L'Analyse Des Données

S1 - Module M5

Master : Economie Appliquée

Chapitre 0: Le LOGICIEL R

Driss TOUIJAR

Faculté des Sc. Juridiques, Economiques et Sociales
Département des Sc Economiques et de Gestion- Fès

30 octobre 2017

References I

- Pierre-André CORNILLON, Arnaud GUYADER, and François HUSSON. *Statistiques avec R*. Paris : PUR cop edition, 2012.
- François HUSSON, SEBASTIEN LE, and J PAGES. *Analyse de données avec R*. Paris : PUR cop edition, 2009.
- Pierre LAFAYE de MICHEAUX, Rémy DROUILHET, and Benoit LIQUET. *Le logiciel R - Maîtriser le langage - Effectuer des analyses (bio)Statistiques*. Springer-Verlag 2 ème edition, 2014.
- Driss TOUIJAR. *Statistique Descriptive Cours, Exercices et Examens corrigés, avec mise en oeuvre sous R*. Octobre 2016.

Table des matières I

1. Introduction Générale
2. Les premiers pas avec R
   - Présentation de R
   - Les fenêtres de R
     - La console
     - La fenêtre graphique
     - fenêtre éditeur de données
     - fenêtre éditeur script
     - fenêtre Rcommander
   - Les Aides sous R
     - L'Aide du Menu
     - Fonction help.start()
     - Fonction help()
   - R comme calculette
3. R et la statistique
4. Les objets de R

Table des matières II

- Les fonctions de R
- les data.frames :
- Les listes

5 Les packages de R

6 Tableau des principales fonctions de R

Avertissement

Ce document est un support de Cours et non pas Le Cours.

Par conséquent, votre présence aux séances du cours est indispensable pour mieux cerner le programme de l'Analyse des Données avec R...

C'est quoi R ?

R est un logiciel de statistique gratuit et libre et open source créé en 1996 par deux universitaires du département de statistique de l'université d'Auckland en Nouvelle-Zélande : Robert Gentleman et Ross Ihaka connus comme "R & R"

C'est quoi R ?

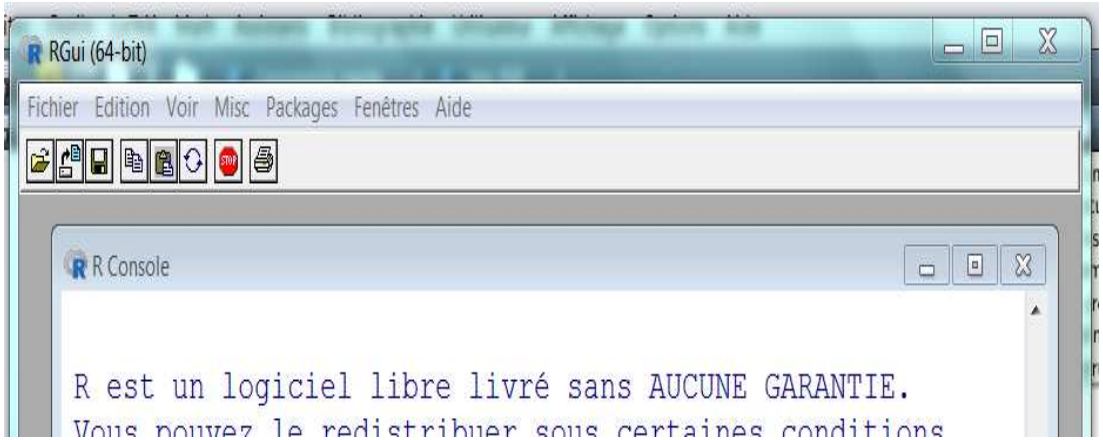
R est un logiciel modulaire . Il est plus qu'un langage de programmation, c'est un environnement mathématique utilisé pour le traitement de données et l'analyse statistique et est multi-plateforme : on peut l'installer sur UNIX, MACosX et Windows. Il évolue plusieurs fois plus vite que les autres logiciels commercialisés.

Comment installer R ?

Il suffit, d'aller sur l'adresse http://CRAN.R-project.org/ où CRAN (Comprehensive R Archive Network) distribue R gratuitement. Vous y trouverez un programme d'installation de R pour chaque plateforme. Par exemple si vous utilisez Windows, il faut télécharger le programme correspondant. L'installation sur votre ordinateur devient très facile si vous suivez les instructions.

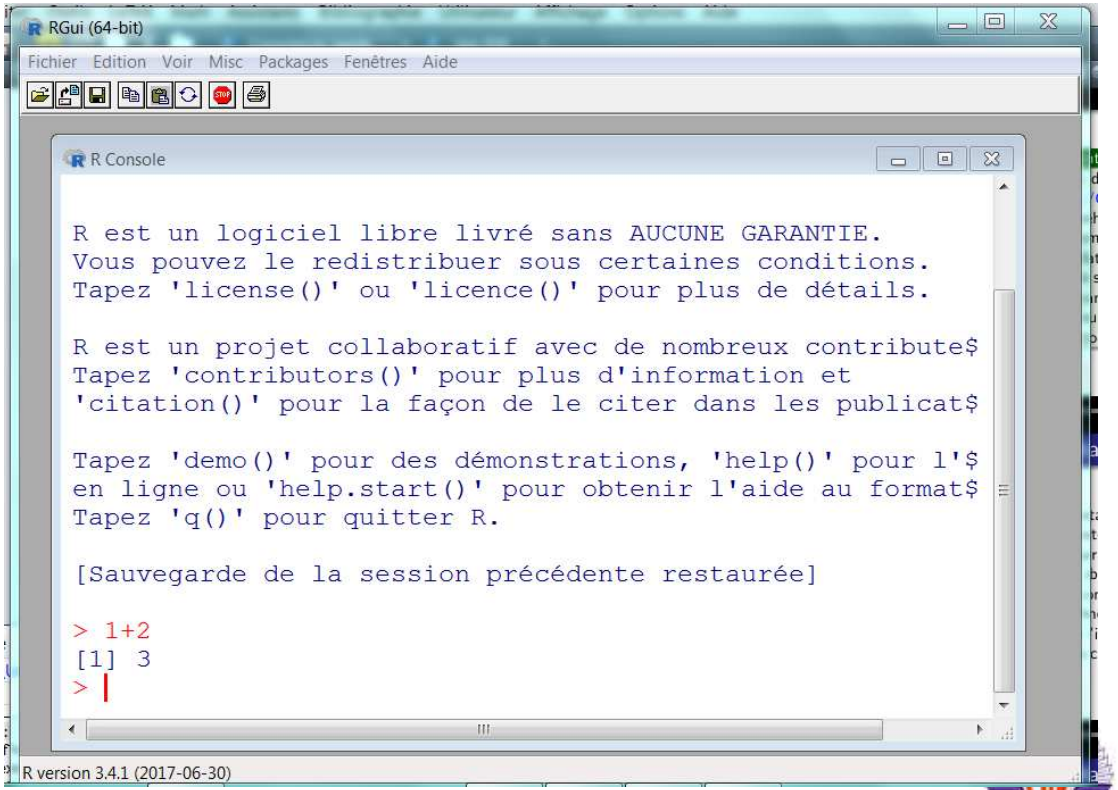
Présentation de R

Une fois installé sur votre ordinateur, vous devez cliquer sur l'icône de R pour obtenir la fenêtre ci-contre.
La petite fenêtre se trouvant à l'intérieur s'appelle la console.



Présentation de R

L'invite (le signe supérieur >) en rouge au début de la ligne veut dire que R est prêt à recevoir vos ordres (les commandes) qui seront à la suite affichés en rouge. Alors que les réponses systèmes seront affichées en bleu.



Plan

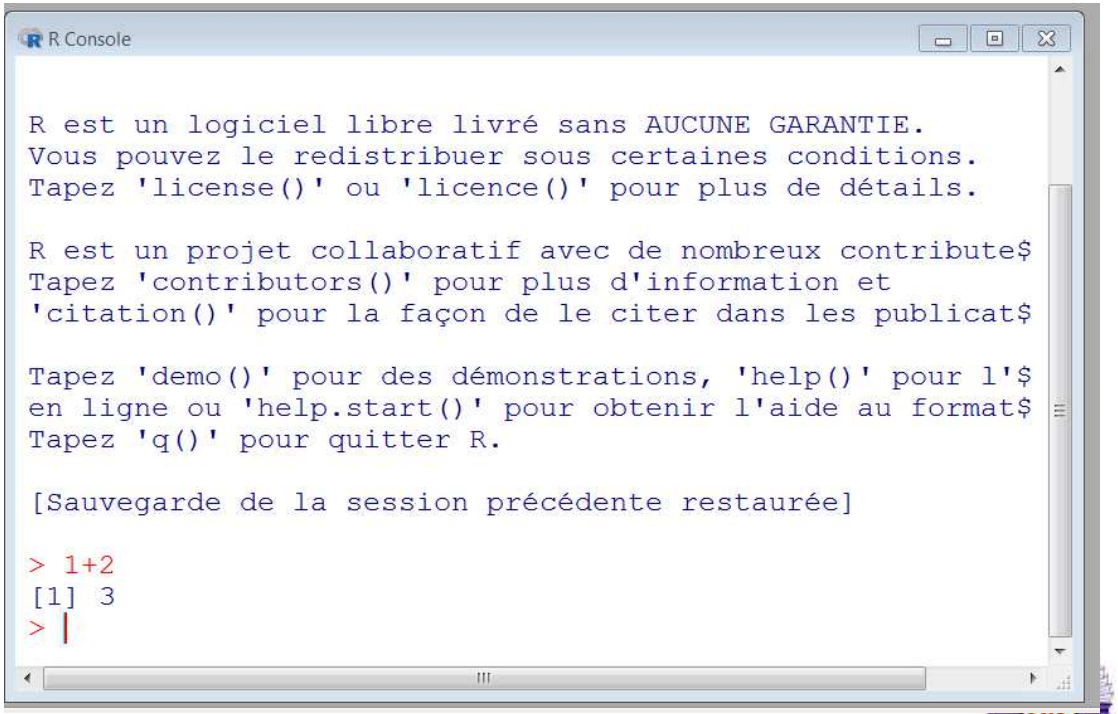
1. Introduction Générale
2. Les premiers pas avec R
   - Présentation de R
   - Les fenêtres de R
     - La console
     - La fenêtre graphique
     - fenêtre éditeur de données
     - fenêtre éditeur script
     - fenêtre Rcommander
   - Les Aides sous R
     - L'Aide du Menu
     - Fonction help.start()
     - Fonction help()
   - R comme calculette
3. R et la statistique
4. Les objets de R
   - Les fonctions de R

La console

R dispose de plusieurs types de fenêtres :

La Console

C'est la fenêtre principale.
C'est là qu'on saisi les commandes une par une (voir ci-contre).

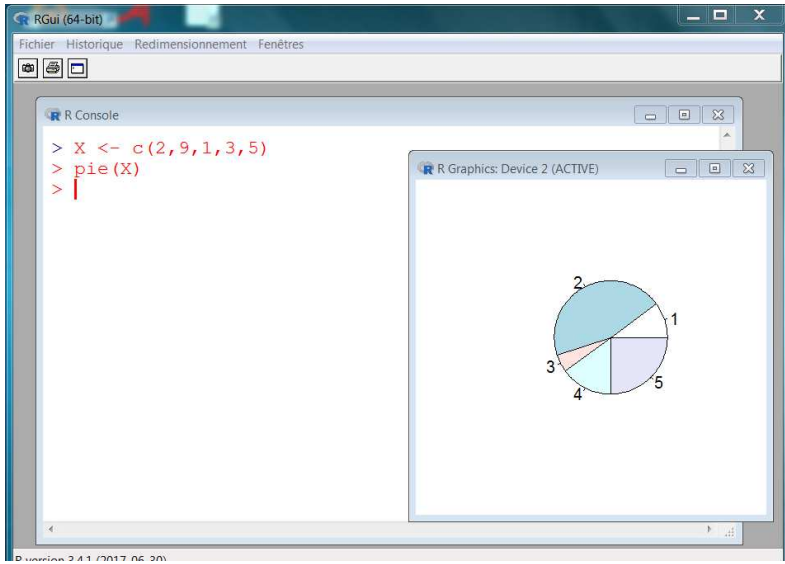


Plan

1. Introduction Générale
2. Les premiers pas avec R
   - Présentation de R
   - Les fenêtres de R
     - La console
     - La fenêtre graphique
     - fenêtre éditeur de données
     - fenêtre éditeur script
     - fenêtre Rcommander
   - Les Aides sous R
     - L'Aide du Menu
     - Fonction help.start()
     - Fonction help()
   - R comme calculette
3. R et la statistique
4. Les objets de R
   - Les fonctions de R

La fenêtre graphique

On l'obtient à la suite d'une commande graphique (*pie(), hist(), plot(), curve(), lines(),. . .*) (voir ci-contre).

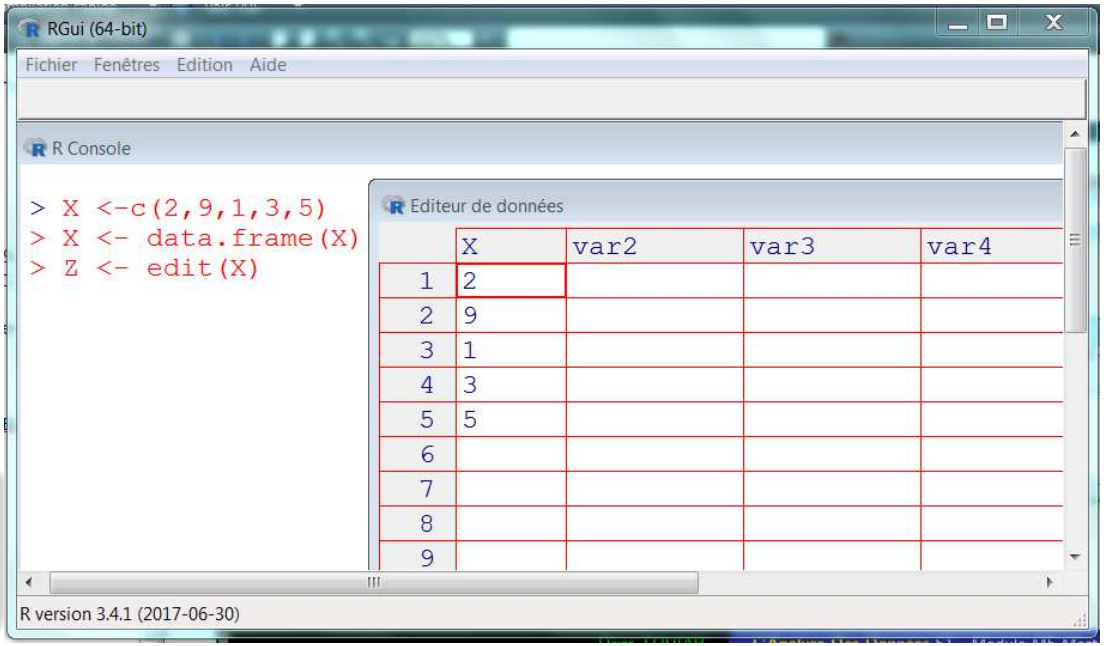


Plan

1. Introduction Générale
2. Les premiers pas avec R
   - Présentation de R
   - Les fenêtres de R
     - La console
     - La fenêtre graphique
     - fenêtre éditeur de données
     - fenêtre éditeur script
     - fenêtre Rcommander
   - Les Aides sous R
     - L'Aide du Menu
     - Fonction help.start()
     - Fonction help()
   - R comme calculette
3. R et la statistique
4. Les objets de R
   - Les fonctions de R

La fenêtre éditeur de données

Elle ressemble à une feuille d'un tableur tel que " Excel ". Elle est obtenue lorsqu'on veut visualiser les données et/ou les modifier (voir ci-contre).



Exemple

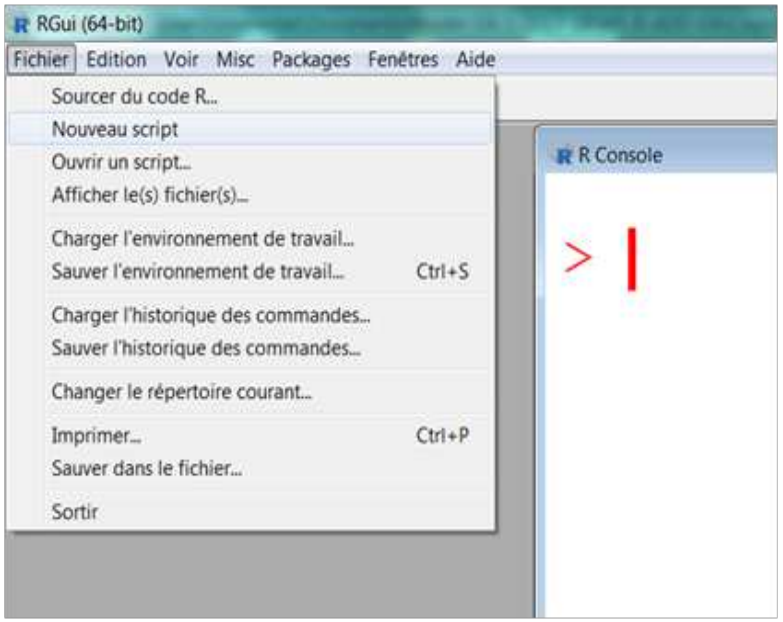
> z <− edit(X)

Plan

1. Introduction Générale
2. Les premiers pas avec R
   - Présentation de R
   - Les fenêtres de R
     - La console
     - La fenêtre graphique
     - fenêtre éditeur de données
     - fenêtre éditeur script
     - fenêtre Rcommander
   - Les Aides sous R
     - L'Aide du Menu
     - Fonction help.start()
     - Fonction help()
   - R comme calculette
3. R et la statistique
4. Les objets de R
   - Les fonctions de R

La fenêtre éditeur script

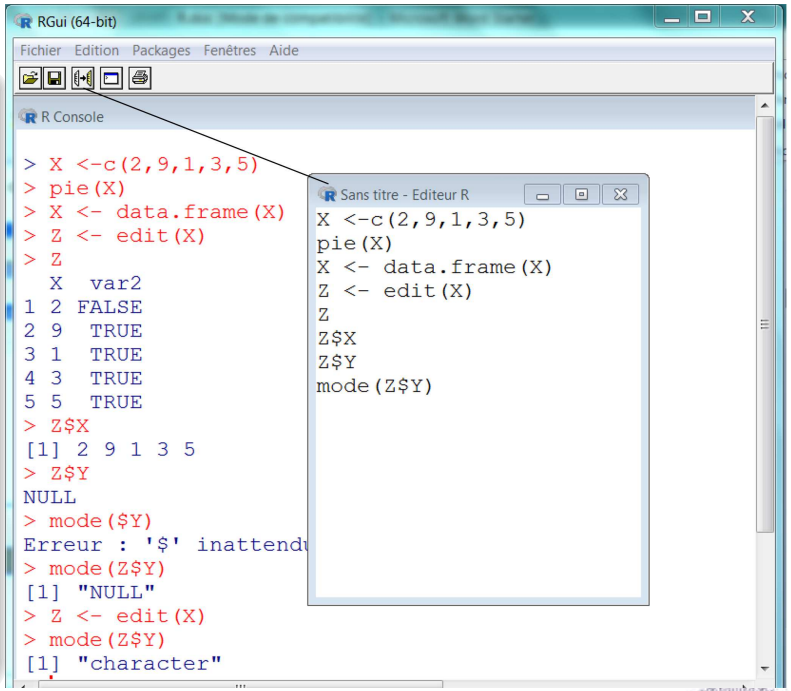
Elle sert à " automatiser " le travail sous R. Vous pouvez toujours exécuter les commandes aux quelles vous aurez souvent recours, sans les retaper, en modifiant seulement les paramètres et/ou les données . Une fois le programme écrit, vous l'enregistrez sous un nom avec l'extension " .R ", par exemple :
" monfich.R ". (voir ci-contre).



La fenêtre éditeur script

Remarque :

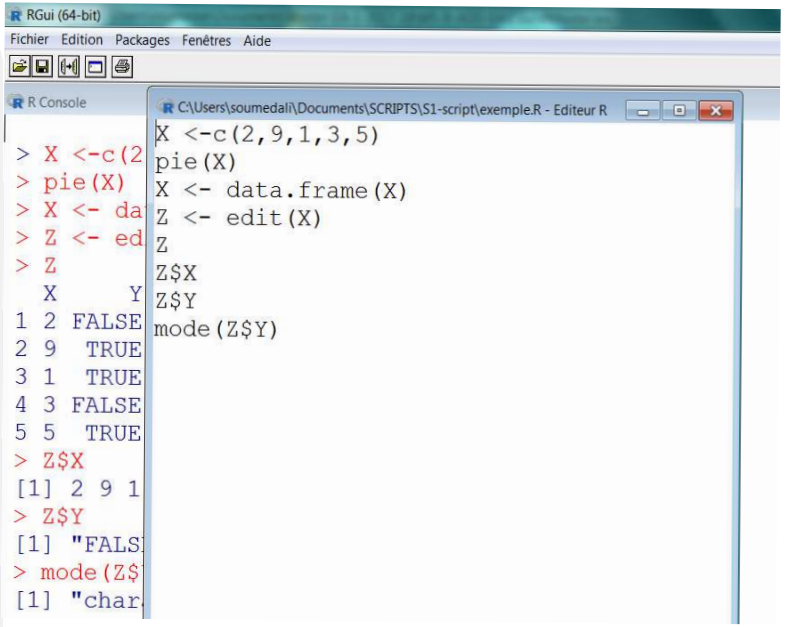
Tous les programmes de mon livre[4] peuvent être copiés (sans les invites(>) ni les (+) se trouvant aux débuts des lignes de commandes) puis collés dans un nouveau fichier « script ». Enfin, on peut exécuter ligne par ligne ou l'ensemble du programme sélectionné.



La fenêtre éditeur script

Remarque :

Si maintenant vous voulez réexécuter votre script, il suffit de cliquer sur le menu ≪Fichier≫ puis sur ≪Ouvrir un script. . . ≫ et vous sélectionnez votre fichier, vous l'ouvrez, vous sélectionnez la (ou les) lignes que vous voulez exécuter (comme indiqué ci-contre) et c'est parti. . .



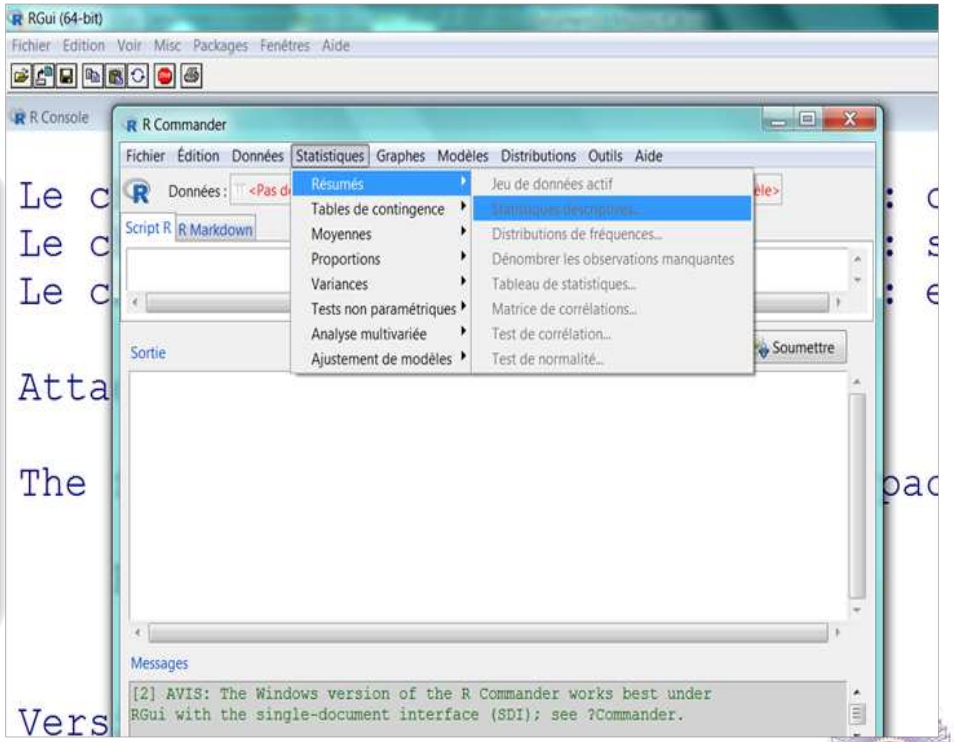
Plan

1. Introduction Générale
2. Les premiers pas avec R
   - Présentation de R
   - Les fenêtres de R
     - La console
     - La fenêtre graphique
     - fenêtre éditeur de données
     - fenêtre éditeur script
     - fenêtre Rcommander
   - Les Aides sous R
     - L'Aide du Menu
     - Fonction help.start()
     - Fonction help()
   - R comme calculette
3. R et la statistique
4. Les objets de R
   - Les fonctions de R

La fenêtre Rcommander

Remarque :

Il existe une autre fenêtre «Rcommander» abrégée en «Rcmdr» qui n'est autre qu'une interface graphique de R mais qui n'est pas très développée.



Aide

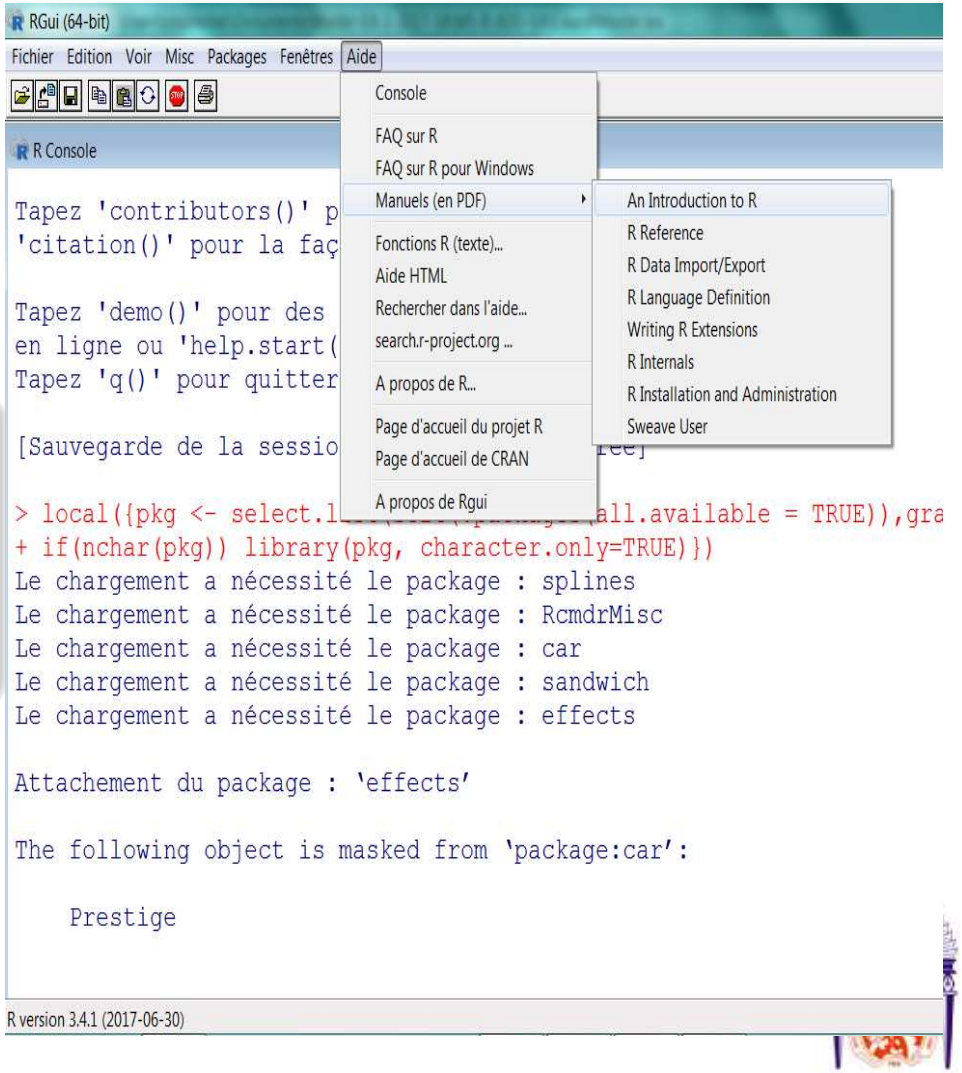
R vous propose plusieurs types d'aides sans connexion...

Plan

1. Introduction Générale
2. Les premiers pas avec R
   - Présentation de R
   - Les fenêtres de R
     - La console
     - La fenêtre graphique
     - fenêtre éditeur de données
     - fenêtre éditeur script
     - fenêtre Rcommander
   - Les Aides sous R
     - L'Aide du Menu
     - Fonction help.start()
     - Fonction help()
   - R comme calculette
3. R et la statistique
4. Les objets de R
   - Les fonctions de R

An Introduction to R

C'est un fichier pdf très pratique pour les premiers pas, mais il est écrit en Anglais !

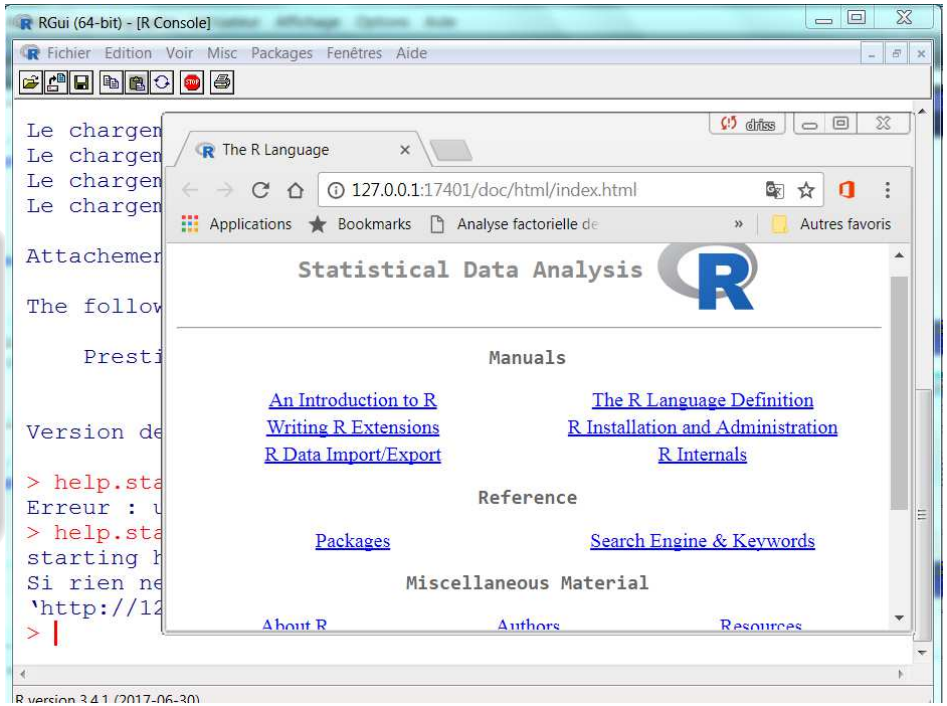


Plan

1. Introduction Générale
2. Les premiers pas avec R
   - Présentation de R
   - Les fenêtres de R
     - La console
     - La fenêtre graphique
     - fenêtre éditeur de données
     - fenêtre éditeur script
     - fenêtre Rcommander
   - Les Aides sous R
     - L'Aide du Menu
     - Fonction help.start()
     - Fonction help()
   - R comme calculette
3. R et la statistique
4. Les objets de R
   - Les fonctions de R

help.start()

Vous donne un help sur tout ce qui concerne le logiciel R à l'aide d'un navigateur au format HTML :



Plan

1. Introduction Générale
2. Les premiers pas avec R
   - Présentation de R
   - Les fenêtres de R
     - La console
     - La fenêtre graphique
     - fenêtre éditeur de données
     - fenêtre éditeur script
     - fenêtre Rcommander
   - Les Aides sous R
     - L'Aide du Menu
     - Fonction help.start()
     - Fonction help()
   - R comme calculette
3. R et la statistique
4. Les objets de R
   - Les fonctions de R

help()

L'aide sur n'importe quelle fonction (par exemple la moyenne) de R est obtenue (en anglais) par la commande :

> help(mean)

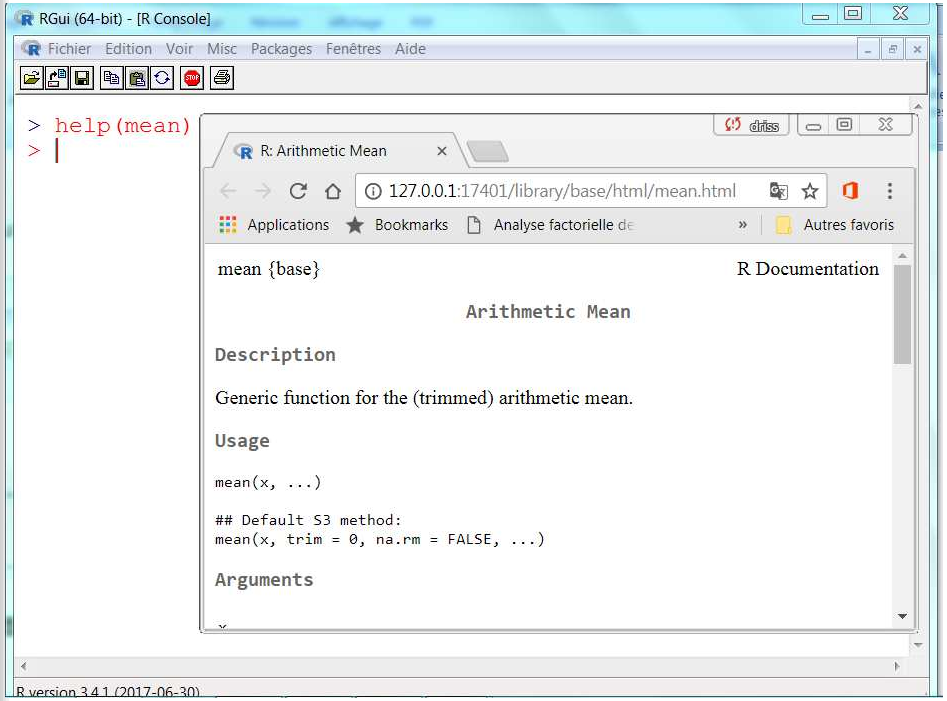
Ou simplement :

> ? mean

D'autres exemples :

> ? "&" : l'aide sur l'operateur logique "et"

> ? "|" : l'aide sur l'operateur logique "ou"



Avec R, on peut faire de simples calculs comme on peut faire des calculs très compliqués :

```
> 1+2^5
[1] 33
```

On obtient le résultat 33 après avoir appuyé sur la touche entrée (ou retour chariot). Si la valeur 33 n'est pas affectée à un objet de R, par exemple une lettre ≪ x ≫, on la ≪ perd ≫. Par contre, une fois stockée dans ≪ x ≫ , on peut toujours l'utiliser dans d'autres formules :

Calculs Simples

```
> x <- 1+2^5 # affectation du nombre à l'objet x
> y <- -2*sqrt(3)*exp(3/x)
> y
[1] -3.793778
> z <- x+y
> z
[1] 29.20622
> pi
[1] 3.141593
> sin(pi/4)
[1] 0.7071068
> sinpi(1/4) # même chose que sin(pi/4)
[1] 0.7071068
```

Calculs Logiques

```
> 2==3

[1] FALSE

> 2==2

[1] TRUE

> b=2>3; b # «;» permet de mettre plusieurs commandes.

[1] FALSE

> x <- c(FALSE, TRUE)
> outer(x,x,"|") #  OR table: Table de vérité (ou)

      [,1] [,2]
[1,] FALSE TRUE
[2,]  TRUE TRUE
```

```
> a="stat" ; a

[1] "stat"
```

Calcusl vectoriels et vectorisés :

Avec R, le calcul vectoriel et matriciel est très simple et prend plusieurs formes :

```
> x <- c(3,-1,.2,sqrt(2),21)
> x

[1]  3.000000 -1.000000  0.200000  1.414214 21.000000
```

1

1. où c() est une fonction appelée collecteur

On complique un petit peu les choses :

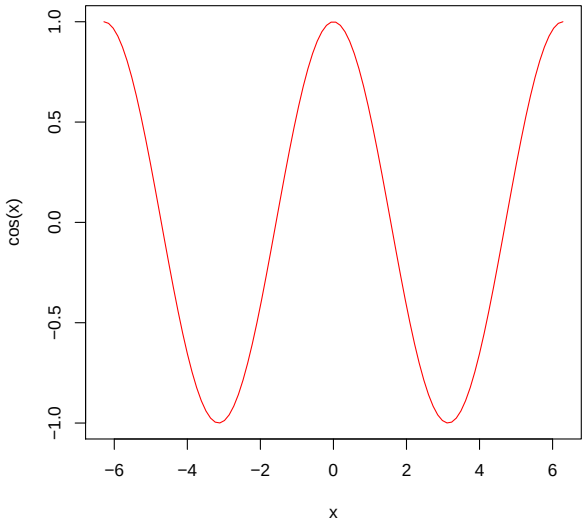
```
> x <- c(3,-1,.2,sqrt(2))
> x
[1]  3.000000 -1.000000  0.200000  1.414214
> y=0:2
> z=c(35,y,c(-6,6),x)
> round(z,1) # arrondir à la première décimale
 [1] 35.0  0.0  1.0  2.0 -6.0  6.0  3.0 -1.0  0.2  1.4
> z[9] # donne le neuvième élément du vecteur z
[1] 0.2
```

Pour écrire une suite arithmétique de 1^{er} terme 5 et de raison -1 :

```
> x <- 5:-5 ; x
 [1]  5  4  3  2  1  0 -1 -2 -3 -4 -5
```

Si on veut tracer une fonction, on a besoin parfois de la fonction seq()

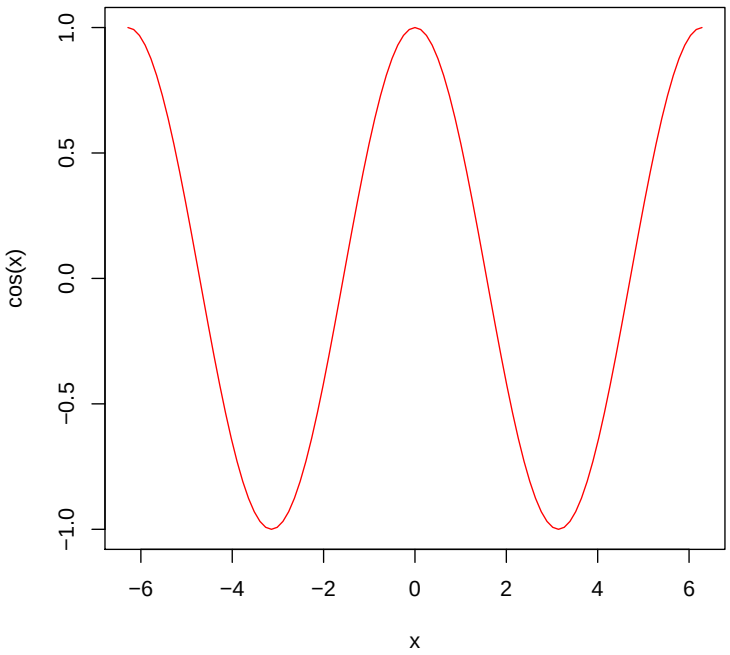
```
> x=seq(-2*pi,2*pi,length=100) #on crée 100  points
> head(x) # affiche quelques premières valeurs de x

[1] -6.283185 -6.156252 -6.029319 -5.902386 -5.775453 -5.648520

> plot(x,cos(x),type="l",col="red")
```



Remarque

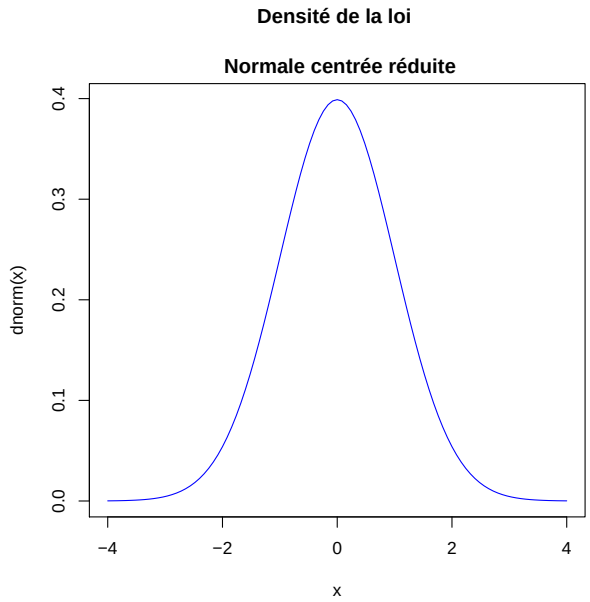
On peut utiliser des commandes plus simples pour avoir le résultat précédent

```
> curve(cos,-2*pi,2*pi,col=2)
```


cos(x)
1.0
0.5
0.0
−0.5
−1.0
−6
−4
−2
0
2
4
6
x

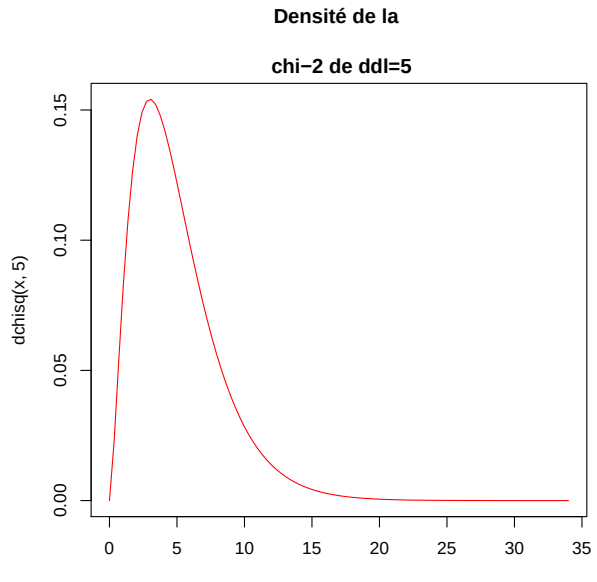
Tracé de la densité de la loi normale

```
> curve(dnorm,-4,4,col=4,
+   main="Densité de la loi \n
+   Normale centrée réduite")
```



Tracé de la densité de la loi χ^2

```
> x <- rchisq(100,5)
> curve(dchisq(x,5),0,34,100,
+   col=2,main="Densité de la \n
+   chi-2 de ddl=5")
```



Revenons à nos calculs :

```
> x = rep(5,7)# on répète « cinq » sept fois.
> x # est donc un vecteur de 7 valeurs : des 5

[1] 5 5 5 5 5 5 5

> X <- 1:5 + 1 ; X

[1] 2 3 4 5 6

> X <- array(c(1,2,3,4,5,6),dim=c(2,3)) ; X #

     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

2

2. array() : tableau de dimension m, si m est un vecteur de longueur 2 (m=c(i, j)), on obtient une matrice

D'une autre façon, on peut utiliser la notion de matrice :

Les Matrices

```
> X <- matrix(c(1,2,3,4,5,6),ncol=3)
> X
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> X[1,2] # élément de ligne1 et colonne2
[1] 3
```

3

3. Attention, sous R, la majuscule diffère de la minuscule x≠X

On peut changer n'importe qu'elle valeur de la matrice, par exemple on change la valeur 4 en 7 :

```
> X <- matrix(c(1,2,3,4,5,6),ncol=3)
> X[2,2] <- 7 ; X
```

```
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    7    6
```

Prenons un échantillon d'opérations dans le cas des matrices carrées (nombre de colonnes égale au nombre de lignes) :

```
> D=diag(rep(3,5),ncol=5);D # matrice diagonale 5 fois des 3
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    3    0    0    0    0
[2,]    0    3    0    0    0
[3,]    0    0    3    0    0
[4,]    0    0    0    3    0
[5,]    0    0    0    0    3
```

Changeons la diagonale par : 1, 2, 3, 4 et 5

```
> D=diag(rep(3,5))
> diag(D)<- 1:5 ; D
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    0    0    0    0
[2,]    0    2    0    0    0
[3,]    0    0    3    0    0
[4,]    0    0    0    4    0
[5,]    0    0    0    0    5
```

Inversons cette matrice et appelons-la : « Inv »

```
> Inv<- solve(D);Inv
```

```
     [,1] [,2]      [,3] [,4] [,5]
[1,]    1  0.0 0.0000000 0.00  0.0
[2,]    0  0.5 0.0000000 0.00  0.0
[3,]    0  0.0 0.3333333 0.00  0.0
[4,]    0  0.0 0.0000000 0.25  0.0
[5,]    0  0.0 0.0000000 0.00  0.2
```

Vérifions maintenant que : $D \times Inv = I$

En effet :

```
> D%*%Inv  # où  %*%  est le produit matriciel
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    0    0    0    0
[2,]    0    1    0    0    0
[3,]    0    0    1    0    0
[4,]    0    0    0    1    0
[5,]    0    0    0    0    1
```

Posons maintenant M une matrice telle que :

```
> M=matrix(c(6,0,2,1,8,6,8,2,2,3,
+ 2,3,1,3,4,4,1,5,8,4,5,1,1,1,9),ncol=5)
>  M
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    6    6    2    4    5
[2,]    0    8    3    1    1
[3,]    2    2    1    5    1
[4,]    1    2    3    8    1
[5,]    8    3    4    4    9
```

```
> det(M) # le déterminant de la matrice M
```

```
[1] -36
```

```
> (N <- D+M)
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    7    6    2    4    5
[2,]    0   10    3    1    1
[3,]    2    2    4    5    1
[4,]    1    2    3   12    1
[5,]    8    3    4    4   14
```

Problème : Supposons qu'on veuille résoudre le système linéaire suivant :

$$\begin{cases} 7x_1 + 6x_2 + 2x_3 + 4x_4 + 5x_1 = 1 \\ 10x_2 + 3x_3 + x_4 + x_5 = 8 \\ 2x_1 + 2x_2 + 4x_3 + 5x_4 + x_5 = 6 \\ x_1 + 2x_2 + 3x_3 + 12x_4 + x_5 = 0 \\ 8x_1 + 3x_2 + 4x_3 + 4x_4 + 14x_5 = 1 \end{cases}$$ où

$$X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix}$$

On peut réécrire ce système comme suit : $NX = B$ où N est la matrice ci-dessus,et où $B = \begin{pmatrix} 1 \\ 8 \\ 6 \\ 0 \end{pmatrix}$

Mise en oeuvre sous R :

```
> B = c(1,8,6,0,1)
> solve(N,B)
```

```
[1] -0.08234979  0.23057940  2.21770386 -0.55209227 -0.4068
```

ça y est c'est fait ! Vérifions maintenant que $NX = B$:

```
> X <- solve(N,B) # on stocke la solution dans l'objet $X$
> N%*%X # on retrouve B où 2.775558e-16 estquasi-nul
```

```
              [,1]
[1,] 1.000000e+00
[2,] 8.000000e+00
[3,] 6.000000e+00
[4,] 2.775558e-16
[5,] 1.000000e+00
```

Enfin, on peut afficher Les Matrices ligne par ligne (respectivement colonne par colonne) :

```
> N[3,]  #  affiche la ligne 3 de la matrice N

[1] 2 2 4 5 1

> N[,2]  #  affiche la colonne 2 de la matrice N

[1]  6 10  2  2  3
```

Exemple

On se propose de faire une petite étude fictive sur 7 étudiants selon deux critères la note (sur 20) en statistique et le sexe de l'étudiant. Ici on se contente de saisir les données à **la main** :

```
> note <- c(11,15,03,08,18,10,13)
> length(note) # le nombre de notes ou la taille de la popu

[1] 7

> sexe<-factor(c("M","F","F","F","M","F","M")) #
> # M désigne masculin et F féminin
> sexe

[1] M F F F M F M
Levels: F M
```

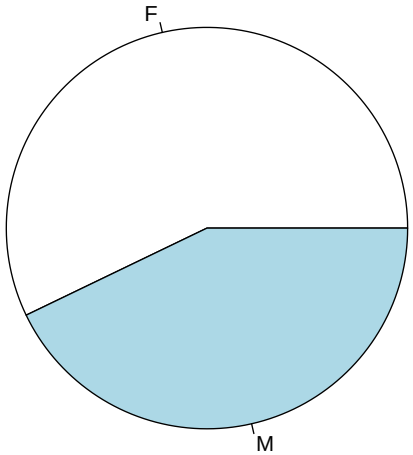
Remarques

La note ainsi définie est considérée comme une série statistique brute. De même pour la série sexe.

On commence par étudier, individuellement, chacune des deux séries statistiques :

```
> mean(note)
[1] 11.14286
> var(note)
[1] 23.80952
> max(note)
[1] 18
```

```
> table(sexe)

sexe
F M
4 3

> pie(table(sexe))
```



Maintenant, on refait l'étude pour les deux séries en même temps :

```
> tableau<-data.frame(note,sexe)
> tableau
  note sexe
1   11    M
2   15    F
3    3    F
4    8    F
5   18    M
6   10    F
7   13    M
```

```
> summary(tableau)
      note        sexe
 Min.   : 3.00   F:4
 1st Qu.: 9.00   M:3
 Median :11.00
 Mean   :11.14
 3rd Qu.:14.00
 Max.   :18.00
> rownames(tableau) <- c("Ahmed","Fati","ayah",
+ "lina","Ali","Mie","Jad")
> tableau
```

```
      note sexe
Ahmed   11    M
Fati    15    F
ayah     3    F
lina     8    F
Ali     18    M
Mie     10    F
Jad     13    M
```

```
> tableau["Ali",]
```

```
    note sexe
Ali   18    M
```

```
> tableau["Ali","note"]
```

```
[1] 18
```

Pour afficher les étudiants n'ayant pas la moyenne :

```
> rownames(tableau[which(tableau$note<10),])
[1] "ayah" "lina"
```

Pour afficher les étudiants ayant la moyenne :

```
> rownames(tableau[which(tableau$note>=10),])
[1] "Ahmed" "Fati"  "Ali"   "Mie"   "Jad"
```

On peut aussi facilement croiser les deux caractères et en déduire un tableau de contingence

```
> table(note,sexe)

     sexe
note F M
  3  1 0
  8  1 0
  10 1 0
  11 0 1
  13 0 1
  15 1 0
  18 0 1
```

Il y a plusieurs types d'objets, on se contentera ici d'en citer quelques uns :

Les fonctions statistiques sous R sont des caractéristiques telle quemoyenne, mode, médiane, variance, étendue, max... ; ou des fonctions mathématiques : log, exp, sin, cos, racine,...
On les reconnaît comme des noms qui se terminent toujours par une parenthèse, exemple : c(), mean(), plot(), matrix(),...
A l'intérieur des parenthèses, on trouve les arguments de la fonction. Pour les fixer, on peut avoir recours à l'aide :

```
> ?plot # ici il ne faut pas ajouter des parenthèses à la f
```

Ces fonctions sont prédéfinies, mais elles peuvent aussi être programmées.

En statistique, un objet data.frame est un tableau de séries brutes (une ou plusieurs) qualitatives et/ou quantitatives définies sur une même population, donc sur les mêmes individus ; autrement dit les séries ont la même longueur.
Pour l'exemple, on reprend les séries note et sexe. On les avait déclaré comme objet : ≪ data.frame ≫ et on l'a nommé ≪ tableau ≫ . Pour en être sûr, on utilise la fonction data.frame class() :

```
> class(note)
[1] "numeric"
> class(sexe)
[1] "factor"
> class(tableau)
```

```
[1] "data.frame"
```

Pour connaître les noms de toutes les séries que contient le data.frame :

```
> names(tableau)
```

```
[1] "note" "sexe"
```

Et si on veut renommer (ou nommer) des séries :

```
> names(tableau)<- c("X1","X2")
> tableau
```

```
  X1 X2
1 11  M
2 15  F
3  3  F
4  8  F
5 18  M
6 10  F
7 13  M
```

Maintenant, si on veut les consulter :

```
> X1

Erreur : objet 'X1' introuvable
```

Par contre, avec la commande suivante, ça marche :

```
> tableau$X1

[1] 11 15  3  8 18 10 13
```

Pour éviter qu'on précède le nom de la série par le nom du « data.frame » et le dollar $, on utilise, pour chaque session de travail, la fonction «attach()» et à la fin de la session, on termine par «detach()» :

```
> attach(tableau)
> X1;X2

[1] 11 15  3  8 18 10 13

[1] M F F F M F M
Levels: F M

> detach(tableau)
```

Remarque :

Lorsque vous importez un tableau de données depuis Excel ou depuis d'autres tableurs vers R, il est automatiquement déclaré « data.frame ». Il y a une autre raison qui nous pousse à étudier les « data.frame », c'est qu'il y a des fonctions de R dont les sorties sont de class « data.frame ».

les listes I

C'est la même chose que data.frame, de plus les données peuvent être de longueurs différentes. La liste peut aussi contenir des sous –listes, des data.frame comme données :

```
> n <- 7
> maliste <- list(tableau,B,n)
> maliste
```

les listes II

```
[[1]]
  X1 X2
1 11  M
2 15  F
3  3  F
4  8  F
5 18  M
6 10  F
7 13  M

[[2]]
[1] 1 8 6 0 1
```

les listes III

```
[[3]]
[1] 7
```

```
> str(maliste)

List of 3
 $ :'data.frame':        7 obs. of  2 variables:
  ..$ X1: num [1:7] 11 15 3 8 18 10 13
  ..$ X2: Factor w/ 2 levels "F","M": 2 1 1 1 2 1 2
 $ : num [1:5] 1 8 6 0 1
 $ : num 7
```

les listes IV

```
> names(maliste)<- c("TAB","B","n")
> maliste

$TAB
  X1 X2
1 11  M
2 15  F
3  3  F
4  8  F
5 18  M
6 10  F
7 13  M

$B
```

les listes V

```
[1] 1 8 6 0 1

$n
[1] 7

> maliste[[2]]

[1] 1 8 6 0 1

> class(maliste)

[1] "list"

> str(maliste)
```

les listes VI

```
List of 3
 $ TAB:'data.frame':        7 obs. of  2 variables:
  ..$ X1: num [1:7] 11 15 3 8 18 10 13
  ..$ X2: Factor w/ 2 levels "F","M": 2 1 1 1 2 1 2
 $ B   : num [1:5] 1 8 6 0 1
 $ n   : num 7
```

Remarque :

Là aussi, lorsqu'on exécute des fonctions de R qui représentent des méthodes avancées en statistique, on obtient des sorties sous forme de list. Il faut donc savoir explorer ces listes pour pouvoir les exploiter.

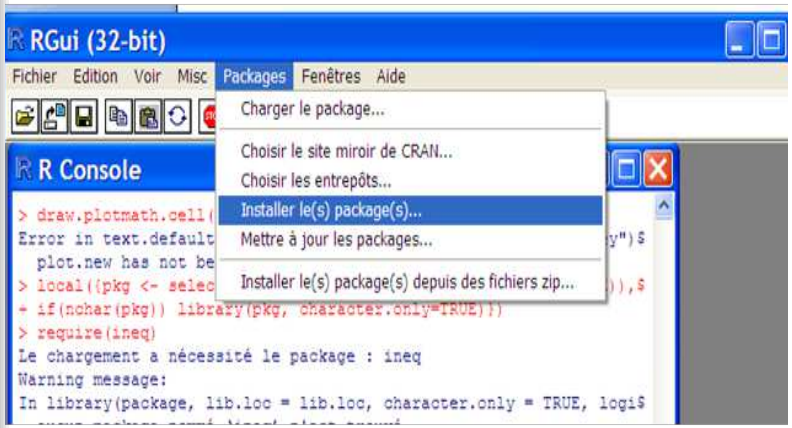
Packages I

Au début de ce chapitre, on a vu que R est un logiciel modulaire puisqu'il fait augmenter ses fonctionnalités à l'aide de bibliothèques externes. Ces packages sont programmés par leurs auteurs pour développer certaines méthodes statistiques ou des applications. Il existe deux types de librairies : celles qui sont indispensables et donc installées par défaut sur R, comme : datasets , base, methods, stat, graphics,. . . , les autres packages qu'il faut télécharger (si besoin est) depuis le CRAN et en suite charger pour la session de travail : LeLogicielR, Rcmdr, FactoMineR, . . . Pour ce faire, et si le package est déjà téléchargé par internet, on dispose de deux commandes qu'on saisit dans la console et qui restent valables pour toute la session de R (par exemple pour charger l'interface graphique Rcommander) :

```
> require(Rcmdr)
```

Packages II

Par contre, si vous ne disposez pas du package, il faut d'abord le télécharger depuis internet. Pour ce faire, vous cliquez sur le menu «packages» puis sur «installer le(s) package(s)», là vous obtenez une liste «miroir CRAN», on peut en choisir un au hasard, par exemple le «France(Paris2)» ce qui vous donne une liste de tous les packages sur le site de CRAN et vous sélectionnez le(s) package(s) que vous désirez...



Packages III

Exemple

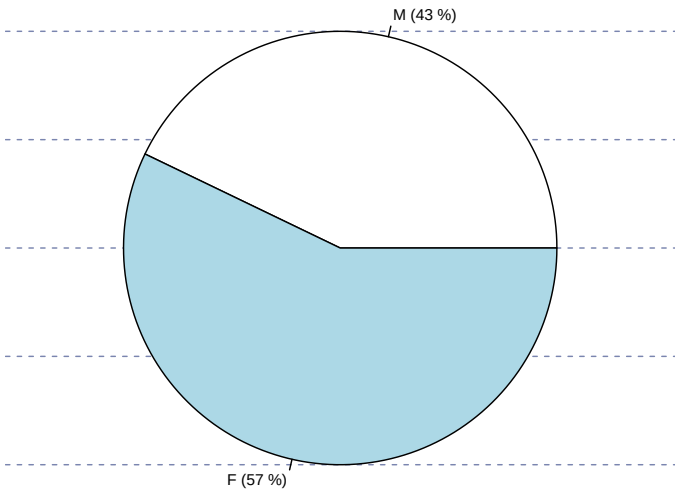
On va se donner une dernière application. Reprenons l'exemple des 7 étudiants selon le critère sexe :

```
> sexe
[1] M F F F M F M
Levels: F M
> camembert(sexe)
```

Le message ≪ Erreur ≫ est normal, car la fonction ≪ camembert() ≫ qui donne un graphique très bien soigné n'existe dans aucun package installé par défaut, mais existe belle et bien dans le package ≪LeLogicielR ≫

Packages IV

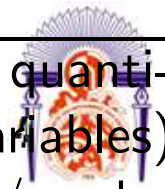
```
> require(LeLogicielR) # la fonction camembert() existe dan
> camembert(sexe)
```



Ça marche maintenant !

Packages I

Fonction	Description
c()	Pour créer une série brute ou simplement un vecteur contenant un nombre (≥ 2) de données.
seq()	Une séquence de nombres avec un pas fixe ou une longueur donnée
cbind()	Coller deux ou plusieurs séries brutes (en colonnes) pour en faire un tableau de plusieurs caractères.
srbind()	Même chose, mais avec des lignes. Par exemple, lorsqu'on veut ajouter un ou plusieurs individus avec leurs valeurs prises pour tous les caractères du tableau.
t()	Transpose un tableau ou une matrice, exemple(transforme une ligne en colonne ou vis-versa).
data.frame()	Appliquée à un ou plusieurs caractères qualitatifs et/ou quantitatifs pour en faire un tableau statistique (Individus×variables) pour une population donnée avec noms des individus et/ou des



Packages II

list()	Tableau constitué de variables de natures et/ou tailles $\neq$.
sort()	Ordonne une série brute par ordre croissant.
unique()	Pour obtenir les modalités d'une série brute mais sans ordre.
table()	Très importante dans le cadre de ce cours. A partir d'une série brute, on construit une distribution statistique en termes d'effectifs, où les modalités sont ordonnées (si le caractère est qualitatif, les modalités sont classées par ordre alphabétique). Aussi, elle nous permet, d'obtenir à partir de deux séries brutes, un tableau de contingence.
ordered()	Permet de classer, par ordre les modalités d'un caractère qualitatif ordinal.
cut()	Transforme une série brute quantitative en variable statistique continue, en effectuant un découpage automatique ou en vous laissant choisir les bornes des classes et par conséquence le nombre

Packages III

sqrt()	Racine carrée : $\sqrt{x}$.
mean()	Moyenne arithmétique d'une série brute quantitative.
var()	variance arithmétique d'une série brute quantitative qu'on doit multiplier par (n-1)/n pour avoir la définition qu'on a adopté ici.
sd()	Ecart-type qu'on doit multiplier par $\sqrt{\frac{n-1}{n}}$ pour avoir la définition qu'on a adopté ici.
median()	La médiane d'une série brute quantitative ou qualitative ordinale.
quantile()	Les quantiles d'une série brute quantitative ou qualitative ordinale.
diff(range())	L'étendue d'une série brute.
round()	Arrondit le résultat à la décimale souhaitée.

Packages IV

summary()	Importante : fournit un résumé statistique d'un ou plusieurs ca ractères de mêmes natures ou de natures différentes.
str()	Donne la structure d'un objet de R, c'est une alternative de sum mary().
lm()	Effectue une régression linéaire et offre un objet list() plein d résultats, qu'on peut explorer facilement grâce à la fonction str(
cor()	Coefficient de corrélation linéaire entre deux variables statistiques
ts()	déclare une série brute comme une série chronologique.
decompose()	Donne toutes les composantes d'une série chronologique d période ≥ 2.
predict()	Donne les prévisions d'une série chronologique.
plot()	Produit un graphique. Selon son argument «type», vous aurez un nuage de points ou une courbe ou une courbe avec un « mar quage » des points,. . . , tapez « ?plot».

Packages V

x11()	Vous permet d'avoir une fenêtre graphique vierge sans écraser la précédente
abline()	Permet de tracer une droite sur un graphique déjà existant, tel que par exemple la droite de régression sur un nuage de points issu de la fonction plot() ; exemple : abline(lm(y x)) après plot(x,y). A ce titre, on peut citer plusieurs autres fonctions de même type qui ne peuvent fonctionner que sur des graphiques existants : segments(), points(), lines(),. . .
expression()	Permet d'écrire une expression comprenant des formules mathématiques, mais qui ne fonctionne qu'avec des fonctions telle que : (text, mtext, axis, legend). Pour amples informations tapez « ?plotmath » ou « ?expression ».
pie()	Fournit un diagramme circulaire lorsque x est une distribution qualitative.
camembert()	Fournit un beau diagramme circulaire lorsque x est une série brute qualitative mais nécessite le package « LeLogicielR ».

Packages VI

barplot()	Fournit un tuyaux d'orgue lorsque x est une distribution qualitative.
Tuyauxorgue()	Fournit un beau tuyaux d'orgue lorsque x est une série brute qualitative mais nécessite le package ≪ LeLogicielR ≫.
hist()	Fournit un histogramme lorsque x représente les bornes des classes avec leurs pondérations.
locator()	Fournit les coordonnées, sur un graphique déjà existant, du point sur le quel vous cliquez par le bouton gauche de la souris, par exemple locator(3) vous oblige à cliquer sur 3 endroits du graphique, donc il faut 3 cliques. Il vous permet aussi de placer des objets (textes, points, segments, symboles,. . .) dans les endroits que vous désirez du graphique.
attach()	Permet d'avoir accès directement aux variables d'une data.frame seulement en écrivant leurs noms ; sinon il faut écrire le nom de la dataframe suivi du dollar ($) et du nom de la variable ; exemple data$X

Packages VII

Ce tableau donne un résumé des principales fonctions de R utilisées dans le cadre de la statistique descriptive. Pour plus amples informations sur une fonction, on peut l'écrire sur la console (sans parenthèses) précédée d'un point d'interrogation. Exemple : ?c