

# L'enseignement et la formation au logiciel R : un retour d'expérience

Julien Barnier  
Centre Max Weber  
`julien.barnier@ens-lyon.fr`  
Twitter : @lapply

19 novembre 2015

# Présentation

# Présentation

Utilisateur (irrégulier) de R depuis une quinzaine d'années.

Initiateur du document *Introduction à R* et du package `questionr`.

Donne des formations à R depuis 7 ans. Public de chercheurs et de doctorants.

## Type de formation

# Formations



# Formations



# Formations

Plan d'une formation type de 2 jours :

# Formations

Plan d'une formation type de 2 jours :

- ▶ **Première demi-journée** : présentation de l'interface.  
Familiarisation avec la ligne de commande de manière très progressive, puis utilisation de scripts. Début de travail avec des données.

# Formations

Plan d'une formation type de 2 jours :

- ▶ **Première demi-journée** : présentation de l'interface.  
Familiarisation avec la ligne de commande de manière très progressive, puis utilisation de scripts. Début de travail avec des données.
- ▶ **Deuxième demi-journée** : poursuite du travail sur des données, première analyses statistiques simples.

# Formations

Plan d'une formation type de 2 jours :

- ▶ **Première demi-journée** : présentation de l'interface.  
Familiarisation avec la ligne de commande de manière très progressive, puis utilisation de scripts. Début de travail avec des données.
- ▶ **Deuxième demi-journée** : poursuite du travail sur des données, première analyses statistiques simples.
- ▶ **Troisième demi-journée** : notion d'indexation et de recodage.

# Formations

Plan d'une formation type de 2 jours :

- ▶ **Première demi-journée** : présentation de l'interface.  
Familiarisation avec la ligne de commande de manière très progressive, puis utilisation de scripts. Début de travail avec des données.
- ▶ **Deuxième demi-journée** : poursuite du travail sur des données, première analyses statistiques simples.
- ▶ **Troisième demi-journée** : notion d'indexation et de recodage.
- ▶ **Quatrième demi-journée** : Approche “marketing” pour donner envie après la demi-journée précédente souvent difficile : cartographie, graphiques, rmarkdown, TraMineR...

Quel R ?

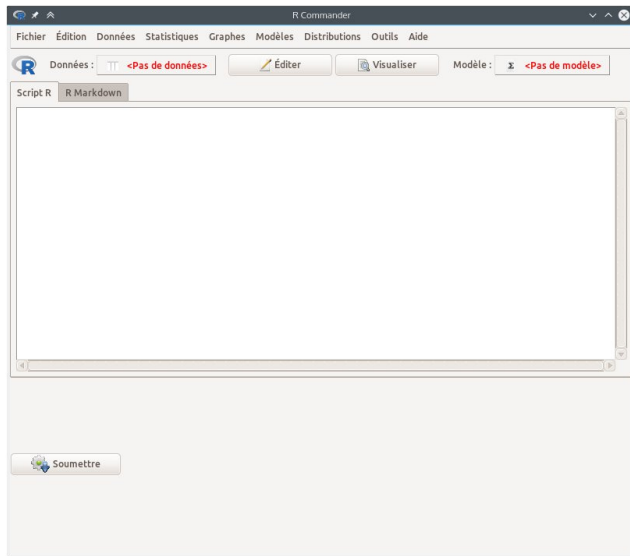
The screenshot displays the RStudio interface with three main panels:

- Source Editor (Left):** Contains an R script for a presentation. The script includes a title slide, a user introduction, a pause, a list of formation types, and a plan for a 2-day formation. It also contains comments about the presentation structure.
- Console (Right):** Shows the output of the R script. It indicates that the file 'slides.Rmd' is being processed, and it lists the contents of the presentation, including the title slide and the list of formation types.
- File Explorer (Bottom Right):** Displays the project structure, showing files such as '.Rhistory', 'ined\_ehess\_2015.Rproj', 'notes\_interv.Rmd', 'slides.Rmd', 'slides.pdf', 'slides.html', 'custom.css', 'img', and 'slides\_elisabeth.Rmd'.

Incontournable aujourd'hui :

- ▶ notion de projets (permet d'éviter les problèmes de répertoires de travail)
- ▶ interface unifiée quel que soit le système d'exploitation
- ▶ séparation de l'écran en quatre volets qui aide à comprendre la logique de R
- ▶ coloration syntaxique, auto-complétion, aide intégrée. . .

# R Commander



# R Commander

Dès le départ, choix de ne pas utiliser *R Commander*.

- ▶ pas convaincu par l'interface
- ▶ trop vite limité
- ▶ on perd en grande partie l'avantage de créer des scripts :  
efficacité, reproductibilité

*Point de vue* : *R Commander* est adapté si on sait dès le départ qu'on pourra se contenter des fonctionnalités proposées par l'interface.

# Interfaces ponctuelles

Aujourd'hui plutôt une démarche “mixte” :

- ▶ On privilégie le code : commandes et scripts dès le départ.
- ▶ En complément ou en support, utilisation ponctuelle d'interfaces graphiques appelées depuis R.
- ▶ Principe de base : ces interfaces ne doivent pas exécuter de commande ou modifier les données.

# Interfaces ponctuelles

Aujourd'hui plutôt une démarche "mixte" :

- ▶ On privilégie le code : commandes et scripts dès le départ.
- ▶ En complément ou en support, utilisation ponctuelle d'interfaces graphiques appelées depuis R.
- ▶ Principe de base : ces interfaces ne doivent pas exécuter de commande ou modifier les données.

Exemples :

- ▶ `irec`, `iorder`, `icut` du package `questionr`
- ▶ `imca` (package en cours de développement)

`shiny` et `htmlwidgets` facilitent le développement de ces interfaces.

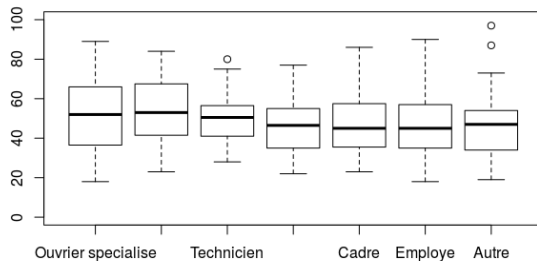
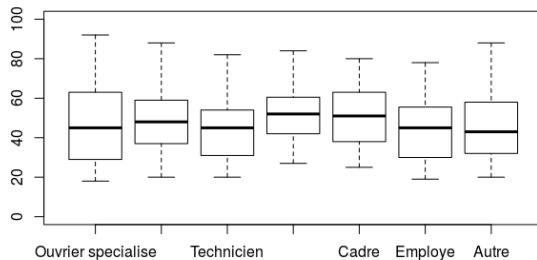
# Évolution du code

Hier :

```
tmp.h <- hdv2003[hdv2003$sexe == "Homme",]  
tmp.f <- hdv2003[hdv2003$sexe == "Femme",]  
par(mfrow=c(2,1), mar=c(3,3,1,1))  
boxplot(age ~ qualif, data=tmp.h, ylim=c(0,100))  
boxplot(age ~ qualif, data=tmp.f, ylim=c(0,100))
```

# Évolution du code

Hier :



# Évolution du code

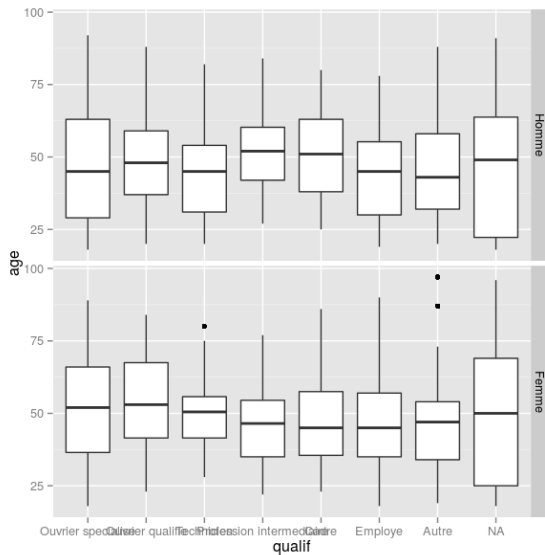
Aujourd'hui :

```
library(ggplot2)

ggplot(data=hdv2003) +
  geom_boxplot(aes(x = qualif, y = age)) +
  facet_grid(sexe ~ .)
```

# Évolution du code

Aujourd'hui :



# Évolution du code

Hier :

```
tmp <- hdv2003[hdv2003$qualif == "Employe", ]  
tab1 <- aggregate(tmp$age, list(tmp$sexe), mean)  
names(tab1) <- c("sexe", "age_mean")  
tab2 <- aggregate(tmp$age, list(tmp$sexe), median)  
names(tab2) <- c("sexe", "age_median")  
merge(tab1, tab2)
```

```
##      sexe age_mean age_median  
## 1 Femme  47.05221         45  
## 2 Homme  44.69792         45
```

# Évolution du code

Aujourd'hui :

```
library(dplyr)

hdv2003 %>%
  filter(qualif == "Employe") %>%
  group_by(sexe) %>%
  summarise(age_mean = mean(age),
            age_median = median(age))
```

```
## Source: local data frame [2 x 3]
##
##      sexe age_mean age_median
##   (fctr)   (dbl)      (dbl)
## 1  Homme 44.69792      45
## 2  Femme 47.05221      45
```

# Quels packages en formation ?

Jusque là, la formation se contentait de R “de base” et du package `questionr` pour des fonctions facilitant des besoins courants.

Aujourd’hui, et de plus en plus, certains packages semblent incontournables :

- ▶ `ggplot2`
- ▶ `dplyr`, `tidyr`
- ▶ `rmarkdown`
- ▶ ...

# Quels packages en formation ?

*Avantages* : fonctionnalités supplémentaires, syntaxe simplifiée et cohérente.

- ▶ affichage de légendes dans des graphiques
- ▶ code avec des *pipes*, plus lisible
- ▶ documents en Markdown et export multi-formats

*Inconvénients* : complexité supplémentaire, logiques de syntaxe et de code différentes.

*Biais* : difficile de ne pas les introduire quand on les utilise au quotidien et que les réponses aux questions posées nous viennent spontanément avec ces packages.

# Quels packages en formation ?

*Question* : Quand et comment les introduire ?

Pas encore de réponse claire.

- ▶ `ggplot2` : dans un deuxième temps.
- ▶ `dplyr`, `tidyr` : pas intégrés aujourd'hui, réflexion en cours.
- ▶ `rmarkdown` : présentation en “bonus”.

# Accompagnement

# Outils et documentation

## Francophones :

- ▶ La documentation en français d'introduction est de plus en plus complète et abondante.
- ▶ Listes de discussion (`R-soc`, `semin-r...`) : peu utilisées.
- ▶ Forum du Cirad : utilisé, mais pas forcément très connu.
- ▶ MOOC

## Anglophones :

- ▶ *StackOverflow*
- ▶ plein d'autres !

# Groupes d'utilisateurs

- ▶ Exemple du groupe d'utilisateurs de R sur Lyon
- ▶ Exemple de l'*Espace quanti* au Centre Max Weber

Merci de votre attention !

# Ressources utiles

- ▶ Document *Introduction à R* :  
<http://alea.fr.eu.org/pages/intro-R>
- ▶ Package `questionr` :  
<http://alea.fr.eu.org/pages/questionr>
- ▶ Site de `ggplot2` : <http://ggplot2.org/>
- ▶ Package `dplyr` :  
<https://cran.r-project.org/web/packages/dplyr>
- ▶ Package `imva` et fonction `imca` (instable, en cours de développement) : <https://github.com/juba/imva>
- ▶ Package associé `scatterD3` permettant des nuages de points interactifs : <https://github.com/juba/scatterD3>