

Documenter avec Roxygen

La documentation roxygen a deux fonctions :

1. Elle vous aide à structurer la documentation de vos fonctions et de vos jeux de données
2. Elle permet de générer automatiquement les différents supports de documentation fournis avec les packages R

Générer les fichiers Rd

- ▶ Les fichiers Rd sont dans un format simple dérivé de Latex
- ▶ Lorsque vous utilisez `?X`, `help("x")` ou `example("x")` R recherche un fichier Rd contenant `\alias {x}`. Il analyse ensuite le fichier, le convertit en html et l'affiche.
- ▶ Ils font partie de la documentation des packages R

- ▶ R
 - ▶ functions.r
- ▶ DESCRIPTION
- ▶ tests
 - ▶ testthat
 - ▶ test-exemple.r
 - ▶ testthat.R
- ▶ man
 - ▶ **add.Rd**
- ▶ vignettes
 - ▶ formationR.Rnw
- ▶ data
- ▶ NAMESPACE

Le processus Roxygen

Ce processus se décompose en trois étapes:

1. Ajouter des commentaires suivant la syntaxe roxygen aux fichiers source.
2. `roxygen2::roxygenise()` convertit les commentaires roxygen en fichiers `.Rd`.
3. R convertit les fichiers `.Rd` en documentaion lisible par chacune et chacun.

NB : Les commentaires roxygen commencent par `#'`, et les autres commentaires continuent à être valides.

Un exemple

```
#' Add together two numbers  
#'  
#' @param x A number  
#' @param y A number  
#' @return The sum of \code{x} and \code{y}  
#' @examples  
#' add(1, 1)  
#' add(10, 1)  
add <- function(x, y) {  
  x + y  
}
```

Éléments de base

- ▶ Les commentaires de Roxygen commencent par `# '` et incluent des tags de type `@tag return`.
- ▶ Les balises (tags) séparent la documentation en morceaux et le contenu d'une balise s'étend de la fin du nom de balise au début de la balise suivante (ou à la fin du bloc).
- ▶ Parce que `@` a une signification spéciale dans roxygen, vous devez écrire `@@` pour ajouter un vrai `@`

Éléments de base

- ▶ Chaque bloc de documentation commence par un texte. La première phrase devient le titre de la documentation. (voir `help()`)
 - ▶ Il doit tenir sur une ligne, être écrit dans le cas de la phrase, et terminer en un arrêt complet.
- ▶ Le deuxième paragraphe contient la description abrégée
- ▶ Le troisième paragraphe contient la description détaillée (optionnelle)
- ▶ NB : `\ Code {}` et `\ link {}` sont des commandes de formatage .Rd
- ▶ NB2 : vos commentaires doivent faire moins de ~ 80 colonnes de large.

On peut aussi, pour ces trois composantes, utiliser des tags explicites :

```
#' @title Sum of vector elements.  
#'  
#' @description  
#' \code{sum} returns the sum of all the values present  
#'  
#' @details  
#' This is a generic function: methods can be defined  
#' or via the \code{\link{Summary}} group generic. For  
#' the arguments \code{...} should be unnamed, and di  
#' first argument.  
sum <- function(..., na.rm = TRUE) {}
```

Améliorer la navigation

- ▶ @ Seealso vous permet de pointer vers d'autres ressources utiles, par soit sur le web\ url {http://www.r-project.org}, soit vers d'autres éléments du code avec ' \code{\link{functionname}}
- ▶ Pour une famille de fonctions associées : @family <family> pour ajouter automatiquement les listes et interconnexions appropriées à la section @ seealso.

```
#' @family aggregate functions  
#'  
#'  
#'  
#'  
#'  
#'
```

Documenter une fonction

- ▶ `@param name description`. Tous les paramètres doivent être documentés! Deux par deux, ça marche aussi :
`@param x,y Numeric vectors`.
- ▶ `@examples` . NB : dans le cas de packages, cette section est exécutée automatiquement (voir R CMD check).
- ▶ `@return description` Décrit la sortie de la fonction.

Autres objets à documenter :

- ▶ Jeux de données
- ▶ Classes S3, S4

Plus d'infos sur: <https://cran.r-project.org/web/packages/roxygen2/vignettes/roxygen2.html>

Des questions?

