiques

4 Import Export

```

1 # Importation de données
2 mesData <- read.delim("Rmatrice.txt")
3
4 # Le troisième argument 'median' signifie que l'on applique (apply) le
   calcul de la médiane sur toutes colonnes.
5 geneMedian
6 length(geneMedian)
7
8 #Question : De la même façon créer un profil médian (échantillon médian).
9 # A compléter
10 profilMedian <- apply(
11
12 #Sauvegarde dans un fichier
13 write.table(x = maMatriceLog, file = "logmat.txt", col.names = NA , row.
   names = TRUE, quote = FALSE, sep = "\t")

```

5 Graphiques

```

1 # Utilisation basique
2 boxplot(maMatriceLog)
3 # Avec quelques paramètres
4 boxplot(maMatriceLog, col = "blue" , las = "2")
5
6 # Sauvegarde dans un fichier image

```

```

7 # Dans l'onglet Plots : Export-'Save Plot As Image'
8 # File name : boxPlot
9 # La même chose avec la commande :
10 dev.print(jpeg, file = "boxPlot.jpeg")
11
12 # La fonction plot permet de tracer une série de données contre une autre (x
    ,y).
13 # Par exemple on sélectionne la première colonne.
14 ech1 = maMatrice[, 1]
15
16 # En supposant que le profil median calculé précédemment se nomme
    profilMedian :
17 # Graphique simple (l'argument log permet d'obtenir une échelle en log pour
    les axes mentionnés)
18 plot(x = profilMedian, y = ech1, log = "xy")
19
20 # Le même graphique avec des paramètres supplémentaires
21 plot(x = profilMedian, y = ech1, log = "xy" , pch = 19, col = "darkgreen",
    xlab = "Profil Median", ylab = "Echantillon 1", main = "Profil median VS
    Echantillon 1")

```

6 Script

Differentes methodes pour executer un script Ouvrir le fichier 'scriptTuto.R'.

Pas à pas : Dans la fenêtre source, placer le curseur sur la première ligne : Cliquer sur l'icone « Run » ou le raccourci clavier Ctrl+Entrée

Par bloc :

- Sélectionner les lignes à exécuter avec la souris ou le clavier Maj+flèches.
- Puis de la même façon que pour le pas à pas cliquer sur « Run » ou Ctrl+Entrée.

Tout le fichier : Cliquer sur Source ou le raccourcis clavier Maj+Ctrl+S

7 Écrire et utiliser une fonction

Objectif : connaître la syntaxe de la création d'une fonction. Savoir mettre en mémoire une fonction pour l'utiliser ('sourcer').

Fonction simple Ouvrir le fichier fonctionTuto.R et le fichier script2Tuto.R : Répéter l'exercice précédent avec le fichier script2Tuto.R. On obtient un message d'erreur : la fonction addition n'existe pas pour R. Pour lui faire connaître il faut 'sourcer' le fichier qui contient la fonction.

Écrire une fonction similaire à addition du style : soustraction, multiplication... Utiliser la fonction écrite dans le fichier script2Tuto.R . Exécuter tout le script.

Pour éviter d'oublier de 'sourcer' le fichier, la méthode standard est d'écrire au début du fichier la commande :

```
> source("fonctionTuto.R")
```

Une autre option propre à RStudio est de cocher la case 'Source on Save' pour le fichier fonctionTuto.R. Cela a pour effet de 'sourcer' automatiquement le fichier fonctionTuto.R à chaque sauvegarde de ce fichier.

8 Package