

# Introduction à R

## 1 Entrer, sortir et trouver de l'aide avec R

R ouvre une session à chaque entrée. Par défaut (sous windows) la session est toujours dans le même répertoire. Pour changer de répertoire utiliser le menu.

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

```
> help(mean)
```

Une aide html est disponible avec

```
> help.start()
```

A la sortie, la session peut être sauvegardée, ce qui en général n'est pas très utile.

## 2 R est une (super) machine à calculer

### 2.1 Calculs simples

Exécutez et commentez :

```
> 2+2
> # blabla
> 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 (donc un vecteur). Effectuons la moyenne (`mean`) de 4, 10 et 16 :

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

En utilisant la ligne suivante

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

trouvez à quoi sert la fonction `c()`.

#### Exercices d'application :

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

## 2.3 Mettre en mémoire plusieurs valeurs

Nous souhaitons stocker un vecteur pour le réutiliser. Nous devons donc *affecter* des valeurs à un nom. Exécuter et commenter ces ordres :

```
> 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. L'affectation est réalisée par `<-` ou `=`.

### Exercices d'application :

1. Calculer le max (`max`) de `x`.
2. Calculer le min (`min`) de `x`.
3. Calculer la moyenne (`mean`) de `x`.
4. Calculer la longueur (`length`) de `x`.
5. Calculer le résumé numérique (`summary`) de `x`.

## 3 R manipule des vecteurs

### 3.1 Calcul vectoriel

Additionnons 2 vecteurs :

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

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

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

Intéressons nous à la multiplication. Exécuter et commenter :

```
> 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`<sup>1</sup>)

```
> w <- c(TRUE,FALSE,FALSE)
> sum(w)
> any(w)
> all(w)
> !w
> (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). Enfin `Inf` est l'infini. Ces valeurs

```
> log(0)
> log(Inf)
> 1/0
> 0/0
> max(c( 0/0,1,10))
> max(c(NA,1,10))
> max(c(-Inf,1,10))
> is.finite(c(-Inf,1,10))
> 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 le vecteur suivant

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

5. Créer le vecteur suivant

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

6. Créer le vecteur suivant

```
[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 :

- 
1. attention aux majuscules

```

> 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.

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

Exécutez et commenter

```

> z=c("aze","fds")
> z[1]
> paste("m",1:3)
> paste("m",1:3,sep="")
> c(paste("m",1:3,sep=""),paste("p",1:4,sep="."),z)

```

## 4 R est un (super) tableur

1. Importer les données du tableau contenu dans le fichier `tab1.csv` dans l'objet R que vous appellerez `don1`. Le resultat de l'affichage à l'écran doit être

```

> don1
  V1 V2
1  1  2
2  0  2
3  3  1

```

2. Importer les données du tableau contenu dans le fichier `tab2.csv` dans l'objet R que vous appellerez `don2`. Le resultat de l'affichage à l'écran doit être

```
> don2
  variable1 variable2
1      -1.0         0
2       2.0        -2
3       3.1         4
```

3. Importer les données du tableau contenu dans le fichier `tab3.csv` dans l'objet R que vous appellerez `don3`. Le résultat de l'affichage à l'écran doit être

```
> don3
      sexe  taille
gege  masculin  180.6
simone féminin  175.2
albert masculin  172.9
```

4. Importer les données du tableau contenu dans le fichier `tournesols.csv` dans l'objet R que vous appellerez `tournesols`. Ce tableau contient les mesures, sur différents individus statistiques (plantes de tournesols sauvages), des variables décrites dans le tableau 1.

Code variable	Descriptif variable
ecotype	code plante
plt	numéro du plant d'un écotype donné
etat	état d'origine de la plante (aux USA)
longitude	longitude du lieu de collecte (aux USA)
latitude	latitude du lieu de collecte (aux USA)
haut	hauteur des plants
semflo	jour de floraison (écart en jour par rapport au premier mai)
rambas	note de ramification basale (entre 0 aucune et 4 maximum)
longfeu	longueur du cumulée du limbe et du pétiole (cm ?)
grlon	longueur maxi de la graine (mm, moyenne sur 15 graines minimum)
huile	pourcentage d'huile

TABLE 1 – Variables mesurées sur les tournesols (dans la station d'essai aux environs de Montpellier).

Ce tableau sera affecté (après importation dans R) dans l'objet appelé `tournesols` et l'affichage du résumé de celui-ci à l'écran doit être :

```
> summary(tournesols)
      ecotype      plt      etat      longitude
Min.   : 209.0   Min.   : 1.000   TEX     : 40   Min.     :-121.77
1st Qu.: 511.0   1st Qu.: 2.000   WYO     : 38   1st Qu.  :-108.54
Median : 775.0   Median : 4.000   COL     : 35   Median   :-104.08
Mean    : 759.1   Mean    : 3.991   CLF     : 34   Mean     :-104.61
3rd Qu.: 975.0   3rd Qu.: 5.000   SDK     : 34   3rd Qu.  :-99.85
Max.     :1150.0   Max.     :10.000   UTA     : 25   Max.     : -88.07
      (Other):121
      latitude      semflo      haut      rambas
Min.     :26.19   Min.     : 18.00   Min.     : 50.0   Min.     :0.000
1st Qu.  :35.00   1st Qu.  : 30.00   1st Qu.  :145.0   1st Qu.  :2.000
Median   :39.17   Median   : 40.00   Median   :190.0   Median   :3.000
Mean     :38.96   Mean     : 40.43   Mean     :186.1   Mean     :2.633
3rd Qu.  :43.77   3rd Qu.  : 47.00   3rd Qu.  :227.5   3rd Qu.  :3.000
Max.     :48.78   Max.     :113.00   Max.     :330.0   Max.     :4.000
```

longfeu	grlon	huile
Min. :10.00	Min. :3.680	Min. :15.24
1st Qu.:27.00	1st Qu.:4.480	1st Qu.:24.70
Median :42.00	Median :4.840	Median :26.69
Mean :39.47	Mean :5.004	Mean :26.75
3rd Qu.:51.00	3rd Qu.:5.370	3rd Qu.:28.86
Max. :70.00	Max. :8.160	Max. :35.50

## 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])
> don3[1,]+don3[2,]
```

## 4.4 R et résumé de tableaux

Exécuter :

```
> summary(tournesols) ## tableau issu de l'importation de tournesol.csv
```

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

## 5 Elimination des valeurs manquantes

Importez le tableau `tournesols_brut.csv`.

Lorsque l'on souhaite connaître les valeurs manquantes (NA) d'un vecteur :

```
> which(is.na(tbrut[, "huile"]))
```

Lorsque l'on souhaite connaître les valeurs manquantes (NA) de tout un tableau :

```
> which(is.na(tbrut), arr.ind=TRUE)
```

1. Affichez (grâce à une fonction) les noms de la première colonne et de la seconde colonne du tableau résultant de l'ordre précédent (NA dans tout un tableau).
2. Affichez la première ligne. Que représente t-elle ?
3. En utilisant la réponse à la question précédente, enlever les lignes qui comportent des valeurs manquantes. Affecter le résultat dans le tableau `tbrut2`.