

# TP1: Introduction à R

*Sonia Tiao*

*15/10/2020*

## 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()
```

## 2. R comme 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:

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

En utilisant la ligne suivante:

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

Trouvez à quoi sert la fonction c().

### Exercice d'application:

1. Calculer la moyenne de 2,3,9,5,4
2. Calculer la somme de 4,6,20
3. Calculer la médiane de 4,6,20

## 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 dans cet ordre et commenter ces ordres :

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

Que remarquez-vous ?

### Exercices d'application :

```
x = c(32,48,10,2)
```

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, commentez:

```
y=c(-1,5,0,2)
x
y
x+y
-y
```

Commenter

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

Commenter

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

### 3-2.Vecteur de logiques

Les logiques sont soit TRUE soit FALSE.

```
w <- c(TRUE,FALSE,FALSE)
sum(w)
all(w)
!w
(TRUE)&(FALSE)
(TRUE)|(FALSE)
(TRUE)|(TRUE)
```

### 3-3. Valeurs spéciales et calculs

- NA est la valeur manquante
- NaN est la valeur « Not a Number »
- Inf est l'infini

```
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'entiers 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 2 2 2 3 3 3 3

### 3-5. Sélectionner des vecteurs

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)]
```

Par des logiques:

```
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'ordres précédents, 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 commentez

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

|   | V1 | V2 |
|---|----|----|
| 1 | 1  | 2  |
| 2 | 0  | 2  |
| 3 | 3  | 1  |

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

|   | 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.xls dans l'objet R que vous appellerez don3.  
Le resultat de l'affichage à l'écran doit être :

|        | sexe     | taille |
|--------|----------|--------|
| aang   | masculin | 172,9  |
| katara | feminin  | 175,2  |
| sokka  | masculin | 180,5  |

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

Exécuter et commenter :

```
rownames(don3)
names(don3)
colnames(don3)
colnames(don1)[2]
colnames(don1)[2] <- "var2"
colnames(don1)
colnames(don1) <- colnames(don2)
colnames(don1) <- c("VAR1" , "VAR2")
```

#### 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 Sélection dans des tableaux

Exécuter et commenter :

```
don1[,1]+don1[,2]
exp(don1[,1])
don3[1,]+don3[2,]
```

#### 4.4 Fusion des tableaux

1. Importez les données du tableau contenu dans le fichier tab3.xls dans l'objet R que vous appellerez avatar1. Le resultat de l'affichage à l'écran doit être:

| nomligne | sexe     | taille |
|----------|----------|--------|
| aang     | masculin | 172,9  |
| katarA   | feminin  | 175,2  |
| sokka    | masculin | 180,5  |

2. Importez les données du tableau contenu dans le fichier tab4.xls dans l'objet R que vous appellerez avatar2. Le resultat de l'affichage à l'écran doit être:

| nomligne | sexe     | taille |
|----------|----------|--------|
| zuko     | masculin | 172    |
| toph     | feminin  | 150,2  |
| iroh     | masculin | 160,5  |
| azula    | feminin  | 176    |

3. Importez les données du tableau contenu dans le fichier tab4.xls dans l'objet R que vous appellerez avatar2. Le resultat de l'affichage à l'écran doit être:

| nomlign | element |
|---------|---------|
| aang    | air     |
| katara  | eau     |
| sokka   | NA      |
| zuko    | feu     |
| toph    | terre   |
| iroh    | feu     |
| azula   | feu     |

4. Fusionner les données avatar1 et avatar2 par lignes, appeler ce nouveau tableau all.avatar
5. Que se passe t'il dans les cas suivant ? Commentez

```
cbind(avatar1,avatar2)

colnames(avatar1)[1] <- 'prenom'
rbind(avatar1,avatar2)
colnames(avatar1)[1] <- 'nomligne'
```

6. Fusionner le tableau all.avatar avec avatar3, et nommer ce nouveau tableau allfusion.avatar:

#### 4.5 Elimination des valeurs manquantes

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

```
which(is.na( allfusion.avatar[, "element"] ))
```

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

```
which(is.na(allfusion.avatar),arr.ind=TRUE)
```

Enlevez du tableau allfusion.avatar la ligne avec cette valeur manquante:

#### 4.6 Créer votre propre dataframe et importer le:

##### Exercice d'application:

Sur R:

1. Créer le vecteur nom contenant les noms de 10 personnes.
2. Créer le vecteur age avec l'âge des 10 personnes (entre 1 et 100 ans). Utiliser le noms des personnes pour nommer les elements du vecteur age.
3. Créer le vecteur cheveux avec la couleur des cheveux de ces 10 personnes. Ce vecteur doit etre un vecteur composés de valeurs type factor.
4. Créer le vecteur taille avec la taille de ces 10 personnes. (en cm)
5. Créer un tableau avec toutes les informations.
6. A partir de ce taleau: créer un sous-tableau contenant: les individus de plus de 18 ans, et mesurent entre 150cm et 180cm. Eliminez les individus qui ont les cheveux "INSEREZ UNE COULEUR ;)".
7. Exportez ce tableau crée dans le dossier TP1.