

# Débuter avec R

Maxime Hervé

UMR 1099 INRA – Agrocampus Ouest – Université Rennes 1 BiO3P

28 avril 2011

# R, c'est quoi ?

Trois caractéristiques importantes :

- multi-plateforme
- gratuit
- libre



# R, c'est quoi ?

Trois caractéristiques importantes :

- multi-plateforme
- gratuit
- libre

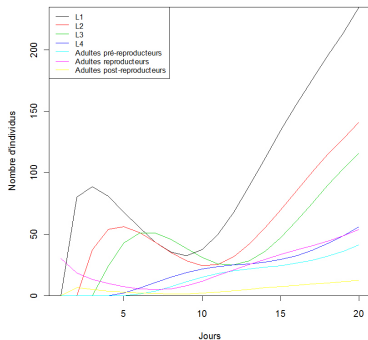
→ N'importe où, n'importe quand !



# R, ça sert à quoi ?

Analyser des données :

- calculs
- tests d'hypothèses
- simulations



## Une table d'ANOVA classique

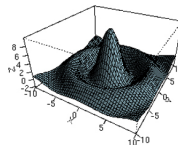
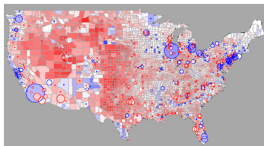
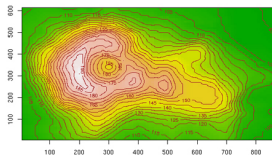
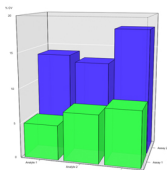
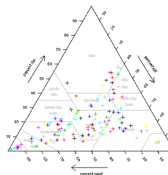
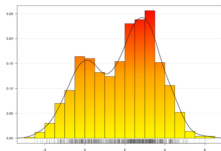
Response: reponse

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
facteur1	1	256.435	256.435	444.92	< 2.2e-16	***
Residuals	38	21.902	0.576			

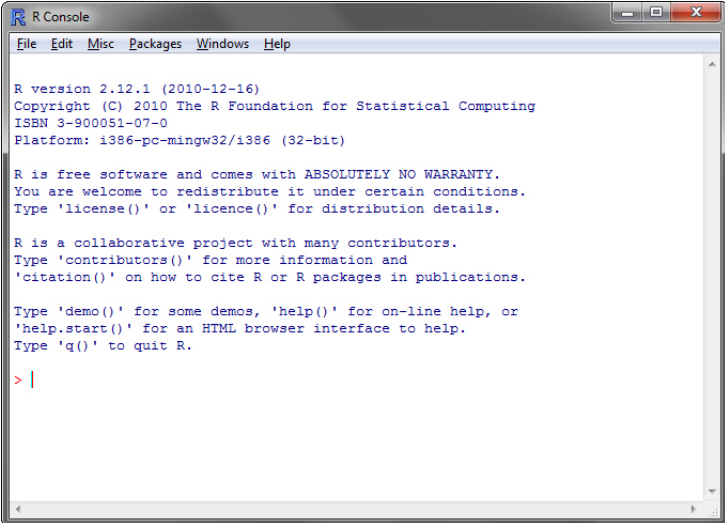
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# R, ça sert à quoi ?

Tracer des graphes :



# R est un langage à la syntaxe intuitive



```
R Console
File Edit Misc Packages Windows Help

R version 2.12.1 (2010-12-16)
Copyright (C) 2010 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: i386-pc-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

# R est un langage à la syntaxe intuitive

## Créer des objets

```
> a <- 5
```

# R est un langage à la syntaxe intuitive

## Créer des objets

```
> a <- 5
```

```
> a
```

```
[1] 5
```



# R est un langage à la syntaxe intuitive

## Créer des objets

```
> a <- 5
> a
[1] 5
> b <- "Bonjour foule en delire"
> b
[1] "Bonjour foule en delire"
```

# R est un langage à la syntaxe intuitive

## Créer des objets

```
> a <- 5
> a
[1] 5
> b <- "Bonjour foule en delire"
> b
[1] "Bonjour foule en delire"
> taille.hommes <- c(171,185,175,169,187,186,175,172,182,184)
> taille.hommes
[1] 171 185 175 169 187 186 175 172 182 184
```

# R est un langage à la syntaxe intuitive

## Créer des objets

```
> a <- 5
> a
[1] 5
> b <- "Bonjour foule en delire"
> b
[1] "Bonjour foule en delire"
> taille.hommes <- c(171,185,175,169,187,186,175,172,182,184)
> taille.hommes
[1] 171 185 175 169 187 186 175 172 182 184
> taille.femmes <- c(160,167,166,167,170,173,161,173,168,165)
```

## Créer des objets

```
> a <- 5
> a
[1] 5
> b <- "Bonjour foule en delire"
> b
[1] "Bonjour foule en delire"
> taille.hommes <- c(171,185,175,169,187,186,175,172,182,184)
> taille.hommes
[1] 171 185 175 169 187 186 175 172 182 184
> taille.femmes <- c(160,167,166,167,170,173,161,173,168,165)
```

De nombreux types de jeu de données peuvent être importés :

- feuilles Excel
- bases de données
- fichiers SAS, SPSS, Stata...
- ...

# R est un langage à la syntaxe intuitive

## Effectuer des opérations

```
> talons <- c(0,5,3,2,0,6,2,2,4,0)
```

# R est un langage à la syntaxe intuitive

## Effectuer des opérations

```
> talons <- c(0,5,3,2,0,6,2,2,4,0)
> taille.femmes - talons
[1] 160 162 163 165 170 167 159 171 164 165
```

# R est un langage à la syntaxe intuitive

## Effectuer des opérations

```
> talons <- c(0,5,3,2,0,6,2,2,4,0)
> taille.femmes - talons
[1] 160 162 163 165 170 167 159 171 164 165
> taille.femmes2 <- taille.femmes - talons
```

## Réaliser des tests statistiques

```
> t.test(taille.hommes,taille.femmes2)
```



## Réaliser des tests statistiques

```
> t.test(taille.hommes,taille.femmes2)
```

```
Welch Two Sample t-test
```

```
data: taille.hommes and taille.femmes2
```

```
t = 5.5886, df = 14.286, p-value = 6.199e-05
```

```
alternative hypothesis: true difference in means is not equal to 0
```

```
95 percent confidence interval:
```

```
8.637141 19.362859
```

```
sample estimates:
```

```
mean of x mean of y
```

```
178.6      164.6
```

## R peut être étendu à l'infini

Si une fonction n'existe pas, on la crée !

*Exemple* : calcul de l'erreur standard d'une moyenne

$$ESM = \frac{\text{écart - type}}{\sqrt{\text{effectif}}}$$

# R peut être étendu à l'infini

Si une fonction n'existe pas, on la crée !

*Exemple* : calcul de l'erreur standard d'une moyenne

$$ESM = \frac{\text{écart - type}}{\sqrt{\text{effectif}}}$$

3 fonctions à utiliser :

- écart - type : `sd()`
- racine carrée : `sqrt()`
- effectif (longueur du vecteur) : `length()`

# R peut être étendu à l'infini

Si une fonction n'existe pas, on la crée !

*Exemple* : calcul de l'erreur standard d'une moyenne

$$ESM = \frac{\text{écart - type}}{\sqrt{\text{effectif}}}$$

3 fonctions à utiliser :

- écart - type : `sd()`
- racine carrée : `sqrt()`
- effectif (longueur du vecteur) : `length()`

## Créer sa propre fonction

```
> err.std <- function(vecteur) { sd(vecteur)/sqrt(length(vecteur)) }
```

# R peut être étendu à l'infini

Si une fonction n'existe pas, on la crée !

*Exemple* : calcul de l'erreur standard d'une moyenne

$$ESM = \frac{\text{écart - type}}{\sqrt{\text{effectif}}}$$

3 fonctions à utiliser :

- écart - type : `sd()`
- racine carrée : `sqrt()`
- effectif (longueur du vecteur) : `length()`

## Créer sa propre fonction

```
> err.std <- function(vecteur) { sd(vecteur)/sqrt(length(vecteur)) }  
> err.std(taille.hommes)  
[1] 2.253256
```

# R peut être étendu à l'infini

Toutes les fonctions sont contenues dans des *packages* :

- certains sont installés avec **R**

# R peut être étendu à l'infini

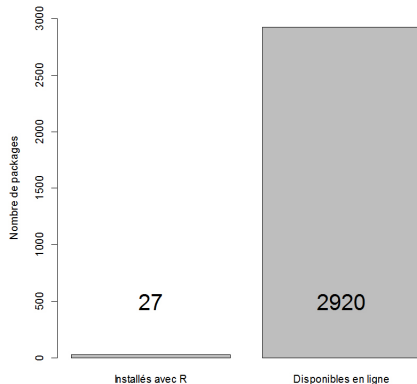
Toutes les fonctions sont contenues dans des *packages* :

- certains sont installés avec **R**
- **R** étant libre, **beaucoup** d'autres sont disponibles en ligne sur le CRAN (*the Comprehensive R Archive Network*)

# R peut être étendu à l'infini

Toutes les fonctions sont contenues dans des *packages* :

- certains sont installés avec **R**
- **R** étant libre, **beaucoup** d'autres sont disponibles en ligne sur le CRAN (*the Comprehensive R Archive Network*)





L'aide fournie avec R :

- manuels

*An Introduction to R*  
*Writing R Extensions*  
*R Data Import/Export*

*The R Language Definition*  
*R Installation and Administration*  
*R Internals*

L'aide fournie avec R :

- manuels

*An Introduction to R*  
*Writing R Extensions*  
*R Data Import/Export*

*The R Language Definition*  
*R Installation and Administration*  
*R Internals*

- aide des fonctions :
  - utilité
  - utilisation
  - exemples
  - mail de l'auteur
  - ...

L'aide en ligne :

- aide des packages

# R ne vous laisse pas seul

L'aide en ligne :

- aide des packages

Utiliser le moteur de recherche intégré à R

```
> RSiteSearch("regression")
```

L'aide en ligne :

- aide des packages

Utiliser le moteur de recherche intégré à R

```
> RSiteSearch("regression")
```

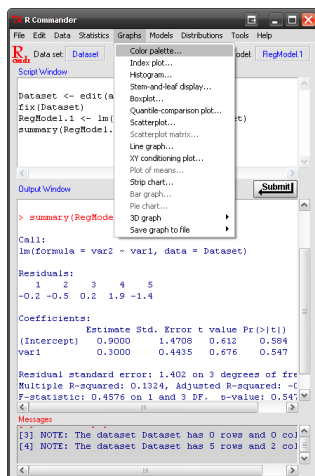
- documents rédigés par des utilisateurs :
  - *R pour les débutants (E. Paradis)*
  - *Lire; Compter; Tester... avec R (C. Genolini)*
  - *R pour les sociologues (J. Barnier)*
  - *R pour les statophobes (D. Poinot)*
  - *Brise Glace - R (A. Robinson & A. Scholesing)*
  - *Aide - mémoire de statistique appliquée à la biologie (M. Hervé)*

# R peut aussi être utilisé *via* une interface graphique

R Commander, JGR, Sci-Views R, Rattle, GrapheR...

Avantages :

- pratique
- rapide
- pédagogique



# R peut aussi être utilisé *via* une interface graphique

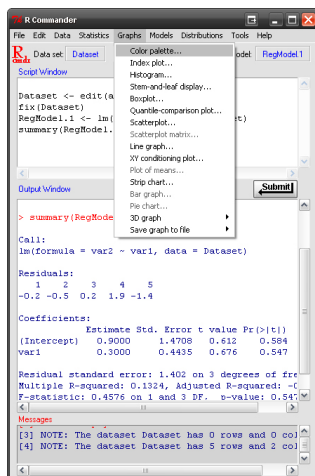
R Commander, JGR, Sci-Views R, Rattle, GrapheR...

Avantages :

- pratique
- rapide
- pédagogique

Inconvénients :

- limité
- évolution lente
- pas de mémoire



## R peut aussi être utilisé *via* une interface graphique

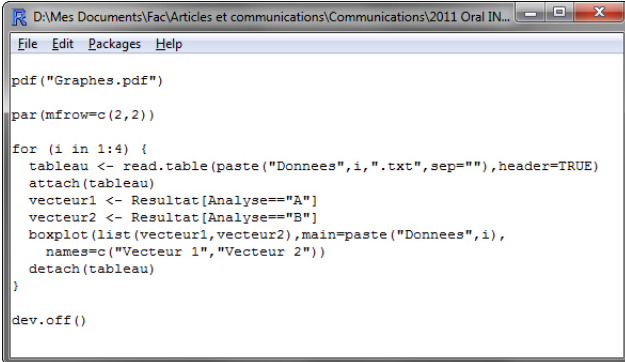
Très peu de lignes de commandes à utiliser → console



## R peut aussi être utilisé *via* une interface graphique

Très peu de lignes de commandes à utiliser → console

Procédure plus longue, répétitive, structurée... → éditeur de script



```
D:\Mes Documents\Fac\Articles et communications\Communications\2011 Oral IN...
File Edit Packages Help

pdf("Graphes.pdf")

par(mfrow=c(2,2))

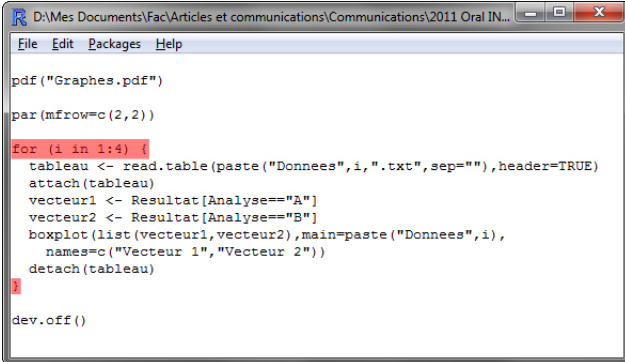
for (i in 1:4) {
  tableau <- read.table(paste("Donnees",i,".txt",sep=""),header=TRUE)
  attach(tableau)
  vecteur1 <- Resultat[Analyse=="A"]
  vecteur2 <- Resultat[Analyse=="B"]
  boxplot(list(vecteur1,vecteur2),main=paste("Donnees",i),
    names=c("Vecteur 1","Vecteur 2"))
  detach(tableau)
}

dev.off()
```

## R peut aussi être utilisé *via* une interface graphique

Très peu de lignes de commandes à utiliser → console

Procédure plus longue, répétitive, structurée... → éditeur de script



```
R D:\Mes Documents\Fac\Articles et communications\Communications\2011 Oral IN...
File Edit Packages Help

pdf("Graphes.pdf")

par(mfrow=c(2,2))

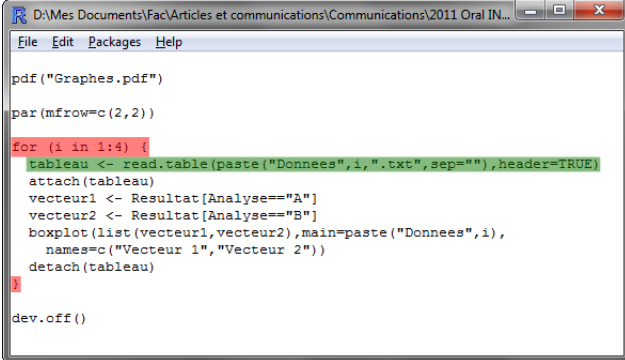
for (i in 1:4) {
  tableau <- read.table(paste("Donnees",i,".txt",sep=""),header=TRUE)
  attach(tableau)
  vecteur1 <- Resultat[Analyse=="A"]
  vecteur2 <- Resultat[Analyse=="B"]
  boxplot(list(vecteur1,vecteur2),main=paste("Donnees",i),
    names=c("Vecteur 1","Vecteur 2"))
  detach(tableau)
}

dev.off()
```

## R peut aussi être utilisé *via* une interface graphique

Très peu de lignes de commandes à utiliser → console

Procédure plus longue, répétitive, structurée... → éditeur de script



```
D:\Mes Documents\Fac\Articles et communications\Communications\2011 Oral IN...
File Edit Packages Help

pdf("Graphes.pdf")

par(mfrow=c(2,2))

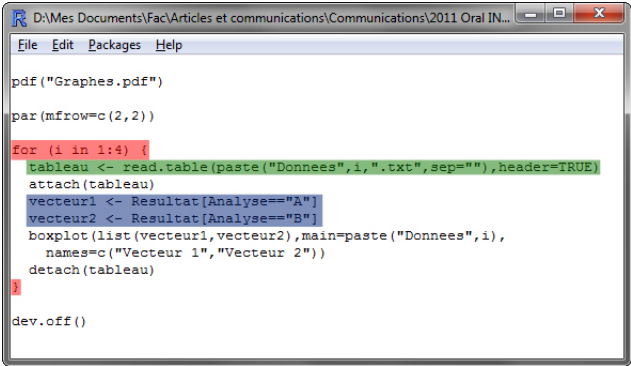
for (i in 1:4) {
  tableau <- read.table(paste("Donnees",i,".txt",sep=""),header=TRUE)
  attach(tableau)
  vecteur1 <- Resultat[Analyse=="A"]
  vecteur2 <- Resultat[Analyse=="B"]
  boxplot(list(vecteur1,vecteur2),main=paste("Donnees",i),
    names=c("Vecteur 1","Vecteur 2"))
  detach(tableau)
}

dev.off()
```

# R peut aussi être utilisé *via* une interface graphique

Très peu de lignes de commandes à utiliser → console

Procédure plus longue, répétitive, structurée... → éditeur de script



```
D:\Mes Documents\Fac\Articles et communications\Communications\2011 Oral IN...
File Edit Packages Help

pdf("Graphes.pdf")

par(mfrow=c(2,2))

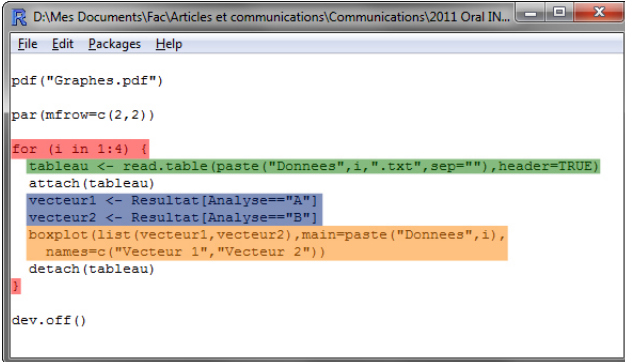
for (i in 1:4) {
  tableau <- read.table(paste("Donnees",i,".txt",sep=""),header=TRUE)
  attach(tableau)
  vecteur1 <- Resultat[Analyse=="A"]
  vecteur2 <- Resultat[Analyse=="B"]
  boxplot(list(vecteur1,vecteur2),main=paste("Donnees",i),
    names=c("Vecteur 1","Vecteur 2"))
  detach(tableau)
}

dev.off()
```

## R peut aussi être utilisé *via* une interface graphique

Très peu de lignes de commandes à utiliser → console

Procédure plus longue, répétitive, structurée... → éditeur de script



```
D:\Mes Documents\Fac\Articles et communications\Communications\2011 Oral IN...
File Edit Packages Help

pdf("Graphes.pdf")

par(mfrow=c(2,2))

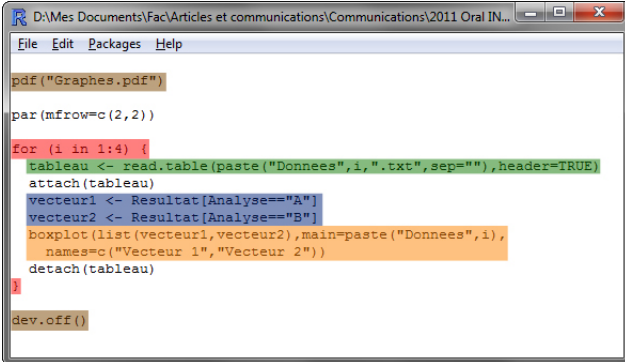
for (i in 1:4) {
  tableau <- read.table(paste("Donnees",i,".txt",sep=""),header=TRUE)
  attach(tableau)
  vecteur1 <- Resultat[Analyse=="A"]
  vecteur2 <- Resultat[Analyse=="B"]
  boxplot(list(vecteur1,vecteur2),main=paste("Donnees",i),
            names=c("Vecteur 1","Vecteur 2"))
  detach(tableau)
}

dev.off()
```

## R peut aussi être utilisé *via* une interface graphique

Très peu de lignes de commandes à utiliser → console

Procédure plus longue, répétitive, structurée... → éditeur de script



```
D:\Mes Documents\Fac\Articles et communications\Communications\2011 Oral IN...
File Edit Packages Help

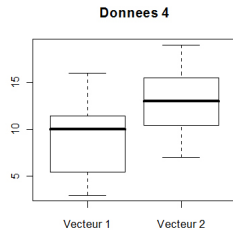
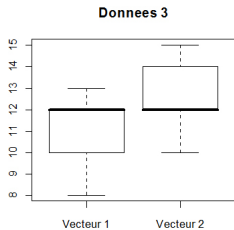
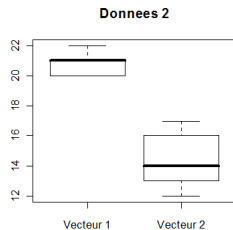
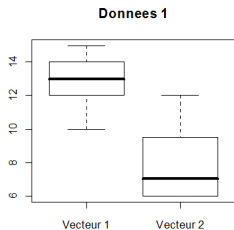
pdf("Graphes.pdf")

par(mfrow=c(2,2))

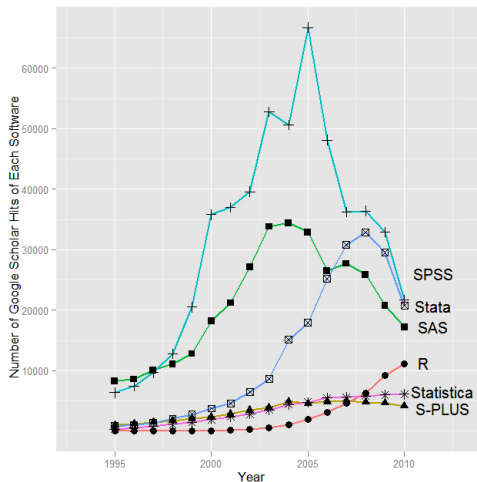
for (i in 1:4) {
  tableau <- read.table(paste("Donnees",i,".txt",sep=""),header=TRUE)
  attach(tableau)
  vecteur1 <- Resultat[Analyse=="A"]
  vecteur2 <- Resultat[Analyse=="B"]
  boxplot(list(vecteur1,vecteur2),main=paste("Donnees",i),
            names=c("Vecteur 1","Vecteur 2"))
  detach(tableau)
}

dev.off()
```

# R peut aussi être utilisé *via* une interface graphique



# R est réellement de plus en plus utilisé





Merci de votre attention