

# Module 3 - Bases de R - Séance 1

DUBii 2019

Hugo Varet, Frédéric Guyon, Olivier Kirsh et Jacques van Helden

2019-01-29

## Les types de données élémentaires

- ▶ les nombres (réels par défaut): `numeric`, `double`

```
x=3.5  
class(x)  
is.double(x)  
is.numeric(x)
```

- ▶ les nombres entiers: `integer`

```
x=3.5  
x=as.integer(x)  
is.integer(x)
```

- ▶ les caractères: `character`

```
x="a"  
x="toto"
```

# Les types de données

- ▶ les booléens: `boolean`

```
x=FALSE  
x=TRUE  
as.integer(x)
```

- ▶ les facteurs: `factor`

très fréquent en R: on verra plus tard

## Les types de données: les vecteurs

- ▶ tout est vecteur
- ▶ vecteur: ensemble de valeurs du même type et indexées

```
x=c(1,3,4,7)
x[1]
x[4]
length(x)
```

- ▶ listes: ensemble de valeurs de types différents et indexées

```
x=list(names=c("toto", "titi", "tutu"), values=c(1,2,4))
print(x)
x[[2]]
x$values
```

# Les types de données: les tableaux de données

- ▶ type de données typique de l'analyse de données
- ▶ data.frame: tableaux dont les colonnes sont de même types

```
A=data.frame(names=c("toto","titi","tutu"),values=c(1,2,4))  
print(A)
```

# Manipulation des vecteurs

```
x=rnorm(26)
```

- ▶ indigage des vecteurs: []

```
x[3]
```

- ▶ vecteur d'indices

```
x[c(3,5)]
```

```
x[1:6]
```

```
x[seq(1,100,by=2)]
```

- ▶ vecteur de booléens=on prend ou pas sous condition

```
x[x>=0]
```

# Manipulation des vecteurs

- ▶ accès aux éléments de vecteur par le nom

```
names(x)=letters  
x[c("a","t","z")]
```

- ▶ accès aux éléments éléments de liste par le nom

```
x=list(names=letters,values=1:26)  
x$names  
x$letters
```

- ▶ data.frame = tableau à deux dimensions

```
A=data.frame(names=c("toto","titi","tutu"),values=c(1,2,4))  
A[1,2]  
A[2,1]  
A[1:2,]  
A[A[,2]>=3,]
```

# Les fonctions de base

- ▶ +,-,\*,\,\*\*
- ▶ cos,sin,log,log10,exp
- ▶ les fonctions sont toutes vectorielles

```
x=runif(10)
x=x+1
print(x)
cos(x)
y=seq(0, pi, len=10)
x+y
x*y
```



# Les fonctions en R

```
ma_fonction=function(nom_arg1,nom_arg2) {  
  # des calculs...  
  return(valeur_retour)  
}
```

- ▶ `ma_fonction`: nom de la fonction (variable comme les autres)
- ▶ `valeur1`, `valeur2`: arguments de la fonction
- ▶ `valeur_retour`: la dernière ligne donne la valeur retournée par la fonction
- ▶ exemple simple:

```
add1=function(x) {  
  return(x+1)  
}
```

```
add1(2)
```

# Les fonctions en R

- ▶ les arguments ont un nom et peuvent avoir une valeur par défaut
- ▶ à l'appel de la fonction:
  - ▶ les arguments sont dans l'ordre
  - ▶ dans le désordre si ils sont nommés
  - ▶ peuvent être absents, si valeur par défaut
- ▶ exemple simple:

```
add2=function(x,y,z=0) {  
  x+2*y+3*z  
}  
add2(1,2)  
add2(y=1,x=2)  
add2(1,2,3)  
add2(y=1,x=2,3)
```