

TP 1 : Initiation à R

1 R et l'interface Rstudio

R est un logiciel libre qui peut être téléchargé sur le site du CRAN (Comprehensive R archive project, <https://cran.r-project.org/>), qui fonctionne sous Microsoft Windows, Mac OS et Linux.

Pour travailler avec R on dispose de l'interface Rstudio, qui est installée sur les ordinateurs de l'université, et dont une version *open source* est téléchargeable à l'adresse <https://www.rstudio.com/>.

Cette interface comporte :

- une console, permettant d'exécuter des commandes R (en bas à gauche),
- un éditeur, permettant d'écrire des scripts R, que l'on peut ensuite lancer dans la console (en haut à gauche),
- un espace en haut à droite permettant d'afficher au choix l'historique des commandes lancées ou la liste des variables en mémoire,
- un espace en bas à droite, permettant d'afficher en particulier une fenêtre d'aide (à utiliser sans modération), une fenêtre d'affichage de graphiques ou un navigateur de fichiers.

2 Prise en main

Ouvrir un nouveau script et l'enregistrer en utilisant un nom se terminant par `.R`. Taper une commande dans le script, par exemple `1+2`. Pour la lancer dans la console, l'association des touches `Ctrl + Entrée` permet d'exécuter la ligne sur laquelle se trouve le curseur ou la zone sélectionnée.

Remarque : si besoin plusieurs lignes peuvent être sélectionnées et lancées en même temps. Pour mettre une ligne en commentaire (cette ligne ne sera alors pas lancée dans la console, même si elle est sélectionnée), la commencer par `#`.

Pour obtenir de l'aide sur une fonction, par exemple `plot`, on peut utiliser directement le moteur de recherche situé dans la fenêtre d'aide, ou taper dans la console `?plot` ou `help(plot)`. Pour lancer une recherche plus générale, par exemple sur le mot `matrix`, taper `help.search('matrix')` dans la console. Écrire dans un script et lancer les commandes suivantes. Interpréter les résultats.

Commandes de base :

```
1+1
0.1*4
a <- exp(2)
a
b <- cos(1); b
```

Création de vecteurs :

```
v <- c(1,9,-4,0.5); v
v[3]
v[1]
1:10
seq(from=0, to=1, by=0.1)
seq(from=0, to=2, length=11)
rep(x=1, times=10)
rep(x=1:2, times=3)
rep(x=1:2, each=4)
```

Opérations sur les vecteurs :

```
A <- c(1,2,3,6); B <- c(0,-4,9,4);  
exp(A)  
A+B  
A*B  
A%*%B  
A/B  
2*A  
A+1  
A+c(1,2,3)  
(A==2)  
G <- (A>1)  
A[G]  
u <- (1:10); sum(u)  
cumsum(u)
```

Création et opération sur les matrices :

```
C <- matrix(A, ncol=2, nrow=2);  
D <- matrix(B, ncol=2, nrow=2, byrow=TRUE)  
t(C)  
C%*%D  
C[1,]; D/C[1,]
```

Valeurs et vecteurs propres. La commande `eigen(C)` permet de calculer les valeurs propres et les vecteurs propres de la matrice `C`.

Tracés de graphiques. Pour tracer le graphique d'une fonction mathématique, on utilise l'instruction `curve`.

```
curve(sin(x), 0, 2*pi)
```

La commande `plot(x=..., y=..., type='l')` permet de tracer une ligne brisée qui relie les points dont les abscisses sont stockées dans `x` et les ordonnées dans `y`. D'autres options de la commande `plot` permettent de tracer des diagrammes en bâtons ou des fonctions en escalier. Pour ajouter un élément sur un `plot`, on utilise `abline` (ajout de droites), `lines` (ajout d'une ligne brisée), `points` (ajout de points).

3 Statistiques descriptives

3.1 Import de données

Pour illustrer les outils basiques de statistiques descriptives, on va charger un jeu de données, en utilisant la commande.

```
Don<-read.delim(  
"http://math.univ-lyon1.fr/homes-www/dabrowski/enseignement/ProbaL2/Donnees.csv")
```

Les données correspondent à l'âge, au poids, à la taille, à la consommation hebdomadaire d'alcool (en nombre de verres bus), au sexe, au ronflement et au tabagisme, d'un échantillon de 100 personnes.

On récupère (par exemple) les données concernant l'âge et le ronflement, de la façon suivante :

```
age<-Don[,1];  
# ou  
Don$age  
#ou  
Don[1:100,1]
```

```
#ou
Don [ , "age" ]
ronfle<-Don [ , 6]
```

ou utiliser `attach(Don)`, pour le faire pour toutes les variables.

Commencez par identifier les variables qualitatives nominales, ordinales et quantitative discrètes et continues. Tapez les variables avec les classes `factor`, `ordered`, `integer` ou `double`.

3.2 Résumés numériques

En utilisant les fonctions `mean`, `var`, `quantile`, déterminer la moyenne empirique, la variance empirique, la variance empirique non-biaisée et l'écart-type empirique pour les variables quantitatives. Déterminer la médiane pour chaque caractéristique ordinale ou quantitative de l'échantillon.

Représenter également les données à l'aide d'un diagramme en boîte `boxplot`. Rappeler les éléments représentés dans ce diagramme.

3.3 Résumés graphiques

- Représenter les variables quantitatives discrètes et continues à l'aide respectivement d'un diagramme en bâtons et d'un histogramme (fonction `hist`). Représenter leurs fonctions de répartition empiriques. (On peut utiliser les fonctions `plot(ecdf(...))` et `hist` si on veut représenter la fonction de répartition et l'histogramme pour les variables continues.)
- Représenter les variables qualitatives par un diagramme en tuyaux d'orgue (fonction `barplot`).
- Donner les tables de contingences pour les variables sexe et tabagisme. Représenter par un nuage de point les variables poids et taille.

4 Lois usuelles discrètes

4.1 Diagrammes en bâtons

Pour tracer le diagramme en bâtons décrivant de la loi binomiale $\mathcal{B}(10, 0.25)$, on peut utiliser les commandes suivantes :

```
x <- 0:10
y <- dbinom(x, size=10, prob=0.25)
plot(x, y, type='h')
```

Tracer de même un diagramme en bâtons des lois $\mathcal{B}(10, 0.5)$, $\mathcal{B}(100, 0.25)$, $\mathcal{P}(2)$, $\mathcal{P}(10)$, $\mathcal{G}(0.75)$, $\mathcal{G}(0.25)$ et de la loi uniforme sur $\{1, 2, 3, 4, 5, 6\}$. Il peut être utile de faire la recherche `distribution` dans l'aide (faire attention à la définition de la loi géométrique $\mathcal{G}(p)$, qui est à support sur \mathbb{N}^* selon le cours).

4.2 Fonctions de répartition

Tracer la fonction de répartition de la loi $\mathcal{B}(10, 0.25)$ (on pourra utiliser la fonction `pbinom` et l'option `type='s'` de la fonction `plot`). Tracer de même la fonction de répartition des lois $\mathcal{B}(100, 0.5)$ et $\mathcal{P}(3)$. Que valent $P(X \leq 3)$ et $P(X > 30)$ si X suit la loi $\mathcal{B}(50, 0.2)$?

4.3 Lien entre loi binomiale et loi de Poisson

Rappeler le théorème de convergence de la loi binomiale vers la loi de Poisson.

Illustrer cette convergence en représentant sur un même graphique les lois $\mathcal{B}(500, 0.02)$ et $\mathcal{P}(10)$.

On pourra pour cela par exemple représenter la loi sous forme de diagrammes en bâtons, et la loi $\mathcal{P}(10)$ à l'aide de la fonction `points`.