Université Claude Bernard Lyon 1 **L2 UE Statistique pour Informatique**
Année 2018 -2019
Automne 2018

TP 1 : Introduction sur R

1.1) Installation de R

R est un logiciel de statistique distribué gratuitement par le CRAN (Compréhensive R Archive Network) à l'adresse suivante http://cran.r-project.org/.
L'installation varie selon le système d'exploitation, mais les fonctionnalités sont les mêmes. L'installation de R est facile, il suffit de suivre les instructions.

Une des interfaces que nous allons utiliser est **Rstudio**. **RStudio** permet d'utiliser R dans un environnement plus convivial. Il est open-source (c'est-à-dire gratuit) et disponible sur http://www.rstudio.com/

Rstudio est formé de trois fenêtres. La **console** est l'endroit où vous pouvez taper les commandes et voir la sortie. L'onglet **environment** (Workspace) affiche tous les objets actifs. L'onglet **Historique** affiche une liste des commandes utilisées jusqu'à présent.
L'onglet **Fichiers** (Files) affiche tous les fichiers et dossiers de votre espace de travail par défaut comme si vous étiez sur une fenêtre PC / Mac. L'onglet **Plots** affiche tous vos graphiques. L'onglet **Packages** répertorie une série de packages ou d'extensions nécessaires pour exécuter certains processus. Pour plus d'informations, consultez l'onglet **Aide** (help).

Exemple :

L'onglet **Environment** stocke tout objet, valeur, fonction et tout ce que vous créez pendant votre session R. Dans l'exemple ci-dessus, si vous cliquez sur les carrés pointillés, vous pouvez voir les données sur un écran à gauche.

L'onglet Historique enregistre toutes les commandes précédentes. Cela permet de tester et d'exécuter des processus. Ici, vous pouvez enregistrer la liste entière ou vous pouvez sélectionner les commandes que vous voulez et les envoyer à un script R pour suivre votre travail.

Avant de commencer le TP1 ouvrez un nouveau script cliquer sur Fichier→Nouveau Fichier→ Script R. Enregistrer dans un sous-répertoire personnel W : avec un nom raisonnable, se terminant par l'extension ***.R****. Pour exécuter une opération dans la console, il suffit de taper* ***Ctrl+entrée*** *avec Rstudio.*

N.B : Vous travaillez maintenant dans le Script.R et pas dans la console directement.

1.2) Les différentes aides

Pour demander de l'aide sur une fonction, par exemple la fonction mean, il suffit d'utiliser :

- help (mean)
- ?mean

1.3) Les objets R

R utilise des fonctions ou des opérateurs qui agissent sur les objets (vecteurs, matrices, etc..).
La **création** d'un objet peut se faire par affectation avec un des trois opérateurs suivants ''<-'', ''->'', ''='' en donnant un nom à cet objet :

```
> a <- 25        # crée l'objet a en lui donnant la valeur 25
> b <- a                 # b reçoit la valeur a
> b = a          # b reçoit la valeur a
> a -> b                 # b reçoit la valeur a
```

Pour connaître les objets de la session, on utilise les fonctions **objects()** ou **ls()**. Pour **supprimer** l'objet a, on tape :

```
> rm(a)
```

Et pour en supprimer plusieurs :

```
> rm (objet1,objet2,objet3)
```

1.4) Le type d'un objet

Il faut connaître les principaux modes ou types de ces objets avant d'aborder les différents objets de R :

- null :NULL
- logical (booléan): TRUE, FALSE
- numeric (nombre réel) :1, 1.1, 1^{e}-09
- complex (nombre complexe) : 1+i
- character (chaine de caractères) : 'bonjour'

Pour connaître le mode d'un objet x de R il faut exécuter la commande :

>**mode(x)**

Ou bien tester l'appartenance d'un objet x à un mode particulier avec par exemples les commandes suivantes :
is.null(), **is.logical()**, **is.numeric()**, **is.complex()**, **is.character()**.

Il est possible aussi de convertir un objet x d'un mode à un autre, voici un tableau récapitulatif :

De	A	Fonction	Conservation
Logique	Numérique	**as.numeric**	FALSE → 0 TRUE → 1
Logique	Caractère	**as.character**	FALSE→"FALSE" TRUE→"TRUE"
Caractère	Numérique	**as.numeric**	''1", ''2", ''3" →1,2,3 ''A"→ NA
Caractère	Logique	**as.logical**	''FALSE"→FALSE
Numérique	Logique	**as.logical**	0→FALSE
Numérique	Caractère	**as.character**	1,2,… → ''1", ''2", …

1.5) Vecteurs et matrices

C'est la structure de données la plus simple. **Elle représente une suite des données de même type.**
Les vecteurs numériques, peuvent être construits par différentes méthodes. Voici les principales :
-Par la fonction collecteur **c** :
> a <- **c**(les éléments du vecteur séparer par une virgule)

-Par l'operateur sequence '' **:** '' :
> 1**:**10
 [1] 1 2 3 4 5 6 7 8 9 10

-Par la fonction **seq** :
> **seq**(from=1,to=10,by=0.5)
 [1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5
[17] 9.0 9.5 10.0
> **seq**(from=1,to=10,length=6)

[1] 1.0 2.8 4.6 6.4 8.2 10.0

-Par la fonction **rep** :

```
> rep(2,6)
[1] 2 2 2 2 2 2
> rep(c(1,2),each=4)
[1] 1 1 1 1 2 2 2 2
```

Pour savoir le nombre d'élément du vecteur il faut utiliser la fonction **length()**

Les matrices sont des objets atomiques, c'est à dire de même mode ou type pour toutes les valeurs. La façon la plus simple pour créer une matrice sous R se fait par le biais de la fonction **matrix().**

```
> m <- matrix(c(1,2,3,4,5,6),ncol=2)
> m
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6

> m <- matrix(1:4,nrow=2,ncol=2)
> m
     [,1] [,2]
[1,]    1    3
[2,]    2    4

> m <- matrix(1:4,nrow=2,ncol=2, byrow=TRUE)
> m
     [,1] [,2]
[1,]    1    2
[2,]    3    4
```

Lorsque la longueur du vecteur est différente du nombre d'éléments de la matrice, R remplit toute la matrice.

```
> m <- matrix(1:4,nrow=3,ncol=3)
Warning message:
In matrix(1:4, nrow = 3, ncol = 3) :
  data length [4] is not a sub-multiple or multiple of the number of rows [3]
> m
     [,1] [,2] [,3]
[1,]    1    4    3
[2,]    2    1    4
[3,]    3    2    1
```

Calculs sur les matrices

Additions	M1+M2
Produit élément par élément	M1*M2
Puissance d'une matrice	M1^…

1.6) Les data-frames (Tableau individus*variables)

Le data-frame est la structure par excellence en statistique. Cette notion est exprimée dans R par le data.frame. Conceptuellement une data frame est une matrice dont les lignes sont des individus et les colonnes représentent des variables quantitatives et/ou qualitatives mesurées sur les mêmes individus.

La principale méthode pour créer un data-frame consistent à utiliser la fonction **data.frame** :

```
> vec1 <- 1:5
> vec2 <- c("a","b","c","a","b")
> df <- data.frame(nom.var1=vec1, nom.var2= vec2)
> df
  nom.var1 nom.var2
1        1        a
2        2        b
3        3        c
4        4        a
5        5        b
```

Récapitulatif

Objet	Modes	Plusieurs modes dans le même objet ?
Vecteur	Numérique, caractère, complexe ou logique	Non
Facteur	Numérique ou caractère	Non
Tableau	Numérique, caractère, complexe ou logique	Non
Matrice	Numérique, caractère, complexe ou logique	Non
Tableau de données (dataframe)	Numérique, caractère, complexe ou logique	Oui
liste	Numérique, caractère, complexe, logique, fonction, expression	Non

1.7) Représenter les données

Les graphiques constituent souvent le point de départ d'une analyse statistique. Un des avantages de R est la facilite avec laquelle l'utilisateur peut produire des graphiques de toutes natures.

Fonction plot

La fonction **plot** est une fonction générique de R permettant de représenter tous les types de données

```
> x <- seq (0,1,length=50)
> y <- sin (2*pi*x)
> plot (x,y)
> plot (x,y,type='l')            # changer le type en ligne au lieu des points
> plot (x,y,type='l',col='red')
```

Fonction hist

Le caractère quantitatif continu est représenté graphiquement par l'histogramme. Dans lequel chaque classe est représentée par un rectangle dont la surface correspond à la fréquence relative.
Sur R pour représenter la distribution d'une variable continue, il existe des solutions classiques déjà programmées, la fonction **hist** en est un example.

Exemple :
On choisi 19 étudiants en L1 de l'université Claude Bernard (choix aléatoire) et on mesure leur poids :

```
> Poids <-
c(62.5,70.5,69.7,65,71.2,72.3,66.5,62.3,69.9,72.9,80,82,59,57.5,61,63.9,53.5,52,54)

> Departement <- c("M","I","M","E","E","D","M","I","M","I","E","D","D","I","M","I","I","D","E")

# 'M' pour Mathématique, 'I' pour Informatique, 'E' pour Economie, 'D' pour Droit

> hist(Poids) #pour tracer l'histogramme des poids
> hist(Poids,breaks=c(50,55,60,65,70,85)) # Breaks pour préciser les intervalles des poids
> ?hist #pour connaître tous les paramètres quand peut changer par exemple couleur 'col'
```

Fonction pie

Maintenant on veut voir comment les étudiants sont distribués selon les départements. Le département est une variable qualitatif donc pour la représenter il faut tracer un diagramme circulaire. La fonction **pie** sous R nous permet d'effectuer cette tâche :

```
> Departement
# Creer le tableau des effectif dans chaque departement
> Effectif_Table <- table(factor(Departement, levels = c("M","I","E","D" )))
> Effectif_Table
> pie(Effectif_Table)
```

1.8) Exercices

Exercice 1 : Représentation graphique d'une variable quantitative

*Dans R il y a un certain nombre de bases de données disponibles, il suffit de tapez **data()** pour voir la liste de base de données.*

A partir du jeu de données cars disponible sous R (utiliser **data(cars)** pour le charger) :

a) Calculer la moyenne, variance, médiane, écart-type des variables speed et dist.

b) Qu'est ce que fait la fonction **summary()** ?

c) Selon le type de variables à notre disposition quelle représentation graphique suggérez-vous? Tracer l'histogramme de la variable **speed** ? Ajouter un titre et des label aux axes x et y.

Exercice 2 : Représentation graphique d'une variable qualitatif

a) Charger le jeu de données iris.

b) Tracer le camembert pour la variable qualitative. Calculer la moyenne, variance, min, max pour les autres variables. Faire la moyenne et la variance par type de species. Que remarquer vous ?
Quelle est l'autre façon de représenter une variable qualitative ?

c) Tracer la boite à moustache de Sepal.Length en fonction de Species. Quel est le but de cette représentation ?

Exercice supplémentaire à faire et remplir le tableau de Termes à retenir a la fin :

1) Que renvoie cette instruction : 1:5^3 ?
2) Que renvoie cette instruction : (1:5)^3 ?
3) Que renvoie ces instructions : var<-5 ? var*3 ?

4) Créer un vecteur M=[10,20,30,40,50,60,70,80], par plusieurs méthode (3 méthodes).
5) Créer un vecteur N=[1,2,3,4,5,6,7,8], par au moins 4 méthodes.
6) Faire l'addition des deux vecteurs.
7) Donner les 6ème, 7ème éléments de M dans un nouveau vecteur ''x''. Donner le 3ème élément de N dans ''y''. Calculer la puissance de x en fonction de y.
8) Créer la matrix A des deux vecteurs M et N. Sélectionner les premier 3 élément se qui forme une matrice de 3 lignes et 2 colonnes. Puis ajouter une colonne [1,-1,1].
9) Créer la matrix B =, calculer : A+B et A*B, déterminant de A et B, l'inverse de la matrice A.

Termes à retenir

Objet	Symbole ou fonction
Flèches d'affectation d'un variable	
Récupérer la nature d'une objet	
Déterminer si un objet est de nature : Numérique, caractère, logique…	
Fonctions pour créer un vecteur	
Fonctions pour grouper de vecteurs lignes et colonnes	
Fonction pour créer une matrice	
Fonction pour calculer le déterminant	
Fonction pour calculer l'inverse	
Fonction pour tracer un histogramme	
Fonction pour tracer un camembert	
Fonction pour tracer un boxplot (boite à moustache)	
Fonction pour tracer un barplot	