









TP 1 ☐ Présentation générale du logiciel , l'aide en ligne et les bases du langage

 est distribué gratuitement à partir du site du CRAN (Comprehensive R Archive Network) ☐ <http://www.r-project.org/>. Je vous suggère d'aller visiter le site officiel de .

 est un système d'analyse statistique créé par Ross Ihaka et Robert Gentleman distribué librement sous les termes de la GNU General Public Licence ☐ son développement et sa distribution sont assurés par plusieurs statisticiens rassemblés dans le R Development Core Team. Pour avoir une liste des chercheurs qui ont participé au développement de 

Tapez sur la fenêtre de commande `contributors()`

Toutes les fonctions de  sont stockées dans une grande bibliothèque, Cette bibliothèque contient des packages de fonctions. Allons donc voir les packages disponibles à l'installation du logiciel .

Tapez la commande `installed.packages()`

Notons les différences de «`priority`» entre les différents packages

Quel impact à la `priority` sur l'utilisation du package ☐


A quoi sert la fonction `library()` ☐ La fonction `library()` sert-elle à tous les packages ☐


Correction ☐:

Les fonctions associées à un package de `priority = «base»` sont directement utilisables. A l'inverse, les fonctions associées à un package de `priority = «recommended»` ou «`NA`» ne sont pas directement utilisables. Le package doit être préalablement chargé en mémoire par l'intermédiaire de la fonction `library()`.

Exemple ☐:

On souhaite faire une analyse discriminante. La fonction qui le permet est la fonction `lda()` du package `MASS`. Or le package `MASS` a une `priority = «recommended»`. Pour utiliser la fonction `lda()` il suffit de charger le package `MASS` en mémoire via la fonction `library`
`library(MASS)`

Le package nommé «`base`» est le cœur de  et contient les fonctions de base du langage pour la lecture et la manipulation des données, la création de certains types de graphiques et certaines analyses statistiques.

Par ailleurs, de nombreux packages, développés par les utilisateurs de , sont disponibles sur le site internet du CRAN et téléchargeable gratuitement à l'adresse suivante ☐ <http://cran.us.r-project.org/src/contrib/PACKAGES.html>. Sur cette page est répertoriée la liste des packages disponibles. Allez donc la visiter....



Création d'objets basiques

Affecter la valeur 27 à l'objet nommé x .
Affecter la valeur 9 à l'objet nommé X .
Visualiser les valeurs de x et X
Que constate t'on ?

différencie les majuscules des minuscules

Remarque Pour affecter la valeur 27 à l'objet nommé x , on peut procéder de deux manières

```
x = 27  
x <- 27
```

Affecter la valeur 5 à l'objet nommé x . Visualiser la valeur de x
Affecter la valeur 9 à l'objet nommé x . Visualiser la valeur de x
Que constate t'on ?

Lorsqu'on affecte une valeur à une variable existante, écrase les valeurs de la variable.

On peut affecter, dans une même commande, des valeurs à plusieurs objets. Il suffit de séparer les affectations par des `,`.

Créer les variables $w = 8$, $name = "Arthur"$ et $dicton = "Vive le logiciel R"$ sur une ligne de commande.

```
w = 8; name = "Arthur"; dicton = "Vive le logiciel R";  
w ; name ; dicton  
[1] 8  
[1] "Arthur"  
[1] "Vive le logiciel R"
```

Gestion de la mémoire les fonctions `ls()` et `rm()`

Consulter les objets en mémoire



Afin de connaître les objets chargés en mémoire on utilise la commande ls().

Stockez les 5 valeurs "aujourd'hui, nous sommes 22 étudiants, bienvenue!!!" "À vos claviers", "la séance risque d'être rude", "je suis là pour vous aider" et 22 dans, respectivement, les objets bienvenue, dicton, attention, aide et nombre.
Affichez les objets en mémoire.

```
bienvenue = "aujourd'hui, nous sommes 22 étudiants, bienvenue!!!"  
diction = "À vos claviers"  
attention = "la séance risque d'être rude",  
aide = "je suis là pour vous aider"  
nombre = 22
```

Affichez les objets dont le nom contient la lettre «n»

```
ls(pat = "n")
```

Affichez les objets dont le nom commence par la lettre «a»

```
ls(pat = "^a")
```

Affichez la liste détaillée des objets en mémoire

```
ls.str()
```

Effacer les objets en mémoire

Effacer tous les objets en mémoire

```
rm(list= ls())
```

Créer 12 objets un = 1, deux = 2, trois = 3, quatre = 4, cinq = "cinq", six = 6, sept = "sept", huit = 8, neuf = "neuf", dix = 10, onze = 11 et douze = 12.

Effacer les objets dont le nom contient la lettre «n»

```
rm(list = ls(pat= "n"))
```

Effacer les objets dont le nom commence par la lettre «d»

```
rm(list = ls(pat= "^d"))
```



L'aide en ligne

R fournit 2 outils d'aide en ligne : «`?<code></code></code>» et «help.search<code></code></code>». Précisons que «?<code></code></code>» recherche les fonctions uniquement dans les packages chargés en mémoire tandis que «help.search<code></code></code>» recherche les fonctions dans tous les packages installés dans l'ordinateur.`

Remarque 1 R est un logiciel anglophone si bien que la requête d'une recherche est préférable en anglais.

Remarque 2 Une fois trouvées les fonctions associées aux différentes analyses, il sera intéressant d'exécuter les exemples joints au fiche détaillée (non pour comprendre les résultats mais simplement pour se familiariser avec les affichages, les sorties, ...)

Quelle est la fonction qui permet de calculer la moyenne d'un vecteur?

`mean()`

Quelle est la fonction qui permet de calculer la variance d'un vecteur?

`var()`

Quelle est la fonction qui permet de calculer la médiane d'un vecteur?

`median()`

Quelle est la fonction qui permet de générer un vecteur aléatoire de loi de poisson?

`rpois()`

Quelle est la fonction qui permet de générer un vecteur aléatoire de loi normale?

`rnorm()`

Quelle est la fonction qui permet de calculer le déterminant d'une matrice?

`det()`

Quelle est la fonction qui permet d'effectuer une analyse de la variance?

`aov()`

Quelle est la fonction qui permet d'effectuer une analyse en composante principale?

`princomp()`

Quelle est la fonction qui permet d'effectuer une régression linéaire?

`lm()`

Quelle est la fonction qui permet d'effectuer une analyse discriminante?

`lda()`

Quelle est la fonction qui permet d'effectuer un clustering de type kmeans?

`kmeans()`

Quelle est la fonction qui permet d'effectuer une régression logistique?

`glm()`



Création d'objets Les vecteurs, les matrices, les data.frames et les listes

Les vecteurs vector()

Créer le vecteur x composé de 5000 zéros
Créer le vecteur x composé de 5000 caractères.
Créer le vecteur x suivant

```
x  
[1] 1.3 2.0 5.2 4.3 2.2
```

```
x = vector("numeric", 5000)  
x = vector("character", 5000) # avec des caracteres nuls; ou bien x = rep("a", 5000)  
x = c(1.3, 2.0, 5.2, 4.3, 2.2)
```

Créer le vecteur suivant

```
x  
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29  
[30] 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55  
[56] 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81  
[82] 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
```

Indication Il n'est pas obligatoire de rentrer les éléments du vecteur à la main

```
x = 1:100
```

Créer un vecteur à 20 éléments, nommé xnorm, composées de nombre aléatoire extrait d'une loi normale de moyenne 3 et de variance 1.

```
xnorm = rnorm(20, 3, 1) ou xnorm = rnorm(20, mean = 3, sd = 1)
```

Afficher la valeur du cinquième élément de xnorm.

```
xnorm[5]
```

Afficher la valeur des 5 derniers éléments de xnorm.

```
xnorm[16:20] ou tail(xnorm,5)
```

Afficher la valeur des éléments d'indices impairs

```
xnorm[seq(1,19, by=2)] ou xnorm[c(T,F)]
```

Afficher la valeur des éléments 1, 4, 8, 12 et 18

```
xnorm[c(1, 4, 8, 12, 18)]
```

Affecter la valeur 0 aux éléments inférieurs à 3 et la valeur 1 aux éléments supérieurs à 3

```
xnorm[xnorm < 3] = 0 ; xnorm[xnorm >= 3] = 1 ;
```



Les matrices `matrix()`

Regarder la fiche détaillée de `matrix`

```
?matrix
```

Créer une la matrice identité de dimension 5 × 5

```
diag(5)
```

Créer une matrice A , de dimension 5 × 5 dont les éléments sont des nombres aléatoires extraits d'une loi normale de moyenne 3 et de variance 10

```
A = matrix(data=rnorm(x=25, mean=3, sd=10^2), nrow=5, ncol=5)
```

Afficher la dimension de la matrice A

```
dim(A)
```

Afficher les éléments de A supérieurs à 3

```
A[A>3]
```

Afficher le nombre d'éléments de A supérieurs à 3

```
length(A[A>3])
```

Remplacer les éléments de A supérieurs à 3 par 1 et les éléments de A inférieurs à 3 par 0

```
A[A<3] = 0; A[A>3]# 1 ;
```

Créer les matrices suivantes

```
B = matrix(1:25, 5, 5) et C = matrix(1:25, 5, 5, byrow = TRUE)
```

Que constatez-vous

```
Les elements du vecteur 1:25 sont stockées dans B par colonne et par ligne dans A
```

Nommer les colonnes de B par C1, C2, C3, C4 et C5 et les lignes de B par L1, L2, L3, L4 et L5

```
colnames(B) = c("C1", "C2", "C3", "C4", "C5")  
rownames(B) = c("L1", "L2", "L3", "L4", "L5")
```

Afficher les valeurs de la troisième colonne de B par le système d'indexation

```
B[, 3]
```

Afficher les valeurs de la première et cinquième ligne de B par le système d'indexation

```
B[c(1, 5), ]
```



Afficher les valeurs de la première colonne de B par le nom. L'accès aux données par le nom fonctionne-t-il dans le cadre des matrices ?

```
B$C1 ; B[,"C1"]
```

La première solution ne fonctionne pas la deuxième marche.

Convertir la matrice B en une data.frame nommé C

```
C = as.data.frame(B)
```

Les variables qualitatives nominales `factor()`

La fonction `factor()` crée des variables qualitatives nominales.

créer la variable nominale, `tignasse`, composée des 10 valeurs suivantes
blond, chatain, brun, brun, roux, blond, brun, chatain, brun, blond

```
tignasse = factor(c(1, 2, 3, 3, 4, 1, 3, 2, 3, 1),  
                 levels = 1:4,  
                 labels = c("blond", "chatain", "brun", "roux")  
                 )
```

Les data.frame `data.frame()`

Regarder la fiche détaillée de data.frame

```
?data.frame
```

Accéder aux valeurs de la troisième colonne de la data.frame C par le nom et par l'indexation

```
C$C3  
C[,3]
```

Créer une data.frame D dont la première colonne est composée des chiffres de 1 à 6 et la deuxième colonne des 6 premières lettres de l'alphabet. Quel type d'objet est la deuxième colonne de D ? Comparer son mode avec le mode de ses niveaux (levels).

```
D = data.frame(a = 1:6, b = letters[1:6]); mode(D$b); mode(levels(D$b));  
Le mode de la deuxième colonne est numeric alors que le mode de ses niveaux est  
character.
```

Créer une data.frame D dont la première colonne est composée de 6 chiffres extraits d'une loi uniforme et la deuxième colonne des 3 premières lettres de l'alphabet.
Que constatez-vous ?

```
D = data.frame(a = runif(6, 0, 1), b = letters[1:3])
```

On constate que le deuxième élément de la data.frame (b) est recyclé afin d'avoir le même nombre de valeurs que l'élément de la data.frame le plus long (a).



Créer une data.frame D dont la première colonne est composé de 5 chiffres extraits d'une loi de poisson de moyenne 8 et la deuxième colonne des 3 premières lettres de l'alphabet. Que constatez vous?

```
D = data.frame(a = rpois(5, 8), b = letters[1:3])
```

On constate que le recyclage ne s'effectue que si l'élément de la data.frame le plus court (b) est un multiple du plus long (a)

Créer une data.frame E à 5 lignes et 3 colonnes avec la contrainte que les colonnes soit de mode numeric, character et logical. Nommer les colonnes de E .

```
E = data.frame(a = 1:5, b = letters[1:5], c = c(T, F, F, T, T))
```

Accéder à la troisième colonne de E par l'indexation et le nom

Par l'indexation: `E[, 3]`
Par le nom: `E$c`

Accéder aux valeurs de la première et cinquième ligne de E par l'indexation

```
E[c(1, 5), ]
```

Modifier la valeur du troisième élément de la deuxième colonne. Est-il possible de modifier les éléments sans tenir compte du mode?

```
E[2][3, 1] = 0
```

Il n'est pas possible de modifier les éléments sans tenir compte du mode. En effet, il faut que chacun des éléments d'une data.frame soit composé de valeurs tous de même mode.

Créer une data.frame, nommé F , composé des éléments $a = c(1, 2, 3)$ et $b = c("a", "b", "c")$. Convertir F en une matrice. Que constatez-vous?

```
E = data.frame(a = c(1, 2, 3), b = c("a", "b", "c"))
```

Convertir la data.frame E en une liste

```
E = as.list(E)
```

Les listes `list()`

Regarder la fiche détaillée de list

```
?list
```

Accéder aux valeurs du deuxième élément E par le nom et par l'indexation

Par le nom: `E$b`



```
E[[2]] #Par l'indexation
```

Accéder au mode et à la longueur des différents éléments de E .

Mode et longueur des différents éléments de E

	Mode	Longueur
Premier élément	mode(E[[1]])	length(E[[1]])
Deuxième élément	mode(E[[2]])	length(E[[2]])

Considérons l'éléments de E de mode numeric. Remplacer les valeurs qui composent l'élément de mode numeric par sa moyenne.

```
E$a[1:3] = mean(E$a)
```

Convertir la liste E en une data.frame

```
E = as.data.frame(E)
```

Conversion de mode

Convertir le vecteur nommé logique en mode numeric puis en mode caractère

```
logique = c(TRUE , F, T, FALSE, TRUE, TRUE)
```

Conversion de la variable logique en mode numérique

```
log_to_num = as.numeric(logique); num_to_char = as.character(log_to_num)
```

Convertir le vecteur nommé caractère en mode numeric

```
caractere = c("1", "2", "3", "A", "/", "T", "%", "-")
```

Que remarquez-vous que signifie le symbole NA

Conversion de la variable caractère en mode numeric

```
char_to_num = as.numeric(caractere)
```

Certaines valeurs de mode character n'ont pas leur correspondance en numeric et sont remplacées par le symbole NA (Not available). R dénote les données manquantes par le symbole NA.

Convertir le vecteur nommé numérique en mode logique

```
numerique = 0:5
```

```
Num_log = as.logical(numerique)
```

Convertir le vecteur nommé facteur en mode numeric (en conservant les valeurs de facteur).

```
facteur = factor(c(1, 1, 3, 10, 5, 7, 7, 7, 4), level = c(1, 3, 4, 5, 7, 10))
```

Pour convertir la variable facteur en mode numeric en conservant les valeurs du facteur. Il faut d'abord convertir la variable en mode character puis en mode numeric.

```
fac_to_num = as.numeric(as.character(facteur))
```