

Analyse de données avec le logiciel R

Michaël Genin, Guillemette Marot

Université de Lille 2

EA 2694 - Santé Publique : Epidémiologie et Qualité des soins

michael.genin@univ-lille2.fr

guillemette.marot@univ-lille2.fr

Programme de la formation

Programme - Module 1

Journée 1

- Introduction générale
 - Présentation du logiciel R
 - Installation du logiciel R et packages
- Les bases d'utilisation du logiciel R
 - Importation et exportation de données
 - Manipulation de données
- Statistique descriptive et estimation
 - Représentations graphiques
 - Indicateurs statistiques

Journée 2

- Estimation et Tests statistiques usuels
 - Intervalles de confiance
 - Test de normalité
 - Tests de comparaison de moyennes ou de proportions
 - Test du χ^2
- Initiation à la production de rapports avec RMarkdown
- Lancement du projet phase 1

Programme - Module 2

Journée 3

- Correction du mini-projet donné à la fin du module 1
- Lien entre plusieurs variables
 - Corrélation et régression linéaire multiple
 - ANOVA
 - Régression linéaire multiple

Journée 4

- Initiation aux analyses multivariées
 - Classification ascendante hiérarchique
 - K-means
 - Régression logistique
 - Tests multiples
- Introduction au package ggplot2 (visualisation de données)

Programme - Module 3

Journée 5

- Correction du mini-projet donné après le module 2
- Synthèse

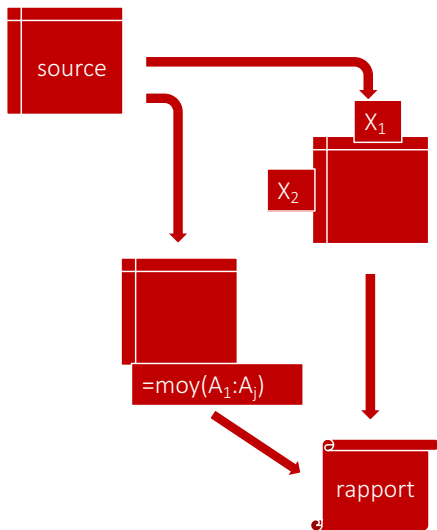
Point étudié

- 1 **Introduction générale**
 - Différentes façons de travailler
 - Présentation du logiciel R
 - Installation du logiciel R
- 2 Utilisation du logiciel R
- 3 Analyse de données avec le logiciel R - Partie 1
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2

Point étudié

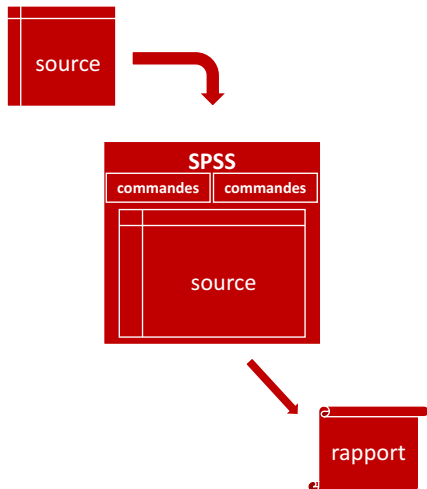
- 1 Introduction générale
 - Différentes façons de travailler
 - Présentation du logiciel R
 - Installation du logiciel R
- 2 Utilisation du logiciel R
- 3 Analyse de données avec le logiciel R - Partie 1
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2

Exemple d'analyse de données avec Excel



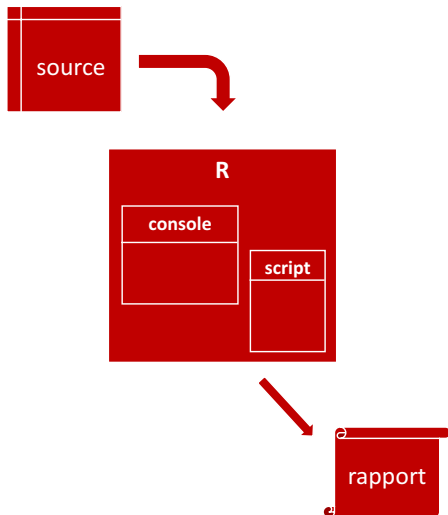
- Ouverture du jeu de données (.xls) :
feuille ou tableur Excel
- Tableaux croisés dynamiques
- Moyenne, médiane, écart type,...
- Copier / Coller dans un Rapport final
(Word)

Exemple d'analyse de données avec SPSS



- Import du jeu de données à la main
- Clic bouton :
 - Analyse 1 = sortie 1
(graphique/nombre/tableau)
 - Analyse 2 = sortie 2
 - Analyse 3 = sortie 3
 - ...
- Sorties non utilisables telles quelles pour le rapport
- Copier / Coller dans un Rapport final (Word)

Exemple d'analyse de données : l'esprit R



- Ouverture du logiciel
- Édition d'un script :
 - Importation des données
 - "Nettoyage", définition de nouvelles variables, typage, etc. = *data management*
 - Analyses (descriptive uni/bivariée, modélisation, . . .)
 - Edition de graphiques
 - Exportation d'un rapport
- Rapport

Point étudié

- 1 Introduction générale
 - Différentes façons de travailler
 - **Présentation du logiciel R**
 - Installation du logiciel R
- 2 Utilisation du logiciel R
- 3 Analyse de données avec le logiciel R - Partie 1
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2

Le logiciel R

- Créé par Ross Ihaka & Robert Gentleman en 1997
- Logiciel gratuit et open source pour l'analyse statistique (calculs et graphiques)
- Version pour UNIX (Linux), Windows et Macintosh
- Facile à installer, léger en ressources

- Complet, nombreuses extensions (bibliothèques)
- Communauté importante et active sur les forums.
- Très utilisé dans le monde universitaire
- Les avantages d'un langage de programmation
 - Liberté de création (fonctions, graphiques)
 - Puissance algorithmique

Point étudié

- 1 Introduction générale
 - Différentes façons de travailler
 - Présentation du logiciel R
 - **Installation du logiciel R**
- 2 Utilisation du logiciel R
- 3 Analyse de données avec le logiciel R - Partie 1
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2

Logiciel R

Site web de référence : <https://www.r-project.org>

- 1)
- 2) Choix d'un miroir : le plus proche (géographie : France)
- 3) Téléchargement



[Home]

Download

CRAN

R Project

About R

Logo

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

- Windows : *base*
- MacOS : *.pkg*
- Linux : suivre les instructions en fonction de la distribution

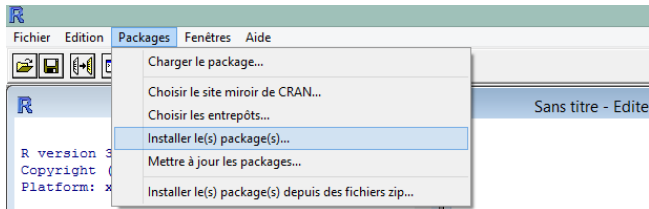
Remarque

- R peut être installé sur une clé usb → absence de droits admin

Installation et utilisation de package

Installer un package :

- Via la commande : `install.packages("maptools")`
- Via le menu



Utiliser un package : chargement dans l'environnement

- Chargement *via* le menu déroulant
- Chargement *via* la commande : `library("maptools")`

Point étudié

- 1 Introduction générale
- 2 **Utilisation du logiciel R**
 - Interface
 - Logiciel R Studio
 - Notion d'objet
 - Fonctions
 - Aide au sein du logiciel R
 - Vecteurs
 - Matrices
 - Dataframe
 - Manipulations avancées sur les dataframes
 - Gestion des données manquantes
- 3 Analyse de données avec le logiciel R - Partie 1
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2

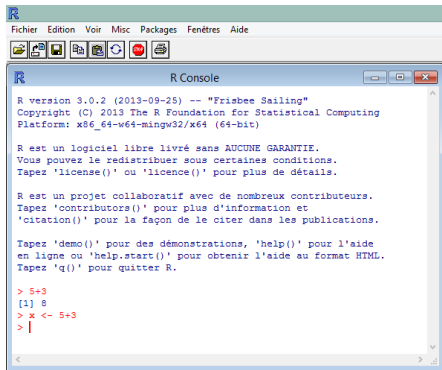
Point étudié

- 1 Introduction générale
- 2 **Utilisation du logiciel R**
 - **Interface**
 - Logiciel R Studio
 - Notion d'objet
 - Fonctions
 - Aide au sein du logiciel R
 - Vecteurs
 - Matrices
 - Dataframe
 - Manipulations avancées sur les dataframes
 - Gestion des données manquantes
- 3 Analyse de données avec le logiciel R - Partie 1
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2

L'interface de R

Console

- Saisie des commandes
- Affiche parfois le résultat des commandes
- Pas de sauvegarde des commandes



```
R
Fichier  Edition  Voir  Misc  Packages  Fenêtres  Aide

R Console

R version 3.0.2 (2013-09-25) -- "Frisbee Sailing"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-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.

> 5+3
[1] 8
> x <- 5+3
> |
```

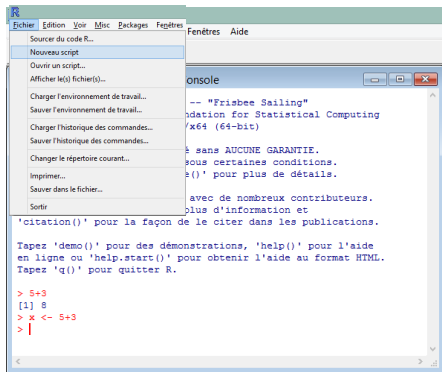
L'interface de R

Console

- Saisie des commandes
- Affiche parfois le résultat des commandes
- Pas de sauvegarde des commandes

Fenêtre de script

- Mémorisation / Edition / Réutilisation
- Execution de la ligne courante ou d'une portion de code sélectionnée :
 - Windows : Ctrl + R
 - MacOS : cmd + entrée



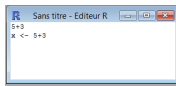
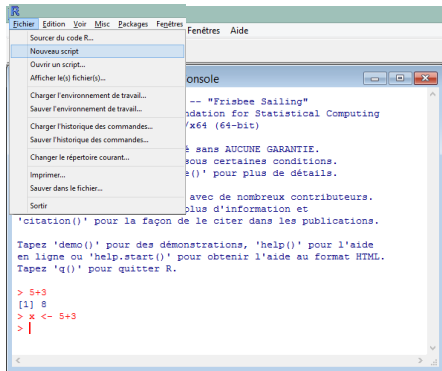
L'interface de R

Console

- Saisie des commandes
- Affiche parfois le résultat des commandes
- Pas de sauvegarde des commandes

Fenêtre de script

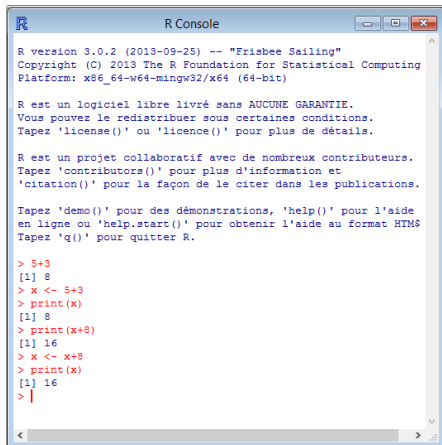
- Mémorisation / Edition / Réutilisation
- Execution de la ligne courante ou d'une portion de code sélectionnée :
 - Windows : Ctrl + R
 - MacOS : cmd + entrée



L'interface de R

Commandes

- Expression :
 - Évaluée directement
 - Résultat affiché dans la console
- Affectation
 - Expression évaluée
 - Résultat stocké dans un **objet**
 - Symbole : `<-`
 - Ex : `x <- 5+3`
 - Affichage d'un objet : `print(x)`
 - *N.B.* : non conseillé d'utiliser le signe `=` pour l'affectation
- Objet
 - `=` Variable
 - Chiffre, chaîne de caractères, etc...
 - Nom sensible à la casse



```
R
R Console

R version 3.0.2 (2013-09-25) -- "Frisbee Sailing"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-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.

> 5+3
[1] 8
> x <- 5+3
> print(x)
[1] 8
> print(x+8)
[1] 16
> x <- x+8
> print(x)
[1] 16
> |
```

L'interface de R - Ecrire le script suivant sous R

- Affecter la valeur 30 à la variable Age
- Afficher Age
- Saisie d'une expression sans affectation :
`print(expression)`
- Affecter le résultat du calcul $23+7*2$ à Age
- Afficher Age
- Affecter la valeur "Jacques" à la variable
Prenom
- Afficher prenom
- Utiliser paste pour afficher "Jacques a 37
ans"

Rappels

```
variable <- valeur  
print(variable)  print(expression)  
paste("chaine1",variable,"chaine2")
```

L'interface de R - Ecrire le script suivant sous R

- Affecter la valeur 30 à la variable Age
- Afficher Age
- Saisie d'une expression sans affectation :
print(expression)
- Affecter le résultat du calcul $23+7*2$ à Age
- Afficher Age
- Affecter la valeur "Jacques" à la variable Prenom
- Afficher prenom
- Utiliser paste pour afficher "Jacques a 37 ans"

```
> Age<-30
> print(Age)
[1] 30
> Age<-23+7*2
> print(Age)
[1] 37
> Prenom<-"Jacques"
> print(Prenom)
[1] "Jacques"
> paste(Prenom, "a", Age, "ans")
[1] "Jacques a 37 ans"
>
```

Rappels

```
variable <- valeur
print(variable)  print(expression)
paste("chaine1",variable,"chaine2")
```

Point étudié

- 1 Introduction générale
- 2 **Utilisation du logiciel R**
 - Interface
 - **Logiciel R Studio**
 - Notion d'objet
 - Fonctions
 - Aide au sein du logiciel R
 - Vecteurs
 - Matrices
 - Dataframe
 - Manipulations avancées sur les dataframes
 - Gestion des données manquantes
- 3 Analyse de données avec le logiciel R - Partie 1
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2

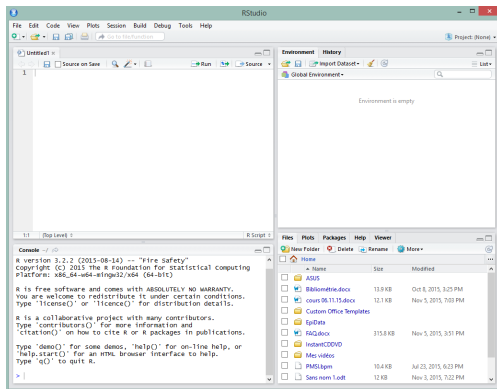
Le logiciel R Studio

- Environnement de développement intégré
- Téléchargement et installation

<https://www.rstudio.com>

Products → Rstudio → Desktop → Open Source Edition

- L'interface R Studio
 - Console
 - Editeur de scripts
 - Navigateur d'environnement
 - Panneau multifonctions :
 - Explorateur fichiers
 - Explorateur packages
 - Aide
 - Figures



Point étudié

- 1 Introduction générale
- 2 **Utilisation du logiciel R**
 - Interface
 - Logiciel R Studio
 - **Notion d'objet**
 - Fonctions
 - Aide au sein du logiciel R
 - Vecteurs
 - Matrices
 - Dataframe
 - Manipulations avancées sur les dataframes
 - Gestion des données manquantes
- 3 Analyse de données avec le logiciel R - Partie 1
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2

Notion d'objet

- "Brique élémentaire" du langage R
- Différents objets : vecteurs, matrices, listes, dataframe, ...
- Les objets se différencient par
 - Leur **mode** qui décrit leur contenu (nature des données)
 - Leur **classe** qui décrit leur structure (structure des données)
- On distingue également deux types d'objets :
 - Objets atomiques (mode homogène)
 - Objet récurifs (mode hétérogène) → listes, ...

Mode : nature des données sous R

Types les plus courants :

- Réel (integer ou numeric)
- Chaîne de caractères (character)
- Booléen (boolean)
- Complexe...
- Fonction `mode()` permet de connaître le mode d'un objet

```
> age<-35
> mode(age)
[1] "numeric"
> nom<-"Rémi"
> mode(nom)
[1] "character"
> bool<-T
> mode(bool)
[1] "logical"
```

Classe : structure des données sous R

Types les plus courants :

- Vecteur : `c()`, `seq()`, `:`, `rep()`
- Matrices : `matrix()`
- Tableau multidimensionnel : `array()`
- Liste : `list()`
- Tableau individus \times variables : `data.frame()`
- Fonction `class()` permet de connaître la classe d'un objet

Règles pour le nom d'un objet

- Constitué uniquement de caractères alphanumériques et deux symboles
 - a-z, A-Z, 0-9, "_", "."
 - Bon exemple : Longueur_bras
 - Mauvais exemple : Longueur&bras
- Nom d'objet est *case sensitive* : R fait la distinction entre majuscule et miniscule
 - Longueur_bras \neq longueur_bras
- Un nom d'objet ne peut pas commencer par un chiffre

Point étudié

- 1 Introduction générale
- 2 Utilisation du logiciel R
 - Interface
 - Logiciel R Studio
 - Notion d'objet
 - **Fonctions**
 - Aide au sein du logiciel R
 - Vecteurs
 - Matrices
 - Dataframe
 - Manipulations avancées sur les dataframes
 - Gestion des données manquantes
- 3 Analyse de données avec le logiciel R - Partie 1
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2

Fonctions sous R

Utilité

- Manipuler des objets : importer et exporter des données, générer des objets...
- Réaliser des opérations sur les objets : calcul de moyenne, quantile, mesurer la longueur d'un objet...
- Afficher du texte, des graphiques
- ...

Généralités

- Définie par son nom et ses paramètres
- Certains paramètres obligatoires, d'autres optionnels (`args()`)
- Forme générale : `nom_fonction(par1=valeur1,par2=valeur2,...)`
- Possibilité de créer ses propres fonctions : `function(...){...}`

Point étudié

- 1 Introduction générale
- 2 **Utilisation du logiciel R**
 - Interface
 - Logiciel R Studio
 - Notion d'objet
 - Fonctions
 - **Aide au sein du logiciel R**
 - Vecteurs
 - Matrices
 - Dataframe
 - Manipulations avancées sur les dataframes
 - Gestion des données manquantes
- 3 Analyse de données avec le logiciel R - Partie 1
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2

Aide au sein du logiciel R

- Aide générale de R : `help.start()`
- Aide sur une fonction :
 - `help("Nom_fonction")` ou `?Nom_fonction`
 - Nom de fonction inconnu : `help.search("Mot-clé")`
- Obtenir l'ensemble des fonctions d'un package : `library(help="Nom_package")`
 - Exemple : `library(help="stats")`
- Recherche dans les listes de diffusion (mailing lists) :
`"RSiteSearch("mot_clé")`

Point étudié

- 1 Introduction générale
- 2 Utilisation du logiciel R
 - Interface
 - Logiciel R Studio
 - Notion d'objet
 - Fonctions
 - Aide au sein du logiciel R
 - **Vecteurs**
 - Matrices
 - Dataframe
 - Manipulations avancées sur les dataframes
 - Gestion des données manquantes
- 3 Analyse de données avec le logiciel R - Partie 1
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2

Vecteurs - Généralités

- Objet atomique : même mode pour tous les éléments
- Etiquette possible pour chacun des éléments du vecteur

```
> mon_vect<-c(1,2,3)
> print(mon_vect)
[1] 1 2 3
> print(mon_vect<-c(1,2,"a"))
[1] "1" "2" "a"
> class(mon_vect)
[1] "character"
> (mon_vect<-c("a"=1,"b"=2,"c"=3))
a b c
1 2 3
> |
```

Fonction de base pour créer un vecteur :
`c(element1, element2,..., elementp)`

Vecteurs - Générer des données

```
> (mon_vect1 <- 1:3)
[1] 1 2 3
> (mon_vect2 <- rep("a", 4))
[1] "a" "a" "a" "a"
> (mon_vect3 <- seq(from=10, to=20, by=5))
[1] 10 15 20
# en combinant ces fonctions
> (mon_vect4 <- c(rep("a", 3), rep("b", 3)))
[1] "a" "a" "a" "b" "b" "b"
> (mon_vect5 <- c(rep(seq(0, 2, 1), 3), seq(0, 1, 1)))
[1] 0 1 2 0 1 2 0 1 2 0 1
# en combinant des vecteurs
> (mon_vect6 <- c(mon_vect3, mon_vect1, 20))
[1] 10 15 20 1 2 3 20
```

`rep(par1, par2)` (`par1` : objet à répéter `par2` : nombre de répétitions)

`seq(par1, par2, par3)` (`par1` : from `par2` : to `par3` : by)

Vecteurs - Manipuler des données

```
# beaucoup d'autres fonctions, entre autres :  
> mon_vect6  
[1] 10 15 20 1 2 3 20  
> rev(mon_vect6)      #- renverser un vecteur  
[1] 20 3 2 1 20 15 10  
> unique(mon_vect6)   #- extraire les éléments différents  
[1] 10 15 20 1 2 3  
> sort(mon_vect6)     #- trier par ordre croissant  
[1] 1 2 3 10 15 20 20  
> head(mon_vect6, 3)  #- extraire les n premières valeurs  
[1] 10 15 20  
> length(mon_vect6)   #- longueur du vecteur (nb d'elements)  
[1] 7
```

Point étudié

- 1 Introduction générale
- 2 **Utilisation du logiciel R**
 - Interface
 - Logiciel R Studio
 - Notion d'objet
 - Fonctions
 - Aide au sein du logiciel R
 - Vecteurs
 - **Matrices**
 - Dataframe
 - Manipulations avancées sur les dataframes
 - Gestion des données manquantes
- 3 Analyse de données avec le logiciel R - Partie 1
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2

Matrices

- Création
 - `matrix(données,nrow=,ncol=,byrow=FALSE)`
- Manipulations basiques
 - `cbind()` : fusionne par colonnes
 - `rbind()` : fusionne par lignes
 - `apply()` : applique une fonction aux lignes (`MARGIN=1`) ou aux colonnes (`MARGIN=2`)
 - `sweep()` : "soustrait" une valeur aux lignes (`MARGIN=1`) ou aux colonnes (`MARGIN=2`)
- N.B. : Certaines fonctions s'utilisent aussi bien sur les matrices que sur les data frames.

Point étudié

- 1 Introduction générale
- 2 Utilisation du logiciel R
 - Interface
 - Logiciel R Studio
 - Notion d'objet
 - Fonctions
 - Aide au sein du logiciel R
 - Vecteurs
 - Matrices
 - **Dataframe**
 - Manipulations avancées sur les dataframes
 - Gestion des données manquantes
- 3 Analyse de données avec le logiciel R - Partie 1
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2

Dataframe

- Tableau individus \times variables
- Plusieurs modes possibles pour les variables du dataframe

Fonctions de génération ou de chargement

- `data.frame()` : génération de données
- `read.table()` : lecture d'un fichier de données (séparateur tabulation)
- `read.csv()` : lecture d'un fichier de données (séparateur ",")
- `read.csv2()` : lecture d'un fichier de données (séparateur ";")
- package `foreign` pour les formats propriétaires (SPSS, SAS, ...)

```
> IMC.data<-data.frame(ID=c("Pierre","Pierrette","Jean","Jeannette","Bernard","Bernadette"),
+                       Taille=c(1.83,1.76,1.82,1.60,1.90,1.66),
+                       Poids=c(67,58,66,48,75,55),Sexe=c("H","F","H","F","H","F"))
> IMC.data
```

	ID	Taille	Poids	Sexe
1	Pierre	1.83	67	H
2	Pierrette	1.76	58	F
3	Jean	1.82	66	H
4	Jeannette	1.60	48	F
5	Bernard	1.90	75	H
6	Bernadette	1.66	55	F

```
>
```

Dataframe - Chargement de données

Données nutriage : un échantillon de personnes âgées résidant à Bordeaux (Gironde, France) a été interrogé en 2000 dans le cadre d'une enquête nutritionnelle. L'échantillon est constitué de 226 sujets et 13 variables.

① Fonction `read.table()` et fichier **nutriage.txt**

```
nutriage.data<-read.table("chemin/nutriage.txt",sep='\t', header=T)
```

`sep='\t'` : séparateur tabulation

`header=T` : la première ligne contient les noms de variables

② Fonction `read.csv2()` et fichier **nutriage.csv**

```
nutriage.data<-read.csv2("chemin/nutriage.csv",dec=".")
```

`dec='.'` : séparateur décimal (à préciser car défaut : ",")

Par défaut, le fichier comprend en première ligne les noms de variables

Dataframe - Chargement de données

Notion de répertoire de travail : il faut donner l'adresse **absolue** du fichier à charger. Il est possible de définir un **répertoire de travail** dans lequel se trouvera le script R ainsi que le(s) fichier(s) de données :

```
setwd("chemin")
```

- ❶ Fonction `read.table()` et fichier **nutriage.txt**

```
nutriage.data<-read.table("nutriage.txt",sep='\t', header=T)
```

- ❷ Fonction `read.csv2()` et fichier **nutriage.csv**

```
nutriage.data<-read.csv2("nutriage.csv",dec=".")
```

Astuce

Sous Windows : pour obtenir le chemin d'un fichier

- Clic droit + propriétés
- Ouvrir une Invite de commande et glisser le fichier dans l'invite
- Remplacer les `"\"` par des `"/`

Dataframe - Exportation de données

❶ Fonction `write.table()`

```
write.table(nutriage.data, "nutriage_export.txt", sep='\t', quote=F, row.names=F)
```

❷ Fonction `write.csv2()`

```
write.csv2(nutriage.data, "nutriage_export.csv", row.names=F)
```

Remarques

- Penser à indiquer l'extension dans le nom de fichier de destination
- `write.table()` est une fonction générique pour tout type de fichier

```
write.csv2(nutriage.data, "nutriage_export.csv", row.names=F)
```

```
write.table(nutriage.data, "nutriage_export.csv", sep=';', quote=F, row.names=F)
```

Dataframe - Manipulations basiques

Fonctions utiles

- `dim()` : affiche le nombre d'individus et le nombre de variables
- `ncol()` : affiche le nombre de colonnes
- `nrow()` : affiche le nombre de lignes
- `dimnames()` : affiche le nom des lignes et des colonnes
- `rownames()` : affiche le nom des lignes (individus)
- `colnames()` : affiche le nom des colonnes (variables)
- `head()` : affiche les premières lignes
- `tail()` : affiche les dernières lignes
- `str()` : affiche la structure du jeu de données
- `transform()` : applique une transformation sur une colonne

Point étudié

- 1 Introduction générale
- 2 Utilisation du logiciel R
 - Interface
 - Logiciel R Studio
 - Notion d'objet
 - Fonctions
 - Aide au sein du logiciel R
 - Vecteurs
 - Matrices
 - Dataframe
 - **Manipulations avancées sur les dataframes**
 - Gestion des données manquantes
- 3 Analyse de données avec le logiciel R - Partie 1
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2

Principe de l'indiçage

Extraire / Remplacer les éléments d'un objet

① Vecteurs

- Extraction : `vect[indice]`
- Remplacement : `vect[indice]<-Valeur`

② Matrices

- Extraction d'un élément : `mat[indice_ligne,indice_colonne]`
- Extraction d'une ligne : `mat[indice_ligne,]`
- Extraction d'une colonne : `mat[,indice_colonne]`
- Remplacement : `mat[indice_ligne,indice_colonne]<-Valeur`

③ Dataframe

- Extraction d'une variable
 - `df$nom_variable`
 - `df[,indice_variable]`
 - `df[, "nom_variable"]`
- Remplacement d'une variable : `<- vect`

Fonctions avancées

Sélection dans un dataframe

```
subset(dataframe, subset=expression_logique,selection=liste_variables)
```

Calcul d'indicateurs statistiques sur une ou plusieurs variables par agrégats de données

```
aggregate(liste_variables, by=list(Var_groupe1=var1,...), FUN=function)
```

Jointure de 2 dataframes

```
merge(df1,df2, by=liste_variables, by.x=,by.y=,...)
```

Ordonner un dataframe selon une variable

```
df[order(df$nom_var),]
```

Ordonner un dataframe selon plusieurs variables (package doBy)

```
orderBy(~var1 + var2+..., data=df)
```

Point étudié

- 1 Introduction générale
- 2 Utilisation du logiciel R
 - Interface
 - Logiciel R Studio
 - Notion d'objet
 - Fonctions
 - Aide au sein du logiciel R
 - Vecteurs
 - Matrices
 - Dataframe
 - Manipulations avancées sur les dataframes
 - **Gestion des données manquantes**
- 3 Analyse de données avec le logiciel R - Partie 1
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2

Gestion des données manquantes

Plusieurs fonctions ne pourront être évaluées en présence de valeurs manquantes, désignées par NA (*Not Available*) dans R.

- `is.na()` : indique les valeurs manquantes
- `complete.cases()` : indique les lignes d'un data frame (individus) n'ayant aucune valeur manquante
- `na.omit()` : enlève les individus avec au moins une valeur manquante

Point étudié

- 1 Introduction générale
- 2 Utilisation du logiciel R
- 3 Analyse de données avec le logiciel R - Partie 1**
 - Statistique descriptive
 - Estimation et tests de comparaison
 - Tests d'indépendance et d'adéquation
 - Tests non-paramétriques
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2

Point étudié

- 1 Introduction générale
- 2 Utilisation du logiciel R
- 3 **Analyse de données avec le logiciel R - Partie 1**
 - **Statistique descriptive**
 - Préambule
 - Analyses descriptives
 - Estimation et tests de comparaison
 - Tests d'indépendance et d'adéquation
 - Tests non-paramétriques
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2

Point étudié

- 1 Introduction générale
- 2 Utilisation du logiciel R
- 3 **Analyse de données avec le logiciel R - Partie 1**
 - **Statistique descriptive**
 - **Préambule**
 - Analyses descriptives
 - Estimation et tests de comparaison
 - Tests d'indépendance et d'adéquation
 - Tests non-paramétriques
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2

Analyses descriptives

Afin de donner une bonne description des variables et d'utiliser les bons outils graphiques, il est essentiel de définir correctement le **type de variable**.

Exemple : Données nutriage, disponibles à l'adresse suivante :

<http://www.biostatisticien.eu/springeR/>

Type de variable dans les dataframes

- Nominale : `as.factor()`
- Ordinale : `as.ordered()`
- Quantitative discrète : `as.integer()`
- Quantitative continue : `as.numeric()`

Fonctions utiles

- `levels()` permet de renommer les modalités d'une variables nominale ou ordinale
- `str()` permet de vérifier le type de variable
- `attach()` permet de travailler avec un dataframe particulier

Point étudié

- 1 Introduction générale
- 2 Utilisation du logiciel R
- 3 **Analyse de données avec le logiciel R - Partie 1**
 - **Statistique descriptive**
 - Préambule
 - **Analyses descriptives**
 - Estimation et tests de comparaison
 - Tests d'indépendance et d'adéquation
 - Tests non-paramétriques
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2

Création de tables

- `table()` : tableau de fréquences pour 1 variable ou tableau de contingence (croisé) pour 2 variables
- `margin.table()` : calcule les marges d'un tableau de contingence
- `addmargins()` : ajoute total (ou autre fonction) sur lignes et/ou colonnes
- `prop.table()` : tableau de contingence avec fréquences relatives

Résumés numériques utiles

- `min()`, `max()` : minimum, maximum
- `mean()` : moyenne; `median()` : médiane
- `summary()` : min, max, moyenne, 3 quartiles (Q_1 , Q_2 = médiane, Q_3)
- `var()`, `sd()` : variance et écart-type
- `IQR()` : distance inter-quartile ($Q_1 - Q_3$)
- Autres : `range()`; `quantile()`, ...

Représentations graphiques

En fonction de la nature des variables !

- `barplot()` : diagramme en bâtons
- `pie()` : diagramme circulaire (camembert)
- `boxplot()` : diagramme en boîte à moustaches
- `hist()` : histogramme
- `plot()` : fonction générique applicable à plusieurs types d'objets
- `mosaicplot()` : diagramme en mosaïque

Remarque

Ces fonctions comportent une multitude de paramètres permettant une personnalisation d'affichage (titre, axes, taille, couleur, etc.)

Point étudié

- 1 Introduction générale
- 2 Utilisation du logiciel R
- 3 **Analyse de données avec le logiciel R - Partie 1**
 - Statistique descriptive
 - **Estimation et tests de comparaison**
 - Tests d'indépendance et d'adéquation
 - Tests non-paramétriques
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2

Bref rappel sur l'estimation et les tests statistiques

Estimation : à partir d'un $\{X_i\}$ de taille n , on estime les borne a et b d'un intervalle pour un paramètre inconnu θ tels que

$$\mathbb{P}(a \leq \theta \leq b) = 1 - \alpha$$

Tests d'hypothèses : \mathcal{H}_0 : hypothèse nulle ; \mathcal{H}_1 : hypothèse alternative

Décision	Réalité	
	\mathcal{H}_0	\mathcal{H}_1
\mathcal{H}_0	Niveau de confiance $1 - \alpha$	β
\mathcal{H}_1	α	Puissance $1 - \beta$

Pour rejeter \mathcal{H}_0 , la probabilité critique (*p-value*) est comparée au risque α

Estimation et test sur 1 ou 2 moyennes

La fonction `t.test()` permet d'estimer l'intervalle de confiance et / ou de tester des hypothèses sur une 1 ou 2 moyennes.

- 1 population : `t.test(x,...)`
- 2 populations indépendantes :
 - `t.test(x,y,var.equal=T,...)` (variances égales)
 - `t.test(x,y,var.equal=F,...)` (variances inégales, par défaut)
 - `t.test(x~y, ...)`
- 2 populations appariées :
 - `t.test(x,y,paired=T, ...)`
 - `t.test(x~y,paired=T, ...)`

Estimation et test sur 1 ou 2 proportions

La fonction `prop.test()` permet d'estimer l'intervalle de confiance et/ou tester des hypothèses sur 1 ou 2 proportions, en utilisant l'approximation d'une loi Binomiale $\mathcal{B}(n, p)$ par la loi Normale $\mathcal{N}(np, \sqrt{np(1-p)})$ sous la condition :

$$n > 30, \min\{np; n(1-p)\} > 5$$

- 1 population : `prop.test(x,n,...)`
- 2 populations indépendantes : `prop.test(c(x1,x2),c(n1,n2),...)`

N.B. : la fonction `binom.test()` produit l'intervalle de confiance et le test exact (sans approximation normale)

Point étudié

- 1 Introduction générale
- 2 Utilisation du logiciel R
- 3 **Analyse de données avec le logiciel R - Partie 1**
 - Statistique descriptive
 - Estimation et tests de comparaison
 - **Tests d'indépendance et d'adéquation**
 - Tests non-paramétriques
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2

Tests d'indépendance et d'adéquation

- `chisq.test()` : test d'indépendance ou d'adéquation basés sur l'approximation par une loi du χ^2
- `fisher.test()` : test exact de Fisher
- `shapiro.test()` : test de normalité d'une distribution (attention, le but est de conserver \mathcal{H}_0 !)
- `ks.test()` : test de Kolmogorov-Smirnov pour 1 ou 2 distributions.

Point étudié

- 1 Introduction générale
- 2 Utilisation du logiciel R
- 3 **Analyse de données avec le logiciel R - Partie 1**
 - Statistique descriptive
 - Estimation et tests de comparaison
 - Tests d'indépendance et d'adéquation
 - **Tests non-paramétriques**
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2

Tests non-paramétriques

La fonction `wilcox.test()` produit des tests basés sur le rang des observations (tests de Wilcoxon ou Mann-Whitney) et représentent une alternative aux tests paramétriques classiques.

- 1 population : `wilcox.test(x,...)`
- 2 populations indépendantes :
 - `wilcox.test(x,y,...)`
 - `wilcox.test(y~x,...)`
- 2 populations appariées : `wilcox.test(x,y,paired=T,...)`

Point étudié

- 1 Introduction générale
- 2 Utilisation du logiciel R
- 3 Analyse de données avec le logiciel R - Partie 1
- 4 Initiation à la production de rapports avec RMarkdown**
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2

cf Exemple sur les accouchements prématurés

Point étudié

- 1 Introduction générale
- 2 Utilisation du logiciel R
- 3 Analyse de données avec le logiciel R - Partie 1
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1**
- 6 Analyse de données avec le logiciel R - Partie 2

Accord de crédit

Objectif

La base de données sur le crédit allemand contient des renseignements concernant 1000 clients ayant contracté un prêt à une banque.

700 de ces clients ont remboursé leurs prêts sans difficulté, tandis que 300 ont eu des difficultés à rembourser leurs prêts (variable `RESPONSE`).

L'objectif de cette étude est de construire un modèle permettant de prédire le risque de défaillance d'un client au moment de l'étude du prêt. Si ce risque est trop élevé, la banque refusera d'accorder un prêt à ce client. On calculera un seuil en fonction de critère économique.

Sources : François Kauffman (université de Caen) - Hans Hofmann (Université de Hambourg) (<http://archive.ics.uci.edu/ml/index.php>)

Accord de crédit

Données

- Le fichier de données brutes `GermanCredit.csv` est constitué de 1000 observations et 32 variables.
- Le fichier `GermanCredit_DescriptifVARS.csv` donne un descriptif des différentes variables

TO DO - Phase 1

- Chargement des données brutes
- Typage des variables selon les informations données dans le fichier `GermanCredit_DescriptifVARS.csv`
- Analyses descriptives univariées
- Analyses bivariées croisant la variable à prédire (`RESPONSE`) avec chacune des autres variables.

Point étudié

- 1 Introduction générale
- 2 Utilisation du logiciel R
- 3 Analyse de données avec le logiciel R - Partie 1
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2**
 - Estimation et test d'une corrélation
 - Régression linéaire
 - Analyse de la variance

Point étudié

- 1 Introduction générale
- 2 Utilisation du logiciel R
- 3 Analyse de données avec le logiciel R - Partie 1
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2**
 - **Estimation et test d'une corrélation**
 - Régression linéaire
 - Analyse de la variance

Estimation et test sur une corrélation

2 variables quantitatives X_1 et X_2

- Faire un nuage de points croisant X_1 et X_2 afin d'apprécier la nature de la liaison :
`plot($X_2 \sim X_1$)`
- `cor.test()` : produit le coefficient de corrélation de Pearson, l'intervalle de confiance ainsi que le test de nullité du coefficient de corrélation.

Point étudié

- 1 Introduction générale
- 2 Utilisation du logiciel R
- 3 Analyse de données avec le logiciel R - Partie 1
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 Analyse de données avec le logiciel R - Partie 2**
 - Estimation et test d'une corrélation
 - Régression linéaire**
 - Analyse de la variance

Bref rappel sur la régression linéaire

Objectif : expliquer une variable quantitative Y en fonction de variables explicatives X_1, X_2, \dots, X_p avec le modèle suivant :

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p + \epsilon = \sum_{j=1}^p \beta_j X_j + \epsilon$$

avec $\epsilon \sim \mathcal{N}(0, \sigma^2)$

L'estimation des coefficients β_j est réalisée à partir d'un échantillon par la méthode des moindres carrés ou de maximum de vraisemblance.

L'hypothèse de normalité sur les erreurs permet de réaliser de l'inférence sur les paramètres.

En présence d'une seule variable explicative, on parle de **régression linéaire simple**, sinon de **régression linéaire multiple**.

Régression linéaire

Formulation du modèle dans R :

Variable à expliquer ~ Variable(s) explicative(s)

Exemples :

- $Y \sim X$: une seule variable explicative
- $Y \sim X_1 + X_2 + X_3$: plusieurs variables explicatives
- $Y \sim X_1 + X_2 + X_3 + X_1 : X_2$: ajout du terme d'interaction entre X_1 et X_2
- $Y \sim (X_1 + X_2 + X_3)^2$: toutes les interactions entre paires
- $Y \sim (X_1 * X_2 * X_3)$: toutes les interactions entre paires et à 3 variables

Régression linéaire - Fonctions utiles en R

- `lm()` : ajuste un modèle de régression
- `summary(lm())` : description des résultats du modèle
- `anova(lm())` : décomposition de variance du modèle
- `confint(lm())` : intervalle de confiance pour les paramètres du modèle
- `predict(lm())` : intervalle de confiance pour nouvelle valeur de $Y|X = x_0$
- `step()` : sélection automatisée de variables (voir aussi `add1()` et `drop1()`)

Analyse des résidus et mesures d'influence

- `residuals()` : résidus du modèle
- `fitted()` : valeurs prédites
- `qqnorm()` : graphique Quantiles-Quantiles pour loi Normale
- `cooks.distance()` : distances de Cook

Point étudié

- 1 Introduction générale
- 2 Utilisation du logiciel R
- 3 Analyse de données avec le logiciel R - Partie 1
- 4 Initiation à la production de rapports avec RMarkdown
- 5 Projet - Phase 1
- 6 **Analyse de données avec le logiciel R - Partie 2**
 - Estimation et test d'une corrélation
 - Régression linéaire
 - **Analyse de la variance**

Analyse de la variance

1 facteur : extension à plus de 2 groupes du test de Student

- `aov()` : ajuste une analyse de variance
- `summary(aov())` : description des résultats
- `anova(aov())` : table d'ANOVA
- `bartlett.test()` : test d'égalité des variances
- `pairwise.t.test()` : tests des comparaisons deux-à-deux (voir aussi `TukeyHSD()`)