

Introduction à R

Manipulation des données

Ricco Rakotomalala

http://eric.univ-lyon2.fr/~ricco/cours/cours_programmation_R.html

(1) R est un langage de programmation. L'objet de base est un vecteur de données.

C'est un « vrai » langage c.-à-d. types de données, branchements conditionnels, boucles, organisation du code en procédures et fonctions, découpage en modules.

Mode de d'exécution : transmettre à R le fichier script « .r »

(2) R est un logiciel de statistique et de data mining, pilotée en ligne de commande. Il est extensible (quasiment) à l'infini via le système des packages.

Les instructions servent à manipuler les objets R c.-à-d. les ensembles de données, les vecteurs, les modèles, etc.

Mode de d'exécution : introduire commandes dans le terminal, manipulation interactive

→ C'est le mode que nous exploiterons dans ce tutoriel.

<http://www.r-project.org/>

The R Project for Statistical Computing

PCA 5 vars
princcomp(x = data, cor = cor)

Clustering 4 groups

Factor 1 [41%]

Factor 3 [19%]

Getting Started:

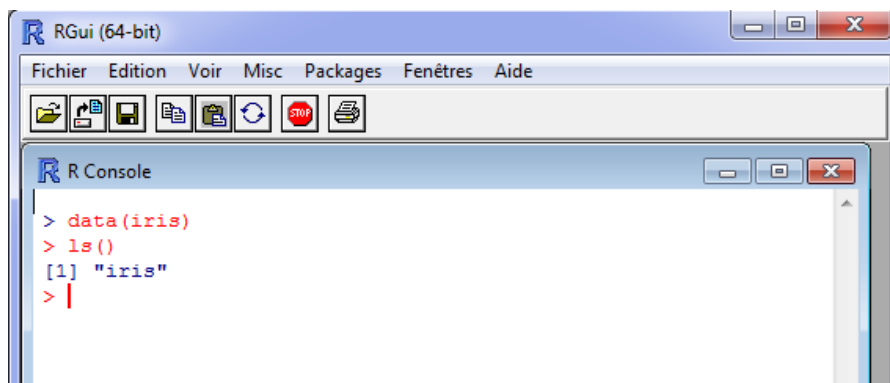
- R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To **download R**, please visit your preferred [CRAN mirror](#).
- If you have questions about R like how to download and install software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

News:

- R version 2.14.1 (December Snowflakes) has been released on 2011-12-22.
- [The R Journal Vol.3/2](#) is available.
- [useR! 2012](#), will take place at Vanderbilt University, Nashville Tennessee, USA, June 12-15, 2012.
- R version 2.13.2 has been released on 2011-09-30.

This server is hosted by the [Institute for Statistics and Mathematics](#) of the [WU Wien](#).

R peut fonctionner sous **Windows**, **Mac OS X**, **Linux**



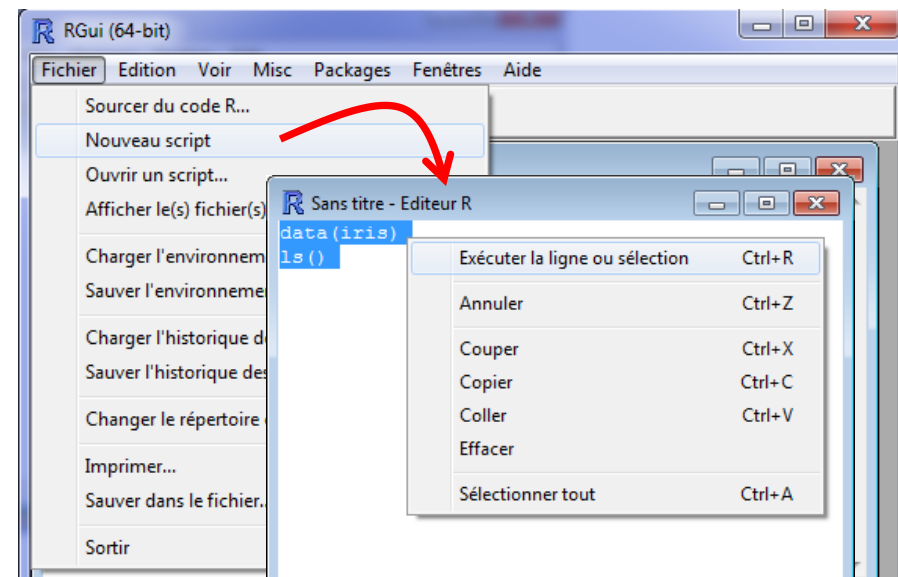
```
> data(iris)
> ls()
[1] "iris"
> |
```

Mode « terminal »

+ interactivité, visualisation immédiate des résultats

+ avec « ↑ », on retrouve les anciennes commandes

- pas de sauvegarde des commandes si fermeture de R (possible en fait, avec *FICHIER / SAUVER L'HISTORIQUE DES COMMANDES*)



Mode « script »

+ interactivité, visualisation immédiate des résultats (CTRL + R)

+ maintien d'une liste « propre » des commandes utiles uniquement

+ possibilité d'E/S (chargement ou sauvegarde d'un fichier script « .r »)

→ mode conseillé pour nous

Si on veut programmer (mode programmation), mieux vaut passer par un éditeur externe (ex. TINN-R, R-STUDIO, ECLISPE + StatET,...)



Charger les données – Structure data.frame

demo_reglog.txt - Bloc-notes

Fichier	Edition	Format	Affichage ?
age	taux	angine	coeur
50	126.0	1	presence
49	126.0	0	presence
46	144.0	0	presence
49	139.0	0	presence
62	154.0	1	presence
35	156.0	1	presence
67	160.0	0	absence
65	140.0	0	absence
47	143.0	0	absence
58	165.0	0	absence
57	115.0	1	absence
59	145.0	0	absence
44	175.0	0	absence
41	153.0	0	absence
54	152.0	0	absence
52	169.0	0	absence
57	168.0	1	absence
50	158.0	0	absence
44	170.0	0	absence
49	171.0	0	absence

Nom du data.frame

Séparateur de colonnes

1^{ère} ligne = nom des variables

Nom du fichier

Point décimal

```
R Console
> heart <- read.table(file="demo_reglog.txt", sep="\t", dec=".", header=T)
> class(heart)
[1] "data.frame"
> summary(heart)
      age      taux      angine      coeur
Min.   :35.00  Min.   :115.0  Min.   :0.00  absence :14
1st Qu.:46.75  1st Qu.:142.2  1st Qu.:0.00  presence: 6
Median :50.00  Median :153.5  Median :0.00
Mean   :51.75  Mean   :151.4  Mean   :0.25
3rd Qu.:57.25  3rd Qu.:165.8  3rd Qu.:0.25
Max.   :67.00  Max.   :175.0  Max.   :1.00
> |
```

« âge », « taux » et « angine » sont considérées comme quantitatives.

« cœur » est une variable qualitative.

Fichier texte, séparateur tabulation.

data.frame = matrice de données = liste de vecteurs de même longueur.

Vecteur = variable.

Les variables sont typées. Les plus utilisées sont « numeric / integer » (variables quantitatives) et « factor » (variables qualitatives)

Remarque : on peut accéder aux variables d'un data.frame avec l'opérateur \$

```
R Console
> class(heart$age)
[1] "integer"
> class(heart$taux)
[1] "numeric"
> class(heart$angine)
[1] "integer"
> class(heart$coeur)
[1] "factor"
> mean(heart$age)
[1] 51.75
> |
```

Package ?

- Un package est une bibliothèque externe
- Sous Windows → fichiers binaires pré-compilés
- Extension .zip
- Il est toujours documenté : fichier HTML (aide sous R) et PDF

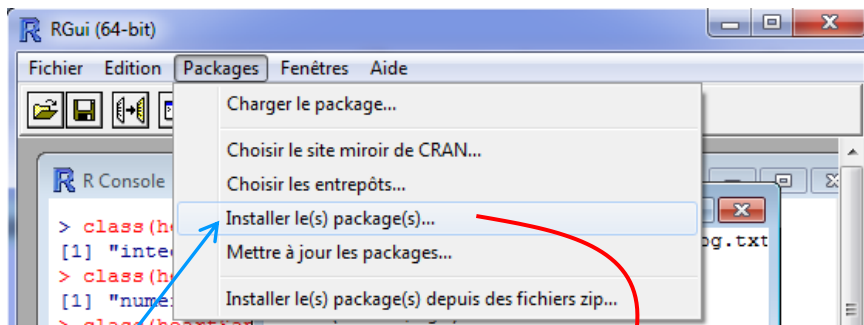
Quel intérêt ?

- Un package contient des collections de fonctions utilisables sous R
- Souvent centrés sur un sujet particulier (ex. *rpart* pour les arbres de décision, etc.)
- Gestion affinée des packages : nous pouvons les installer, désinstaller, charger, décharger et mettre à jour à notre guise

Ce système permet d'augmenter considérablement la puissance de R !!!

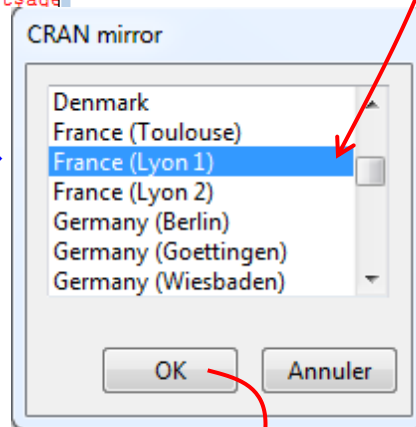
Ex. installer et charger le package « xlsx » permettant de lire directement les fichiers Excel (*.xls et *.xlsx)

Installation (une fois) et chargement d'un package (à chaque utilisation)

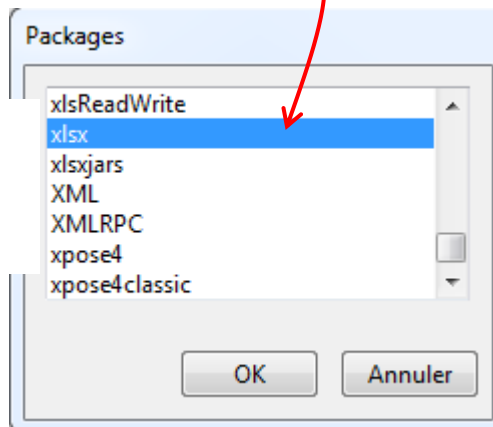


(1) Lancer l'installation

(2) Choisir le site miroir de téléchargement →



(3) Choisir le package à installer →



Après installation...

> **library(nom du package)** permet de charger la bibliothèque
> **library(help=nom de package)** permet de lister ses fonctions

Ex.

#charger la librairie xlsx

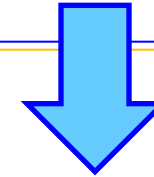
library(xlsx)

#lister les fonctions de la librairie

library(help=xlsx)

Comprendre la structure d'un data frame (ensemble de données) [1/2]

- > Changement du répertoire courant [`setwd`]
- > Chargement du package [`library`]
- > Lecture du fichier Excel (1^{ère} feuille, nom de variable sur 1^{ère} ligne) [`read.xlsx`]
- > Statistiques descriptives sur l'ensemble des variables [`summary`]
- > `ls()` liste le contenu de la mémoire
- > Remarquer le rôle de l'opérateur d'affectation `<-`



```
> #vider le contenu de la mémoire
> rm(list=ls())
> #charger la librairie xlsx
> library(xlsx)
> #lister les fonctions de la librairie
> #library(help=xlsx)
> #changement du répertoire courant
> setwd("C:/_Travaux/university/Cours_Universite/Supports_de_cours/Informatique/R/Slides/fichiers exemples")
> #charger les données
> heart.full <- read.xlsx(file="heart.xlsx",sheetIndex=1,header=T)
> #description des variables de l'ensemble de données
> print(summary(heart.full))
```

age	sexe	typedouleur	sucré	tauxmax
Min. :29.00	feminin : 87	A: 20	A:230	Min. : 71.0
1st Qu.:48.00	masculin:183	B: 42	B: 40	1st Qu.:133.0
Median :55.00		C: 79		Median :153.5
Mean :54.43		D:129		Mean :149.7
3rd Qu.:61.00				3rd Qu.:166.0
Max. :77.00				Max. :202.0


angine	depression	coeur
non:181	Min. : 0.0	absence :150
oui: 89	1st Qu.: 0.0	presence:120
	Median : 8.0	
	Mean :10.5	
	3rd Qu.:16.0	
	Max. :62.0	

```
> #lister le contenu de la mémoire
> ls()
[1] "heart.full"
```

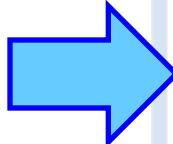

Data.frame = collection de variables

Accès aux variables (colonnes) avec \$

heart.full



\$age	\$sexe	\$typedouleur	\$sucre	\$tauxmax	\$angine	\$depression	\$coeur
70	masculin	D	A	109	non	24	presence
67	feminin	C	A	160	non	16	absence
57	masculin	B	A	141	non	3	presence
64	masculin	D	A	105	oui	2	absence
74	feminin	B	A	121	oui	2	absence
65	masculin	D	A	140	non	4	absence
56	masculin						
59	masculin						
60	masculin						
63	feminin						
59	masculin						



```
R Console
> print(heart.full$age)
 [1] 70 67 57 64 74 65 56 59 60 63 59 53 44 61 57 71 46 53 64 40
 [21] 67 48 43 47 54 48 46 51 58 71 57 66 37 59 50 48 61 59 42 48
 [41] 40 62 44 46 59 58 49 44 66 65 42 52 65 63 45 41 61 60 59 62
 [61] 57 51 44 60 63 57 51 58 44 47 61 57 70 76 67 45 45 39 42 56
 [81] 58 35 58 41 57 42 62 59 41 50 59 61 54 54 52 47 66 58 64 50
 [101] 44 67 49 57 63 48 51 60 59 45 55 41 60 54 42 49 46 56 66 56
 [121] 49 54 57 65 54 54 62 52 52 60 63 66 42 64 54 46 67 56 34 57
 [141] 64 59 50 51 54 53 52 40 58 41 41 50 54 64 51 46 55 45 56 66
 [161] 38 62 55 58 43 64 50 53 45 65 69 69 67 68 34 62 51 46 67 50
 [181] 42 56 41 42 53 43 56 52 62 70 54 70 54 35 48 55 58 54 69 77
 [201] 68 58 60 51 55 52 60 58 64 37 59 51 43 58 29 41 63 51 54 44
 [221] 54 65 57 63 35 41 62 43 58 52 61 39 45 52 62 62 53 43 47 52
 [241] 68 39 53 62 51 60 65 65 60 60 54 44 44 51 59 71 61 55 64 43
 [261] 58 60 58 49 48 52 44 56 57 67
> |
```

```

> class(heart.full$age)
[1] "numeric"
> #'longueur' de 'age' -> nombre d'observations
> length(heart.full$age)
[1] 270
> #accès par indices - plage de valeurs
> heart.full$age[1:10]
[1] 70 67 57 64 74 65 56 59 60 63
> #accès par indices - qqs valeurs dispersées
> heart.full$age[c(2,5,8)]
[1] 67 74 59
> #stat.descriptives - moyenne
> mean(heart.full$age)
[1] 54.43333
> #stat.descriptives - quantiles
> quantile(heart.full$age,probs=c(0.1,0.5,0.9))
10% 50% 90%
 42  55  66
> #stat.descriptives sur une partie des valeurs
> mean(heart.full$age[1:10])
[1] 63.5

```

class()

length()

x[a:b]

x[c(.,.,.,.)]

mean()

quantile()

```

> #type de 'sexe'
> class(heart.full$sexe)
[1] "factor"
> #description des valeurs
> levels(heart.full$sexe)
[1] "feminin" "masculin"
> #nombre d'observations de 'sexe'
> length(heart.full$sexe)
[1] 270
> #accès par indices - plage de valeurs
> heart.full$sexe[1:10]
[1] masculin feminin masculin masculin feminin masculin masculin m
[9] masculin feminin
Levels: feminin masculin
> #accès par indices - qqs valeurs dispersées
> heart.full$sexe[c(2,5,8)]
[1] feminin feminin masculin
Levels: feminin masculin
> #fréquences
> table(heart.full$sexe)

feminin masculin
      87      183
> #codes internes des valeurs
> unclass(heart.full$sexe)
 [1] 2 1 2 2 1 2 2 2 2 1 2 2 2 2 1 1 2 2 2 2 2 2 2 2 1 1 1 1 2 1 2 2
[38] 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 1 1 1 1 2 1 1 2 2 1 2 1 2 2 2 1 1
[75] 1 2 2 1 1 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 1 1 1 2 1 2 2 2
[112] 2 1 1 2 1 2 1 1 2 2 2 2 1 2 1 2 1 2 2 1 2 2 2 2 1 1 2 1 2 2 2 2
[149] 2 1 2 1 1 1 1 1 2 2 2 1 2 1 2 2 2 1 1 2 1 2 2 2 2 1 2 1 2 2 2 2
[186] 2 2 2 1 2 2 2 2 2 2 1 1 1 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 1 2 2 1
[223] 2 2 1 2 1 1 1 2 1 2 2 2 2 1 1 2 2 2 2 2 2 1 1 1 2 2 1 2 2 2 2 2
[260] 2 1 2 2 2 2 2 2 1 2 2
attr(,"levels")
[1] "feminin" "masculin"

```

- class()
- levels()
- length()
- x[a:b]
- x[c(.,.,.,.)]
- mode()
- table()
- unclass()

```

> m <- heart.full
> nrow(m)
[1] 270
> ncol(m)
[1] 8
> m[1,1]
[1] 70
> m[1:5,2:4]
  sexe typedouleur sucre
1 masculin         D    A
2 féminin         C    A
3 masculin         B    A
4 masculin         D    A
5 féminin         B    A
> m[c(2,5,8),2:4]
  sexe typedouleur sucre
2 féminin         C    A
5 féminin         B    A
8 masculin         D    A
> m[2:4,c(1,3,6)]
  age typedouleur angine
2  67           C   non
3  57           B   non
4  64           D   oui
> m[1,]
  age      sexe typedouleur sucre tauxmax angine depression      coeur
1  70 masculin           D    A    109   non           24 presence
> m[5:6,c("age","angine")]
  age angine
5  74   oui
6  65  non

```

Points importants

- > Affectation à une variable **m** pour faciliter l'édition (pas nécessaire)
- > **nrow()** et **ncol()**
- > Accès indicé [ligne et colonne]
- > Utilisation de plages d'indices
- > Indices non contigus
- > Absence d'indice = toutes les lignes ou colonnes

```

> m[m$age<30,]
  age  sexe typedouleur sucre tauxmax angine depression  coeur
215  29 masculin      B   A    202   non           0 absence
> m[m$age<=34 & m$sexe=="masculin",c("age","sexe","coeur")]
  age  sexe  coeur
175  34 masculin absence
215  29 masculin absence
> m[m$age<=34 | m$age>=76,c("age","sexe")]
  age  sexe
74   76  feminin
139  34  feminin
175  34 masculin
200  77 masculin
215  29 masculin
> a <- m[m$age<=45 & m$sexe=="masculin",c("angine","coeur")]
> nrow(a)
[1] 39
> ncol(a)
[1] 2
> table(a$coeur)

absence presence
      27      12
> k <- table(a$angine,a$coeur)
> class(k)
[1] "table"
> print(k)

      absence presence
non         24        5
oui         3         7
> print(k[1,2])
[1] 5

```

Points importants

- > Intégrer des conditions dans les restrictions
- > Conditions complexes avec ET (&) et OU (|)
- > Le résultat (**a**) est aussi de type data frame, **nrow()** et **ncol()** sont opérationnels
- > **table()** effectue des comptages, avec tris à plat ou croisés
- > Le résultat (**k**) est de type table (≈ matrice), accès indexé possible

```

> #croisement coeur-angine
> e <- table(m$angine,m$coeur)
> print(e)

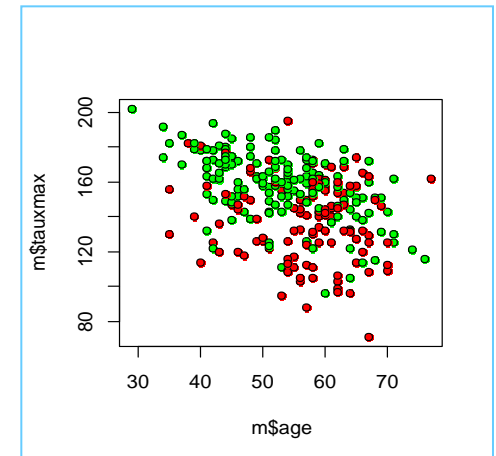
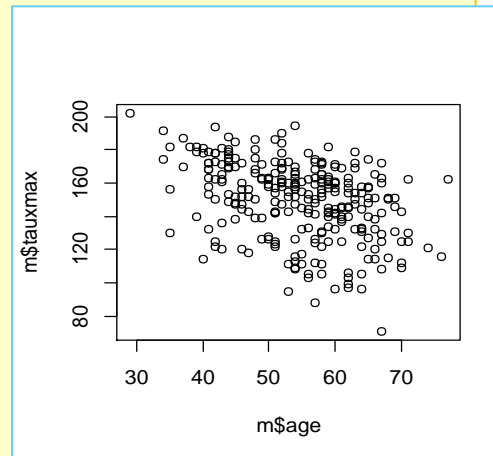
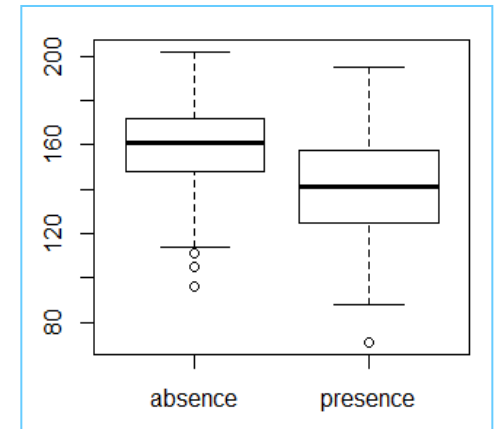
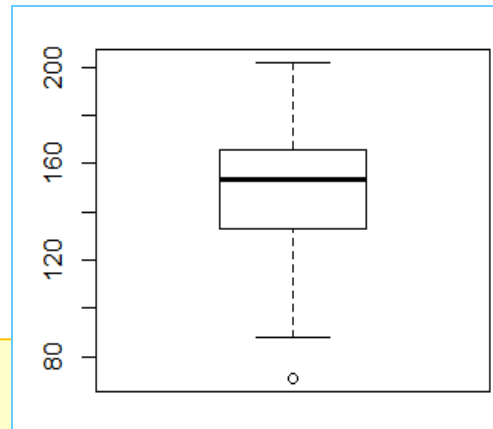
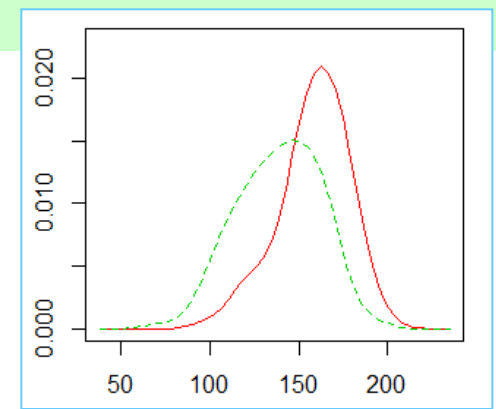
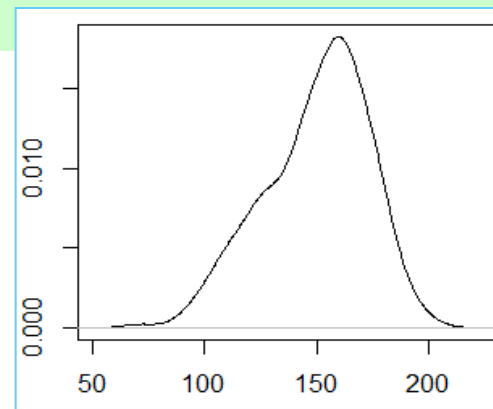
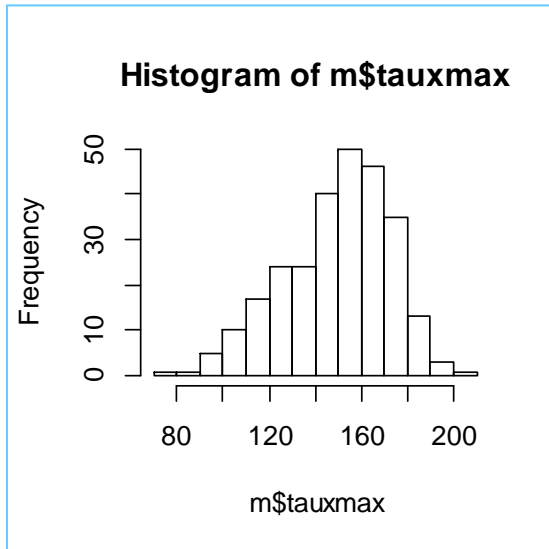
      absence presence
non      127      54
oui       23      66
> #nombre de angine=oui
> print(sum(e[2,]))
[1] 89
> #proportion de malades parmi les angine=oui
> print(e[2,2]/sum(e[2,]))
[1] 0.741573
> #age moyen selon le sexe
> tapply(X=m$age, INDEX=m$sexe, mean)
  feminin masculin
55.67816 53.84153
> #age moyen selon le sexe et l'occurrence de l'angine
> b <-tapply(X=m$age, INDEX=list(m$sexe,m$angine), mean)
> print(b)

      non      oui
feminin 55.72464 55.50000
masculin 52.62500 55.76056
> #écart entre la plus petite et la plus grande moyenne
> d <- max(b)-min(b)
> print(d)
[1] 3.135563
> #écart entre min et max dans chaque sous-groupe
> tapply(X=m$age, INDEX=m$sexe, fonction(x) {max(x)-min(x)})
  feminin masculin
      42      48
> ls()
[1] "a"      "b"      "d"      "e"      "heart.full"
[6] "k"      "m"

```

Points importants

- > Exploitation de l'objet **table**
- > L'outil **tapply()** pour le calcul conditionnel
- > **tapply** renvoie un objet table, exploitable également
- > Utilisation d'une **fonction personnalisée** dans tapply
- > Listage du contenu de la mémoire avec **ls()**



```
#graphiques 1D - distributions  
hist(m$tauxmax)  
plot(density(m$tauxmax))  
#graphiques pour comparaisons  
library(sm)  
sm.density.compare(m$tauxmax,m$coeur)  
#comparaisons avec boxplot  
boxplot(m$tauxmax)  
boxplot(m$tauxmax ~ m$coeur)  
#graphiques 2D  
plot(m$age,m$tauxmax)  
plot(m$age,m$tauxmax,pch=21,bg=c("green","red")[unclass(m$coeur)])
```

```

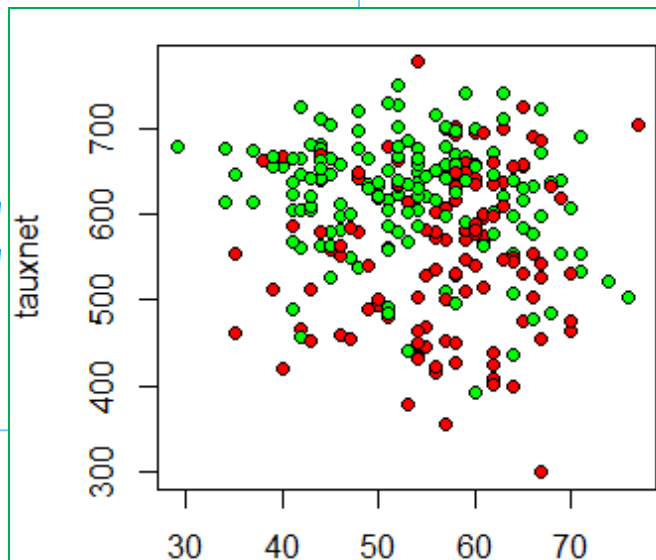
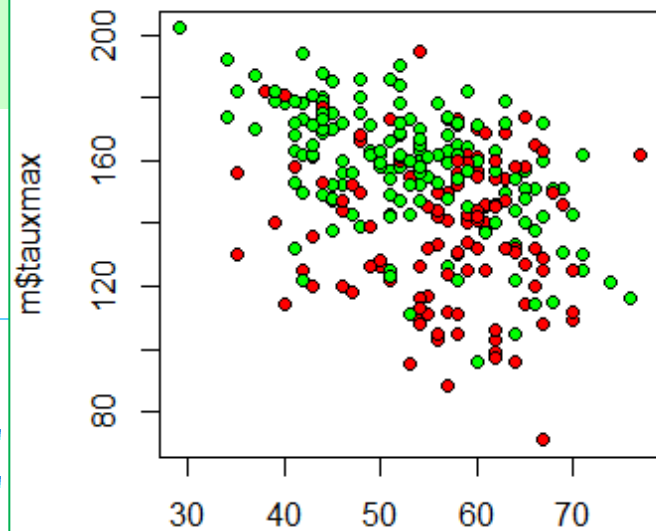
> #afficher les 6 premières valeurs de age
> head(m$age)
[1] 70 67 57 64 74 65
> #age2 est un nouveau vecteur des valeurs triées
> age2 <- sort(m$age)
> #afficher les 6 premières valeurs de age2
> head(age2)
[1] 29 34 34 35 35 35
> #trier un data frame selon une ou plusieurs variables
> head(order(m$age)) #création d'un index selon l'âge
[1] 215 139 175 82 194 225
> head(m[order(m$age),]) #tri selon un critère
  age     sexe typedouleur sucre tauxmax angine depression  coeur
215  29 masculin           B     A     202   non           0 absence
139  34  féminin           B     A     192   non           7 absence
175  34 masculin           A     A     174   non           0 absence
82   35 masculin           D     A     130  oui            16 presence
194  35 masculin           D     A     156  oui            0 presence
225  35  féminin           D     A     182   non           14 absence
> head(m[order(m$age,m$tauxmax),]) #tri selon deux critères
  age     sexe typedouleur sucre tauxmax angine depression  coeur
215  29 masculin           B     A     202   non           0 absence
175  34 masculin           A     A     174   non           0 absence
139  34  féminin           B     A     192   non           7 absence
82   35 masculin           D     A     130  oui            16 presence
194  35 masculin           D     A     156  oui            0 presence
225  35  féminin           D     A     182   non           14 absence

```


Création de nouvelles variables

Insertion dans un data frame existant

```
> #lister les variables du data frame
> colnames(m)
[1] "age"          "sexe"          "typedouleur"  "sucre"
[5] "tauxmax"      "angine"        "depression"    "coeur"
> #graphique nuage de points
> plot(m$age,m$tauxmax,pch=21,bg=c("green","red")[unclass(m$coeur)])
> #création de la variable tauxnet
> tauxnet <- m$taux*log(m$age)
> #graphique nuage de points
> plot(m$age,tauxnet,pch=21,bg=c("green","red")[unclass(m$coeur)])
> #ajouter la nouvelle variable au data frame "m"
> m <- cbind(m,tauxnet)
> #lister les variables
> colnames(m)
[1] "age"          "sexe"          "typedouleur"  "sucre"
[5] "tauxmax"      "angine"        "depression"    "coeur"
[9] "tauxnet"
> #sauvegarde des données dans un fichier XLSX
> write.xlsx(m,file="heart-output.xlsx",row.names=F)
```



De la documentation à profusion sur internet (n'achetez jamais des livres sur R)

Site du cours

http://eric.univ-lyon2.fr/~ricco/cours/cours_programmation_R.html

Aide mémoire

<http://www.duclert.org/>

Quick-R

<http://www.statmethods.net/>

POLLS (Kdnuggets)

Data Mining / Analytics Tools Used - <http://www.kdnuggets.com/polls/2011/tools-analytics-data-mining.html>

(Mai 2012, R en 1^{ère} position; 2^{nde} en 2011 et 2010)

What languages you used for data mining / data analysis?

<http://www.kdnuggets.com/polls/2011/languages-for-data-mining-analytics.html>

(Août 2012, langage R en 1^{ère} position ; idem en 2011)

Article New York Times (Janvier 2009)

“Data Analysts Captivated by R’s Power” - http://www.nytimes.com/2009/01/07/technology/business-computing/07program.html?_r=1