

MST - Logiciel R

TP n° 1

Introduction au logiciel R

Installation du logiciel R, environnement de travail

1. Les fichiers nécessaires à l'installation du logiciel R sont distribués à partir du site Internet CRAN (Comprehensive R Archive Network) à l'adresse suivante :
`http://www.r-project.org/`
2. Sélectionner un miroir CRAN puis **télécharger** et **installer** le logiciel en fonction de votre système d'exploitation.
3. R démarre chaque session au sein d'un **répertoire de travail** créé par défaut. Par exemple, sous Windows : `C:\Users\Utilisateur\Documents`.
On peut connaître celui-ci grâce à la commande `getwd()` puis éventuellement le modifier par l'intermédiaire du menu Fichier ou à l'aide de la commande `setwd()` ; par exemple : `setwd("D:/Cours/MST/TP R")`
4. Lorsqu'on **quitte** une session R, le logiciel propose de sauvegarder l'ensemble du travail réalisé au sein de deux fichiers :
 - .RData : environnement de travail (objets R, variables, fonctions, données),
 - .Rhistory : historique des commandes exécutées.

Premiers pas sous R

1. Ouvrir le logiciel R. Dans une première étape, les commandes seront saisies dans la fenêtre R `console`.
2. R permet de réaliser les opérations habituelles d'une calculatrice à l'aide des opérateurs usuels : `+`, `-`, `*`, `/`, `exp`, `log`, `^`, etc. Exécutez par exemple les commandes suivantes :

```
> exp(log(5))^2
> 1/sqrt(2*pi)*exp(-1/2)
```
3. Pour affecter une valeur à un objet, on utilise indifféremment le symbole égal (`=`) ou une "flèche" à gauche (`<-`) :

```
> x = 45
> y <- 5
> Y <- 9
```

Remarques : R différencie minuscules et majuscules. Le séparateur décimal est le point.
4. Saisir le nom de la variable permet d'en lire le contenu :

```
> x
> y/x/Y
```
5. Il est possible de soumettre plusieurs commandes sur une même ligne :

```
> factorial(5) ; choose(32,4) ; round(pi,4) ; floor(sqrt(2))
```

6. Affecter une valeur à un objet et afficher simultanément le contenu :
`> (s = sum(1:10))`
7. Il est possible de naviguer dans l'historique des commandes précédemment exécutées à l'aide des flèches ↑ ou ↓ du clavier.

La fenêtre de script

En début de session R, il est vivement conseillé d'ouvrir une fenêtre de script afin de saisir les différentes commandes R à soumettre et de les gérer plus facilement : conserver les commandes importantes, effacer les commandes erronées ou inutiles, insérer des commentaires, exécuter plusieurs commandes simultanément, programmer ou écrire ses propres fonctions.

1. Ouvrir un **nouveau script**. Dans la fenêtre de script, écrire les deux expressions ci-dessous en langage R puis exécuter les.

$$\left(9^2 + \frac{19^2}{22}\right)^{\frac{1}{4}}; \frac{99^2}{2206\sqrt{2}}$$

Remarque. L'exécution d'une commande dans la fenêtre de script s'effectue à l'aide du raccourci clavier CTRL R (en situant le curseur n'importe où sur la commande). Pour soumettre plusieurs commandes simultanément, sélectionner l'intégralité des commandes puis CTRL R.

2. À l'aide du symbole #, ajouter un **commentaire** après chaque expression. Tout ce qui suit ce symbole est ignoré par R.
3. **Sauvegarder** le script dans le répertoire de travail courant puis fermer le script.
4. **Exécuter** l'intégralité du script à l'aide de la commande
`source("script.R", echo=T)`

Résumé numérique d'une petite série statistique

1. Saisir la série statistique ci-dessous (à noter la fonction `c` qui permet sous R de saisir une série de valeurs sous la forme d'un vecteur) :
`> age = c(26, 25, 23, 19, 22, 21, 26, 27, 22, 18, 31, 32, 38, 27, 24, 25)`
2. Lister le contenu du vecteur `age` :
`> age`
3. Il est très simple de transformer les valeurs d'un vecteur :
`> age^2 ; sqrt(age) ; log(age)`
4. Sélectionner des valeurs de la série par leurs indices, par des conditions logiques :
`> age[3] ; age[10:12] ; age[c(2,4,6)] ; age[-(1:5)] ; jeune = age[age<20]`
5. Les commandes suivantes permettent successivement de donner le nombre de valeurs de la série, de les ordonner, d'en obtenir le min, le max, la somme, la moyenne, la médiane, l'écart-type et la variance :
`> length(age) ; sort(age) ; min(age) ; max(age) ; sum(age)`
`> mean(age) ; median(age) ; sd(age) ; var(age)`
 Remarque : la variance calculée par R correspond à l'estimation non biaisé de la variance (avec `sd = sqrt(var)`).
6. Soumettre la fonction générique `summary(age)` qui produit un résumé statistique succinct des valeurs de la série.
7. Que représente la valeur suivante ?
`> mean(age^2) - mean(age)^2`

Représentation graphique d'une série statistique

1. La commande générique `plot` produit un graphe dit indexé. Les valeurs sont tracées selon leur rang (ou indice) dans la série :


```
> plot(age)
```

 Reproduire le même graphe, mais avec les valeurs triées.
 Modifier le type de tracé des valeurs en ajoutant à la fonction l'argument `type` :


```
> plot(sort(age), type="h")
```
2. Une représentation plus classique des valeurs individuelles : représentation axiale (ou nuage de points) :


```
> stripchart(age)
```
3. Comparer le graphe précédent avec celui fourni par la commande suivante :


```
> stripchart(age, method="stack", pch=20, col="blue")
```
4. Construire un histogramme :


```
> hist(age)
```
5. Un histogramme plus joli :


```
> hist(age, col="steelblue4", border="white", main="Histogramme de la variable Age", xlab="Age de l'enquêté", ylab = "effectif")
```

 Remarque : la commande `colors()` affiche toutes les couleurs disponibles sous R.
6. Un histogramme avec découpage en classes d'amplitudes inégales :


```
> hist(age, breaks=c(18,20,24,27,32,38), main="Histogramme de la variable Age", col="steelblue4", border="white", xlab="Age de l'enquêté", ylab = "densité")
```
7. Tracer une boîte à moustaches :


```
> boxplot(age)
```
8. Une belle boîte à moustaches rose et cintrée :


```
> boxplot(age, col="pink", horizontal=TRUE, notch=TRUE, xlab="Age de l'enquêté")
```
9. Le paramètre graphique `mfrow` permet de partitionner une feuille graphique ; par exemple :


```
> par(mfrow=c(3,3))
```

 Plus précisément, on a ici la possibilité d'avoir 9 graphiques disposés selon 3 lignes et 3 colonnes. Après avoir soumis la commande précédente, représenter tous les graphiques précédents dans une même fenêtre.

L'installation de nouveaux packages (ou librairies)

Pour des réaliser des traitements statistiques spécifiques (analyse de données, tests d'hypothèses, etc.) ou pour appliquer des méthodes spécifiques à un domaine d'application (économétrie, données de survie, etc.), il est parfois nécessaire d'installer de nouvelles librairies. Cela se fait en deux temps :

1. **Installation** du package (en le téléchargeant depuis un miroir CRAN). Par exemple :


```
> install.packages("FactoMineR", dependencies=TRUE)
```
2. **Chargement** du package pour qu'il puisse être utilisé dans la session courante :


```
> library(FactoMineR)
```

L'étape d'installation (téléchargement) est faite une fois pour toutes, mais le chargement de la librairie doit être effectuée à chaque nouvelle session de R. Ces deux opérations peuvent être réalisées à partir du menu **Packages**.

Application : installer puis charger la librairie `e1071` de R.

Retour sur l'exercice 1 du TD 1

Retrouver les principaux résultats de l'exercice 1 du TD n°1 pour la série statistique :

$$x = c(6.11, 1.14, 4.76, 1.41, 6.83, 4.17, 7.16, 4.28, 14.84, 17.4)$$

1. Trier la série puis calculer la moyenne, les quartiles et la variance de x .
2. Produire un graphe indexé puis une représentation axiale (nuage de points) des données.
3. Calculer une estimation non biaisée du coefficient d'asymétrie puis du coefficient d'aplatissement.
4. Retrouver les valeurs des indicateurs précédents à l'aide des fonctions `skewness` et `kurtosis` de la librairie `e1071` :
> `skewness(x, type=2)`
> `kurtosis(x, type=2)`
5. Représenter la fonction de répartition empirique $F_n(x)$ à l'aide de la commande :
> `plot(ecdf(x))`
6. Tracer la boîte à moustaches et identifier les éventuelles observations extrêmes.

Quelques commandes utiles

L'aide dans R

1. Pour obtenir de l'**aide** sur une fonction, on la fait précéder d'un point d'interrogation :
> `?boxplot`
On y trouve la liste de ses arguments, leur valeur par défaut ainsi que des exemples d'utilisation de la commande.
2. Dans le cas de graphiques, on peut également obtenir et afficher une série d'exemples :
> `example(boxplot)`
3. Plus généralement, une aide en ligne est obtenue grâce à la commande
> `help.start()`

La gestion des fenêtres graphiques

1. Pour visualiser plusieurs graphiques simultanément, il est possible d'ouvrir une nouvelle fenêtre graphique à l'aide de la commande :
> `x11()`
La dernière créée est toujours la fenêtre dite active. Pour utiliser une ancienne fenêtre (par exemple la n°3), on doit la rendre à nouveau active :
> `set.dev(3)`
2. Un graphique R peut être copié-collé dans un document du type Open Office ou Word ou encore exporté en format vectoriel ou postscript.