

Formation Logiciel R



Institut Pasteur
de Nouvelle-Calédonie

1

1^o séance

Découverte et premiers pas
avec le logiciel R

2

Qu'est ce que



- R : un logiciel et un langage
- R : un logiciel libre
- R : un langage orienté objet
- R : un système statistique et graphique

3

Mode de fonctionnement

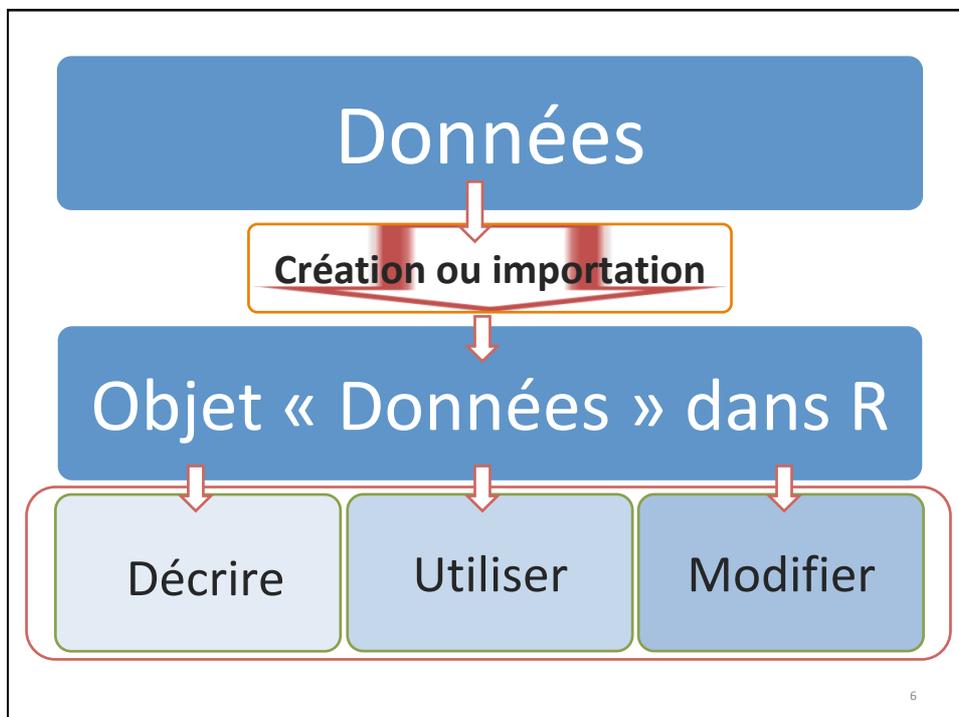
- On utilise généralement R interactivement, selon un cycle question/réponse :
- Vous entrez une commande et tapez la touche "Retour à la ligne".
- R exécute cette commande
- R attend une autre commande

4

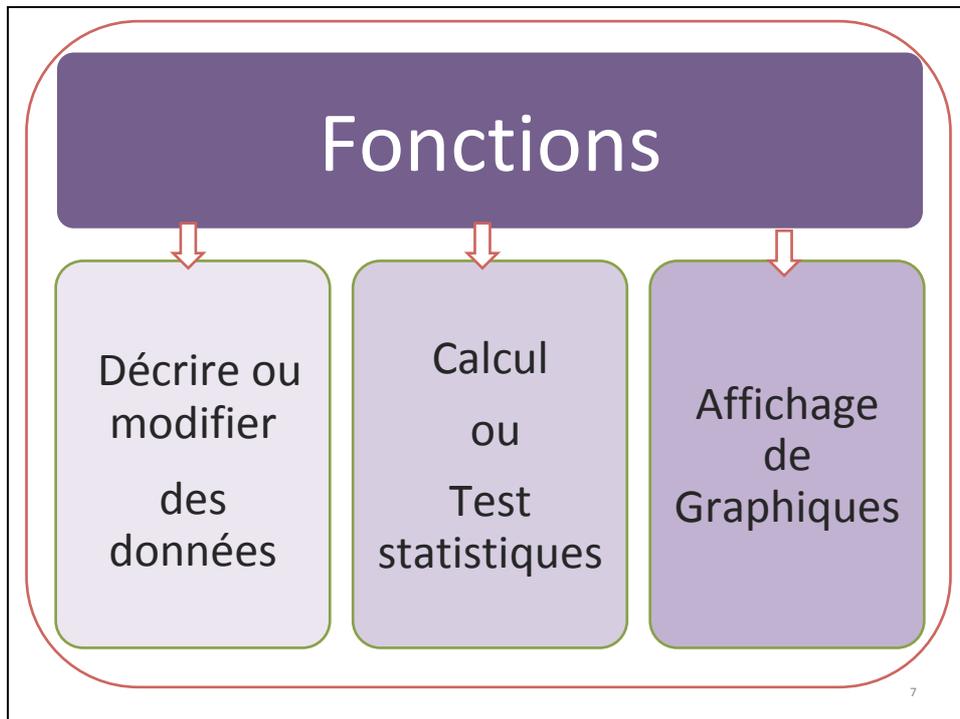
Langage orienté « objet »

- Fonctionne avec des objets
 - Données : vecteur, matrice, liste, table
 - Fonction: `sum () table()`
- Manipulation des objets
 - Instructions
 - > prompt – débiter
 - # pour décrire les instructions
 - + indique que R attend la suite de l'instruction

5



6



Consignes pour les objets « Fonction »

- Le nom des objets «Fonction » est écrit généralement en minuscule et est suivi d'une parenthèse : `sum()`
- Pour connaître le contenu d'une fonction
→ Taper son nom sans parenthèse ex: `sum`
- **Arguments** : paramètres à l'intérieur des parenthèses
Ex: `sum (1,4)`

NB: La virgule est le séparateur des arguments

→ pour les nombres décimaux on utilisera donc le point (écriture anglo-saxonne)

8

Consignes pour les objets « Données »

- Le nom des Objets « Données »:
 - Alphanumérique: A b 3
 - Débuter par une lettre
 - Seul le point « . » est accepté dans le nom de l'objet
 - R différencie minuscule et majuscule
 - Nom ≠ nom ≠ NOM
 - Mydata.1 ≠ Mydata1

9

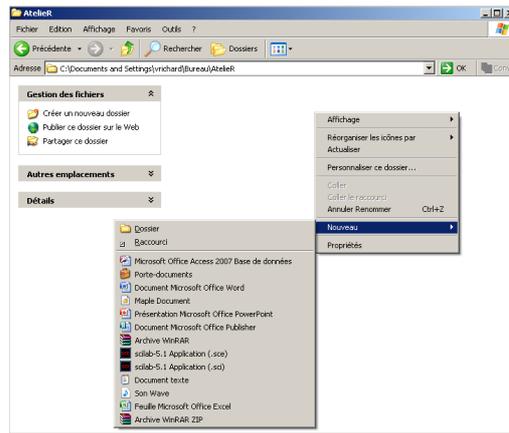
Conseils pour les objets «Données »

- Pour différencier dans un programme les objets « Données » des objets «Fonction » qui sont systématiquement écrits en minuscule (ex: `sum()`), il est préférable de nommer ces objets « données » en commençant avec une première lettre en Majuscule (ex: `Mydata`)

10

Pour commencer cette séance de prise en main

- Créer un répertoire appelé « Atelier » sur le bureau et y créer un fichier quelconque
(clic droit sur la souris, puis nouveau fichier choisir par exemple document Texte et le nommer « temp »)



11

Prise en main du logiciel

- Installer R à partir du site internet

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

- Lancer R
- Le logiciel s'ouvre sur la fenêtre de travail intitulée:



Rconsole

12

Démarrer R - Console

```
R version 2.15.0 (2012-03-30)
Copyright (C) 2012 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: i386-apple-darwin9.8.0/i386 (32-bit)
```

```
R est un logiciel libre livré sans AUCUNE GARANTIE.
Vous pouvez le redistribuer sous certaines conditions.
Tapez 'license()' ou 'licence()' pour plus de détails.
```

```
R est un projet collaboratif avec de nombreux contributeurs.
Tapez 'contributors()' pour plus d'information et
'citation()' pour la façon de le citer dans les publications.
```

```
Tapez 'demo()' pour des démonstrations, 'help()' pour l'aide
en ligne ou 'help.start()' pour obtenir l'aide au format HTML.
Tapez 'q()' pour quitter R.
```

```
>
```

13

Installer une extension

- R utilise un système de bibliothèques, les packages.
- Chaque bibliothèque est une collection regroupant des outils d'une même thématique.
- R est lui même une bibliothèque appelée "base"
- Certaines bibliothèques sont automatiquement disponibles lorsque R est lancé, d'autres doivent être chargées avec la fonction `library()`

14

installed.packages()

- Certaines bibliothèques sont pré-installées avec R . La liste des bibliothèques installées est donnée par la fonction :
`installed.packages()`
- Il y a de nombreuses autres bibliothèques développées par des utilisateurs de R disponibles sur le site du CRAN (Comprehensive R Archive Network).

15

search()

Certaines bibliothèques sont automatiquement disponibles lorsque R est lancé. A n'importe quel moment, la liste des bibliothèques chargées est donnée par la fonction `search()` :

`search()`

16

install.packages()

- De nouvelles bibliothèques peuvent être téléchargées et installées
- avec la fonction `install.packages()`. Par exemple, pour installer
- la bibliothèque `epiR` (si elle n'est pas déjà installée), on peut utiliser :

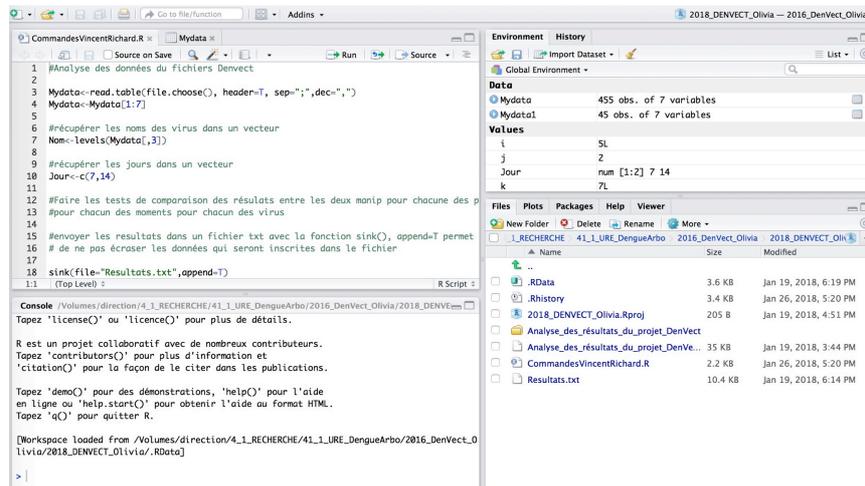
```
install.packages("epiR")  
library(epiR)
```
- La dernière commande donne la liste de toutes les pages d'aide de la bibliothèque.

17

Les éditeurs pour R

18

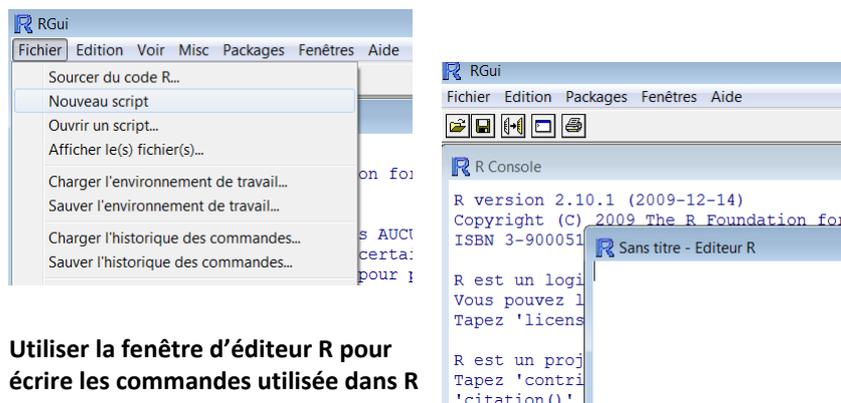
R STUDIO



19

Utiliser l'éditeur de R (1)

- Dans le menu fichier , choisir nouveau script

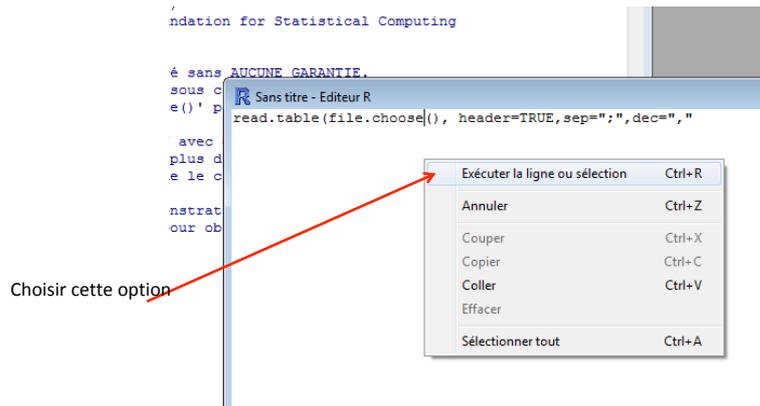


Utiliser la fenêtre d'éditeur R pour écrire les commandes utilisées dans R

20

Utiliser l'éditeur de R (2)

- Transférer les commandes → clic droit de la souris



21

Ouvrir un script

- Sous Mac
 - Menu fichier > nouveau document ou :

Choisir cette option



22

Répertoire de travail (1)

- Par défaut, R lit et écrit dans le répertoire de travail
- Connaître le répertoire de travail de R :

getwd()

```
[1] "C:/R/xxxxxxx"
```

Le répertoire par défaut est celui du logiciel

- **Changer le répertoire de travail de R :**

Utiliser le menu :

fichier-> 'Changer le répertoire courant'

23

Répertoire de travail (2)

- Pour définir le répertoire de travail : fonction **setwd()**
- Le chemin du répertoire de travail peut être fastidieux à écrire

```
setwd("C:/User/nomutilisateur/Documents/Atelier")
```

- **Première solution :**

- `setwd(dirname(file.choose()))`

- `nom<-file.choose()` # permet de choisir un fichier de façon interactive
- `dirname(nom)` # extrait le chemin d'un fichier

- **Deuxième solution :**

- `setwd("~/")` # le répertoire perso de l'utilisateur est : ~
- `getwd()#C:/user/nomutilisateur/Documents`

- `setwd("~/Atelier")` # erreur si ce répertoire n'existe pas

24

Fonctions utiles

- Fonction d'aide
 - **help()** ou **?**
 - Exemple : `help(sum)` ou `?sum`
- Pour citer R dans une publication :
 - **citation()**
- Pour lister les dernières commandes tapées
 - **history()**

25

history()

- Donne les dernières commandes utilisées au cours de la session de travail
- Pour sauvegarder les commandes dans un script il faut faire un copier/coller au fur et à mesure dans le bloc note
- Enregistrer le fichier texte avec l'extension ".R«
- **>history(Inf)** #permet d'avoir toutes les instructions saisies

26

help.start()

R a beaucoup trop d'outils pour que quiconque puisse les retenir tous, il est donc très important de savoir comment retrouver les informations pertinentes en utilisant le système d'aide.

help.start()

ouvre une fenêtre avec une interface pour l'aide de type HTML.

27

RsiteSearch()

La fonction **RSiteSearch()** fait une recherche dans l'ensemble des documents (manuels, documentation, archives des listes de diffusion) du site de R

28

help.search()

Quand vous voulez obtenir de l'aide sur un sujet donné, mais que vous ne savez pas quelle est la bonne page d'aide, la fonction `help.search()` est très utile.

Essayez par exemple :

```
> help.search("logarithm")
```

29

help()

- `help(sujet)` que l'on peut aussi écrire `?sujet` affiche la page d'aide pour le sujet ou la fonction sujet.
- Toutes les fonctions de R ont une page d'aide.
- Quand on connaît le nom de la fonction ou du sujet qui nous intéresse, c'est en général le meilleur moyen d'apprendre à l'utiliser.

30

example()

Les pages d'aide sont généralement très détaillées. Elles contiennent souvent, entre autres :

- Une section "See Also" qui donne les pages d'aide sur des sujets apparentés
- Une section "Description" de ce que fait la fonction
- Une section "Exemples" avec du code illustrant ce que fait la fonction documentée. Ces exemples peuvent être exécutés directement en utilisant la fonction `example()` , essayez par exemple :

`example(plot)`

31

Prise en main du logiciel

- Taper **Ctrl L** → pour nettoyer la fenêtre 'Rconsole' sous windows (alt cmd L sous Mac)
- La flèche (clavier) ↑ permet de faire défiler les commandes déjà tapées
- La souris permet de sélectionner, de copier-coller les lignes d'instruction déjà écrites dans Rconsole

32

Opérateurs (1)

- Arithmétique

- Addition +
- Soustraction -
- Multiplication *
- Division /
- Puissance ^

- R comme calculatrice

```
> 10/3
[1] 3.333333
> 25^0.5 # identique à la fonction sqrt(25)
[1] 5
> x=4
> y=3;x+y
[1] 7
> sum(x,y)
```

33

Opérateurs (2)

- Assignment

Permet de créer un objet

```
<-
->
=
```

Exemple

```
>x<-4
>y=3
>5->z
```

La fonction **ls()** liste tous les objets créés présent en mémoire

La fonction **rm()** supprime une variable créée

Exemple: ls(); rm(x); ls()

Pour tout supprimer: **rm(list=ls())**

34

Opérateurs (3)

- Comparaison

- > supérieur
- < inférieur
- <= inf ou égal
- >= sup ou égal
- == égal
- != différent

Opérateur particulier ~ utilisé dans les analyses multivariées

- Logique

- & et
- | ou
- ! non

- Extraction

- [] \$

Exemples

```
>Age<-c(2,4,5,10,8,7,12,11)
>Age[Age>=5 & Age<10]
>Age[Age<5 | Age>10]
>Age[Age!=5]
```

35

Attributs spéciaux

- Qui exécutent une fonction

Condition

ifelse()

if(){} else{}

Répétition

for(){}

while(){}

repeat{}

- Qui ne font rien

NA valeur manquante

Inf infini (**-Inf**)

NULL objet vide

TRUE Vrai

FALSE Faux

- Constantes

letters 26 lettres en minuscule

pi $\pi=3,141593$

36

Exemples

```
>Var.x<-c(1,2,6,10,15,8,9,18,4,14,8,21)
>Var.y<-ifelse(Var.x<10,TRUE,FALSE)
>Var.y
>Var.z<-ifelse(Var.x<3,0,ifelse(Var.x>2&Var.x<9,1,2))
>Var.a<-letters
>Var.b<-letters[1:10]
>Var.c<-c(1:10)
>Var.d<-paste(Var.b,Var.c,sep=" - ")
```

Fonction **paste()** pour concaténer des chaînes .

37

paste()

```
Nom<- " " #écrivez votre nom en majuscule
Prenom<- " " #votre prénom commençant par une
majuscule
Annonce<-paste("Bonjour",Prenom, Nom, sep=" ")
Annonce
```

38

Manipulations des objets « Données »

Données

Création ou importation

Objet « Données » dans R

39

VARIABLES

Une variable est une caractéristique
de quelque nature que ce soit
des individus constituant une population
ou un échantillon

40

Types de variables

- **Quantitatives**

- Continues
(continuous data)

- Discrètes
(discrete data)

- Temporelles

- **Qualitatives**

- Ordinales
(ordinal data)

- Nominales
(nominal data)

- Binaires
(binary or dichotomous data)

41

Variables quantitatives continues

Les valeurs de la variable varient de façon continue et sont exprimées avec une unité de mesure.

Exemples

Poids kg

Taille m

Cholestérol g/l

Pression artérielle cm de hg

La précision de la mesure est le plus souvent limitée par l'instrument de mesure.

42

Variables quantitatives discrètes

La variable ne prend que quelques valeurs entières dénombrables, un nombre fini de valeurs.

Exemples

Congès annuels	Jours/an
Rappel de vaccins	Nbre d' injections
Parité	Nbre d' enfants
Activité de soin	Nbre de consultation

La transformation d' une variable quantitative continue en une variable quantitative discrète s' appelle la discrétisation ou groupement par classe (ex : classes d' âge).

43

Variables temporelles

La variable quantitative qui utilise les unités de mesure du temps

Exemples

Age de grossesse	semaine
Date de naissance	JJ/MM/AAAA
Durée d' hospitalisation	Jours
Délai d' intervention	Heures

Il existe des variables de durée et des variables servant à définir un instant donné (date) à partir duquel il est possible de calculer une durée.

44

Variables qualitatives ordinales

Exemples

La variable qui s'exprime en classes qui peuvent être ordonnées selon un échelle de valeurs	Niveau d'étude	Prim., second., sup.
	Complications d'une maladie	Modérée, moyenne, sévère
	Score	De faible à élevé
	Intensité d'une douleur	De faible à élevé

La codage numérique pour le traitement informatique ne doit pas les faire passer pour des variables qualitatives discrètes et n'autorise donc pas de les manipuler de façon arithmétique.

45

Variables qualitatives nominales

Exemples

Les classes ne peuvent être hiérarchisées pour cette variable.	Groupe sanguin	A,B,O, AB.
	Etat civil	Célibataire, marié, divorcé...
	Couleur des yeux	Bleu, vert, marron, ...
	Toiture	Tuile, tôle, paille...

L'ordre de présentation de ces variables est arbitraire.

46

Variables binaires

Exemples

C' est un type particulier de variables nominales qui ne peuvent prendre que deux valeurs.

Etat de santé	Malade, sain.
Sexe	Homme, femme.
Signe clinique	Présence, absence
Statut vaccinal	Vacciné, non vacciné

Autres appellations: variables dichotomiques, variables booléennes (vrai/faux), variables de Bernouilli codés 0 ou 1.

47

Exercice

A	B	C	D	E	F	G	H
NUMID	IDENTIF	SEXE	DAT_NAI	TAILLE en cm	NATION	Nbre Enfants	NIV_ETU
1	David	M	20/10/58	184	Fr	5	PRIM
2	Benoit	M	12/05/63	165	Mad	3	SUP
3	Laurence	F	05/01/70	156	Belg	3	SECOND
4	Walter	M	31/08/44	176	All	2	SECOND
5	Sylvie	F	23/11/84	163	Mad	1	SUP

De quel type sont les colonnes B,C,D,E,F,G,H?

48

Questionnaire

Inclusion dans l'étude : Enfants de 0 à 5 ans présentant une diarrhée de moins de 5 jours.

Nom : _____ **Prénom :** _____

Age : |_|_| mois **Sexe :** Masculin Féminin

Date de naissance : |_|_|_|_|_|_|_|_|_|_| (jj/mm/aaaa)

Episode diarrhéique

Date de la visite : |_|_|_|_|_|_|_|_|_|_| **Durée de la diarrhée en jour :** |_|_|

Heure de prélèvement de selles : |_|_|_|_|

Selles : Pâteuses Liquide
Présence de sang Glaireuses

Nombre de selles par jour : |_|_| **Température axillaire :** |_|_|_|°C

Poids : |_|_|_|_| Kg **Taille :** |_|_|_|_| cm

Etat nutritionnel : Normal Limite Malnutri

Signes de déshydratation : Oui Non

49

Traitement informatique des données

50

Codage sous R

Inclusion dans l'étude : Enfants de 0 à 5 ans présentant une diarrhée de moins de 5 jours.

Nom : **character** Prénom : **character**
 Age : **numérique** mois Sexe : **Num ou char / factor**

Date de naissance : **Char/date**

Episode diarrhéique

Date de la visite : **Char/date** Durée de la diarrhée en jour : **num**

Heure de prélèvement de selles **Char/ .heure**

Selles : Pâteuses Liquide
 Présence de sang Glaireuses **logical**

Nombre de selles par jour : **numérique** Température axillaire **numérique°C**

Poids **numKg** Taille **num cm**

Etat nutritionnel **factor** Normal Limite Malnutri

Signes de déshydratation Oui Non **logical**

51

Nature du contenu des objets sous R

Nature		
null	Objet vide	NULL
logical	booléen	TRUE, FALSE ou T,F
numeric	Nombre réel	12, 2.56, pi, 1e-10
complex	Nombre complexe	2i
character	Chaîne de caractère	'bonjour', « Female »

Pour connaître le type d'objet

mode()

52

Création des objets « Données » (1)

- **Objets simples** (vecteur à un seul élément)

```
> x<-4
```

- **Vecteurs**

```
>Age<-c(3,5,7,10) #fonction c() pour créer un vecteur
```

```
>mode(Age)
```

```
>Sexe<-c(" Homme ", " Femme ")
```

```
>Sexe<-rep(c(" Homme ", " Femme "),c(15,16)) #fonction rep()
```

```
>mode(Sexe)
```

```
>Tranchage<-rep(c(" 0-4 ", " 5-9 ", " >10 "),c(8,9,14))
```

```
>mode(Tranchage)
```

```
>Poids<-seq(10,25,by=.5) #fonction seq()
```

Accéder aux éléments d'un vecteur

```
>Poids[16]
```

```
>Poids[c(1,16,31)]
```

```
>Poids[Poids>20]
```

53

Factoriser un vecteur sous R

fonctions		
factor	Factorisation	Male,Female
ordered	Factorisation ordonnée	Primaire, secondaire, universitaire

```
Niveau<-factor(rep(c("uni","sec","prim","uni","sec"),c(5,16,18,22,7)))  
table(Niveau)
```

```
Niveau2<-ordered(rep(c("uni","sec","prim","uni","sec"),c(5,16,18,22,7)), levels=c("uni","sec","prim"))  
table(Niveau2)
```

54

Manipulation des dates

- Les champs « dates » importés sont de type character
- On utilise alors les fonctions `as.Date()` ou `strptime()`
- `Dcalend<-c("17/02/92","27/02/92","14/01/92",
"28/02/92","01/02/92","02/01/92")`
- `Dcalend`
- `Dcalend.2<-as.Date(Dcalend,"%d/%m/%y")`
- `Dcalend.2`

55

Manipulation des dates

- `Dcalend.1<-c("17/02/1992","27/02/1992","14/01/1992",
"28/02/1992","01/02/1992","02/01/1992")`
- `Dcalend.21<-as.Date(Dcalend.1,"%d/%m/%y")`
- `Dcalend.21`
- `Dcalend.22<-as.Date(Dcalend.1,"%d/%m/%Y")`
- `Dcalend.22`

56

Manipulation des dates

- `difftime()`
- `Temps<-difftime(Dcalend.21,Dcalend.22, units="days")`
- Temps
- Créer un objet date mois par mois
- `Dcalend.3<-seq(from=as.Date("15/01/08","%d/%m/%y "), to = as.Date("15/12/08", "%d/%m/%y"), by="month")`

57

Manipulation des dates

- `as.character(Dcalend21, "%U")`
#numéro de semaine
- `as.character(Dcalend21, "%Y")`
#année à quatre chiffres
- `as.character(Dcalend21, "%y")`
#année à deux chiffres
- `as.character(Dcalend21,"%j")`
#jour de l'année

58

Création des objets « Données » (2)

- **Matrices**

Série de valeurs à 2 dimensions (lignes et colonnes)

```
>Chiffre<-c(1:15) #vecteur des chiffres de 1 à 15
```

```
>Chiffre
```

```
>Matrice.1<-matrix(Chiffre,ncol=3)
```

```
>Matrice.1
```

- **Matrices (autres méthodes)**

```
>Matrice.1<-c(1:15) #création d'un vecteur de 1 à 15
```

```
>dim(Matrice.1)<-c(5, 3) # dim() transforme le vecteur en matrice
```

```
>Matrice.1
```

Création d'une matrice à 3 lignes et remplissage en commençant par les lignes

```
>Matrice.2<-matrix((1:15)*100,nrow=3, byrow=TRUE)
```

```
>Matrice.2
```

59

Création des objets « Données » (3)

- **Data frames** (les colonnes sont des variables)

```
>Dataframe.1<-data.frame(Matrice.2)
```

```
>Dataframe.1
```

```
>names(Dataframe.1)[1]<- "TOUS" #fonction names() pour donner un nom à une variable, ici c'est le nom de la première colonne
```

```
Dataframe.1[,2]
```

>Taille<-c(130,132,145,136,140)

```
>Poids<-c(45,46,48,34,44)
```

```
>Sexe<-c(1,2,1,2,1)
```

```
>Age<-c(12,11,13,12,12)
```

```
>Dataframe.2<-data.frame(Taille,Poids,Sexe,Age)
```

```
>Dataframe.2
```

60

Création des objets « Données » (4)

- Data frames

```
Sexe<-rbinom(31,size=1,prob=.48) #distribution binomiale
Sexe1<-ifelse(Sexe==0,'Femme','Homme')
Mydata<-data.frame(Sexe,Sexe1,Poids,Tranchage)
Mydata[1:5,]
summary(Mydata)
Mydata$Sexe<-as.factor(Mydata$Sexe)
summary(Mydata)
```

```
NB : rnorm() #fonction pour une distribution normale
      rnorm(10) ou rnorm(10,mean=100)
      runif() #fonction pour une distribution uniforme
      runif(10) ou runif(10,min=20,max=21)
```

61

Avant de finir

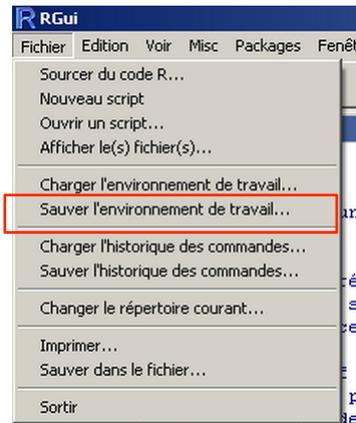
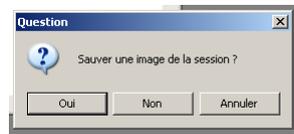
- Penser à enregistrer votre script pour bien garder l'ensemble de vos commandes.



62

L'environnement de travail

- Les objets « données » créés au cours d'une session peuvent être sauvegardés. Les fichiers de données R portent l'extension .Rdata
- La commande « sauver l'environnement de travail » copie toutes les données en mémoire dans un fichier .Rdata
- À la fermeture, R vous propose de sauvegarder l'environnement de travail.

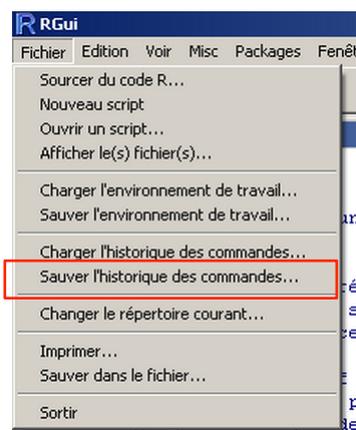


Il est recommandé de créer un dossier pour chaque analyse de données indépendantes

63

Historique des commandes

- Les commandes utilisées au cours d'une session peuvent être sauvegardées.
- La commande « sauver l'historique des commande » copie toutes les commande en mémoire dans un fichier.txt



Il est recommandé de créer un dossier pour chaque analyse de données indépendantes

64