

# R initiation

*Datastorm - B. Thieurmél*

## 1 Trouver de l'aide avec R

Pour avoir de l'aide sur la fonction `mean`, il suffit de taper :

```
?mean  
help(mean)
```

On peut également utiliser l'onglet **help** de **RStudio**.

## 2 Premiers calculs

### 2.1 Expressions simples

Exécutez et commentez :

```
2 + 2  
# blabla  
2 + 2 # ceci est une addition  
pi  
exp(2)  
log(10)  
sin(5*pi)  
(1+3/5)*5
```

### 2.2 Calculs sur plusieurs valeurs

Si nous voulons faire une moyenne de notes, il faut pouvoir manipuler plusieurs valeurs ensemble (et donc un vecteur).

Effectuons la moyenne de 4, 10 et 16 :

```
mean(c(4, 10, 16))
```

*Exercices d'application :*

- Calculer la moyenne de 1, 3, 5, 4, et 8.
- Calculer la somme (`sum`) de 4, 10 et 16
- Calculer la médiane (`median`) de 4, 10 et 16

### 2.3 Mise en mémoire

Nous souhaitons stocker un vecteur pour le réutiliser. Nous devons donc affecter des valeurs à une variable.

Exécutez et commentez :

```
x <- pi  
print(x)  
x
```

```
objects()
y = pi
objects()
y
x <- c(4, 10, 16)
print(x)
x
```

**Conclusion :** l'affectation crée l'objet ou écrase l'objet si il existe. L'affectation est réalisée par `<-` ou `=`.

*Exercices d'application :*

- Calculer le max (`max`) de x.
- Calculer le min (`min`) de x.
- Calculer la moyenne (`mean`) de x.
- Calculer la longueur (`length`) de x.
- Calculer le résumé numérique (`summary`) de x.

## 3 Manipulation des vecteurs

### 3.1 Calcul vectoriel

Additionnons 2 vecteurs :

```
y <- c(-1, 5, 0)
x
y
x + y
-y
```

Commentez les deux derniers ordres ci-dessus et ceux ci-dessous

```
x + 2
abs(y)
```

Intéressons nous à la multiplication. Exécutez et commentez :

```
x * y
x / y
x^2
```

Une nouvelle opération :

```
1:3
1:10
-1:5
-(1:5)
```

### 3.2 Vecteur de logiques

Les logiques sont soit `TRUE` soit `FALSE` (que l'on peut abrévier par T et F)

```
w <- c(TRUE, FALSE, FALSE)
sum(w)
any(w) # au-moins une valeur TRUE
all(w) # toutes les valeurs TRUE
```

```
!w # négation

# opérateurs logiques
(TRUE) & (FALSE)
(TRUE) | (FALSE)
(TRUE) | (TRUE)
```

### 3.3 Valeurs spéciales et calculs

La valeur NA est la valeur manquante. La valeur NaN est la valeur *Not a Number* (forme indéterminée). Inf est l'infini, et NULL pour la valeur nulle.

```
log(0)
log(Inf)
1/0
0/0
max(c( 0/0, 1, 10))
max(c(NA, 1, 10))
max(c(NA, 1, 10), na.rm = T)
max(c(-Inf, 1, 10))
is.finite(c(-Inf, 1, 10))
is.na(c(NA, 1, 10))
which(is.na(c(NA, 1, 10)))
is.nan(c(NaN, 1, 10))
```

### 3.4 Créer des vecteurs

1. Créer le vecteur d'entier de 5 à 23.
2. Créer le vecteur de 6 à 24 allant de 2 en 2.
3. Créer le vecteur de 100 valeurs régulièrement espacées entre 0 et 1.
4. Créer les vecteurs suivants :

```
[1] 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
```

```
[1] 1 1 1 2 2 2 3 3 3 4 4 4 5 5 5
```

```
[1] 1 1 2 2 2 3 3 3 3
```

### 3.5 Sélection dans un vecteur

Sélectionnons les coordonnées dans un vecteur par leur numéro :

```
x[1]
x[2]
x[c(1, 2, 3)]
x[1:3]
x[c(2, 2, 1, 3)]
x[c(1:3, 2, 1)]
x[-1]
x[-c(1, 2)]
x[-(1:2)]
```

Sélectionnons les coordonnées dans un vecteur par des logiques. Pour cela, commenter les ordres suivants :

```
objects()
vec1 <- c(3, NA, 4)
objects()
vec2 <- c(FALSE, TRUE, FALSE)
objects(pattern = "vec*")
vec2
vec1
vec1[vec2]
is.na(vec1)
```

Tapez l'ordre suivant sans essayer de l'interpréter (dans un premier temps).

```
vec1 <- runif(20) ; vec1[vec1 > 0.5] <- NA
```

En utilisant le groupe d'ordre précédent, remplacez les valeurs manquantes de `vec1` par 0 ; retournez sur la ligne précédente et l'interpréter.

## 4 Importation et manipulation de fichiers

Importer les fichiers `tab1.csv`, `tab2.csv`, `tab3.csv` dans des variables **R** `don1`, `don2` et `don3`. Le resultat de l'affichage à l'écran doit être :

```
don1
```

```
##   V1 V2
## 1  1  2
## 2  0  2
## 3  3  1
```

```
don2
```

```
##   variable1 variable2
## 1      -1.0         0
## 2       2.0        -2
## 3       3.1         4
```

```
don3
```

```
##   nomligne   sexe taille
## 1     gege masculin 180.6
## 2    simone  feminin 175.2
## 3   albert masculin 172.9
```

```
don3
```

```
##   nomligne   sexe taille
## 1     gege masculin 180.6
## 2    simone  feminin 175.2
## 3   albert masculin 172.9
```

Quels sont les types de chaque variable (quantitatif ou qualitatif) ?

```
summary(don3)
```

### 4.1 Les noms des variables (colonnes) et des individus (lignes)

Exécuter et commenter :

```
rownames(don3) ## c'est un vecteur
names(don3) ## c'est un vecteur
colnames(don3) ## c'est un vecteur
colnames(don1)[2]
colnames(don1)[2] <- "var2"
colnames(don1)
colnames(don1) <- colnames(don2)
colnames(don1) <- c("variable1","variable2")
```

## 4.2 Sélection dans des tableaux

Exécuter et commenter :

```
don1[1,]
don3[, "sexe"]
don3$sexe
don3[,2]
don3[,c(FALSE, TRUE)]
don3[,c("taille", "sexe")]
don1[1,2]
don1[,1:2]
don1[-1,]
don1[c(2, 3),c(2, 1)]
don1[,c(TRUE, FALSE, TRUE), ]
don1[don1[,1]>0, ]
don1[-(1), ]
don1[,c(2, 1)]
```

## 4.3 Opération sur les colonnes

Exécuter et commenter :

```
don1[,1] + don1[,2]
exp(don1[, 1])
don1[1,] + don1[2,]
```

## 4.4 Fusionner des tableaux

Importer les tableaux **test1.csv**, **test2.csv** et **test3.csv** dans les variables **don1**, **don2** et **don3**.

Exécutez et commentez :

```
don4 <- cbind(don2, don3)
don4
don5 = rbind(don1, don2)
don5
rbind(don1, don3)
objects()
rm(don4, don5)
objects()
```

Utilisation de la fonction merge

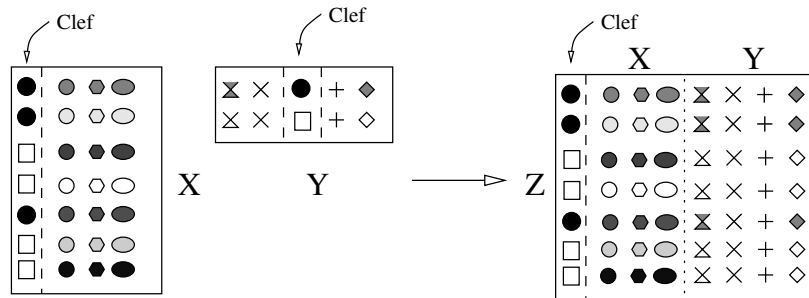


Figure 1: Fusion par clef : `merge(X,Y,by="clef")` où `clef` est le nom d'une variable commune à X et Y

Il est possible de fusionner deux tableaux selon une clef grâce à l'ordre classique `merge` :

Importer les données du fichier `tournesol_propre.csv` (dans que l'on nommera `tpropre`) ainsi que celle du fichier `meteo_tournesol.csv` (dans un tableau nommé `meteo`). Fusionner les tableaux `tpropre` et `meteo` avec la clef `ecotype`.

Sauvegarder le résultat dans un fichier `.RDS` (`saveRDS`) et réimporter le dans **R** (`readRDS`).

## 5 Traitement des facteurs

1. Importer l'objet `mtcars` (`data(mtcars)`).
2. Donner un résumé numérique de chaque variable de la matrice `mtcars` (voir `summary`).
3. Créer un facteur (de nom `conso`) en découpant en classe la variable `mpg`. Le découpage sera fait de 5 en 5 en partant de 10 (voir `cut` et `seq`).
4. Afficher les niveaux (ou modalités) de `conso` (`levels`).
5. Créer une nouvelle colonne `conso2` égale à `conso`.
6. Fusionner la première et la seconde modalité de `conso2` en une seule de nom "fuse" (`levels`).
7. Transformer `conso` en facteur ordonné (`ordered`).
8. Voir les effectif de chaque modalité (`table`)
9. Voir les pourcentages de chaque modalité (`table, sum, length`)
10. Refaire un résumé numérique
11. Faire la moyenne de chaque variable de la matrice `mtcars` (`colMeans`).
12. Faire la somme, variable par variable, des 4 premières colonnes `mtcars` (`colSums`).

## 6 Chaînes de caractères (pour aller plus loin)

Exécutez et commentez :

```
str <- c("Logiciel", "R")
str[1]
nchar(str)
tolower(str)
toupper(str)
paste(str, collapse = "-")
paste(str[1], str[2], sep = "-")
paste("Jour", 1:3)
paste("Jour", 1:3, sep = "")
c(paste0("Jour", 1:3), paste("Heure", 1:4, sep = "."), str)
```

```
gsub("Logiciel", "Software", str)
grep("Logiciel", str)
grep("Logiciel", str, value = TRUE)
grepl("Logiciel", str)
```

Essayez d'extraire les adresses mail valides du vecteur ci-dessous

```
v_mail <- c("robert.dupont@gmail.com", "12345@.fr", "contact@hotmail.fr",
            "blabla", "12345@", "rdupont@gmail.com")
```