



CRÉATION DE PACKAGE R

Camille COCHENER

Florian KROELL

Romain MOREAU

Lundi 12 octobre 2015

Qu'est ce qu'un package ?

- Ensemble de programmes R
- Code, données, documentation, tests
- Pour des méthodes particulières/domaines d'applications

Exemple : FactoMineR

Nom	Modifié le	Type	Taille
data	03/06/2015 11:00	Dossier de fichiers	
doc	03/06/2015 11:00	Dossier de fichiers	
help	03/06/2015 11:00	Dossier de fichiers	
html	03/06/2015 11:00	Dossier de fichiers	
Meta	03/06/2015 11:00	Dossier de fichiers	
R	03/06/2015 11:00	Dossier de fichiers	
DESCRIPTION	03/06/2015 11:00	Fichier	1 Ko
INDEX	03/06/2015 11:00	Fichier	6 Ko
MD5	03/06/2015 11:00	Fichier	2 Ko
NAMESPACE	03/06/2015 11:00	Fichier	2 Ko

Fonctions : HCPC, MCA, MFA...

Un package est un ensemble de fichiers qu'il faut générer

Comment créer un package ?

Il faut créer la structure du package

- *Créer l'arborescence de fichiers*

Il faut créer un programme

- *Ecriture des fonctions*

Il faut remplir les fichiers du package

- Namespace
- Description

Il faut écrire la documentation

- *Ecriture de l'aide*

Avant toutes choses...

- Installation des packages nécessaires
devtools, roxygen2, testthat, knitr
- S'assurer que la version de Rstudio est la plus récente
- Télécharger **Rtools** compilateur de C
- **devtools::install_github("hadley/devtools")** met à jour devtools

On peut continuer...

Comment créer un package ?

- Plusieurs méthodes...

↳ `package.skeleton("nom package", code_file= "prog.R")`

Créer plus d'éléments que nécessaires

↳ en utilisant Rstudio

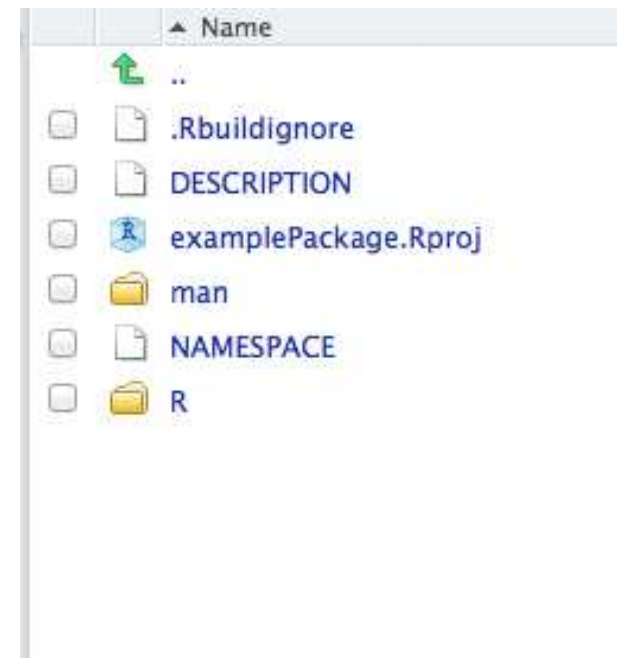
Plus simple

3 fichiers :

Répertoire R (code),

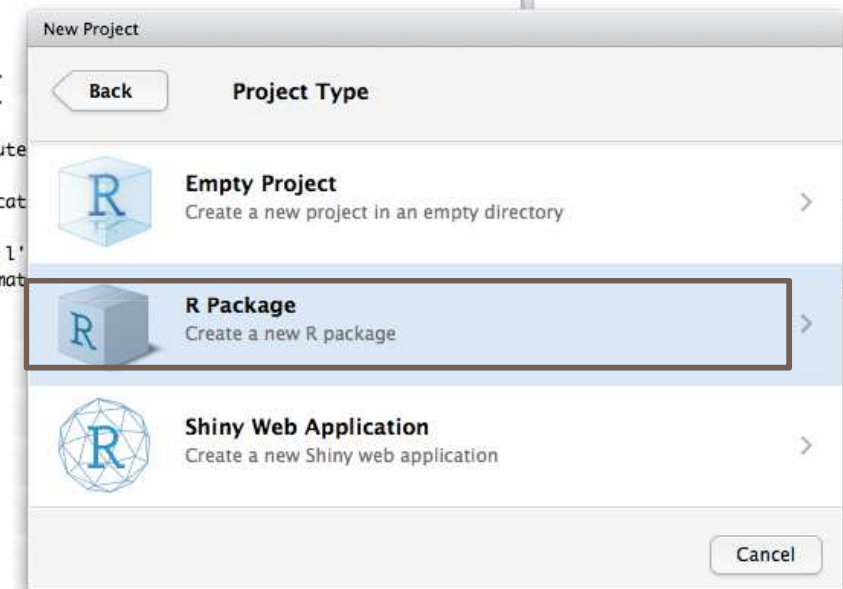
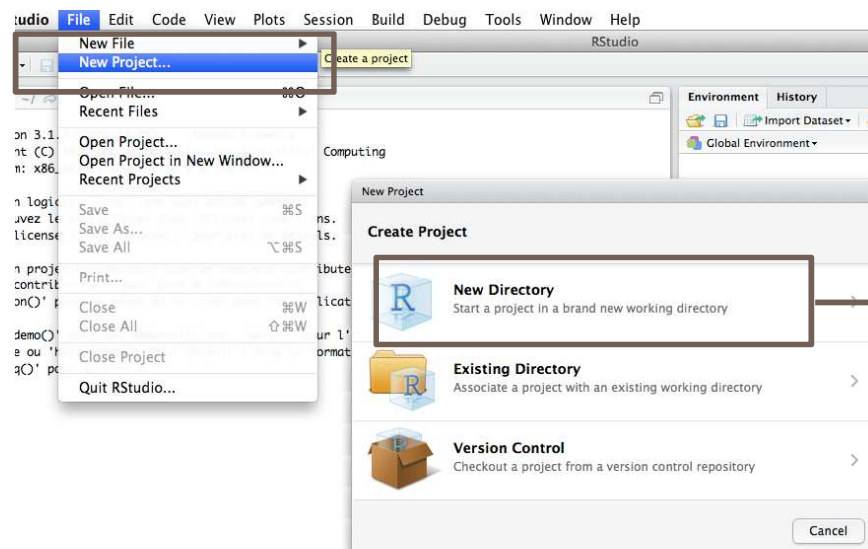
Fichier description et

fichier namespace



Deux façons de créer le package dans Rstudio

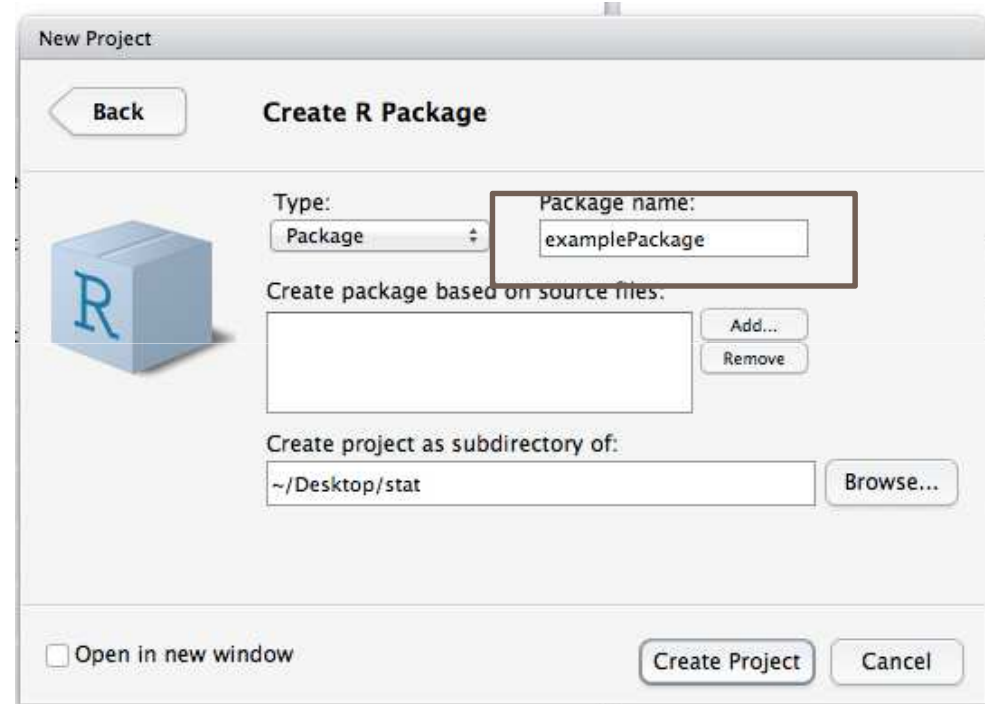
- Avec l'onglet



- Avec le package *devtools* en utilisant la fonction *create()*

Choix du nom du package

- Lettres, nombres, points
- Doit commencer par une lettre
- Ne doit pas finir par un point
- Doit être le plus explicite possible



Dans notre exemple, le package créé s'appelle
examplePackage

Création des fonctions

- Ecriture du script
- Ne pas mettre toutes les fonctions dans un seul script
- Ne pas mettre une seule fonction par script

```
statFisher <- function(mod.anova, col.dd1, col.sq, row.fact, base.comp) {  
  if (class(col.dd1) != "numeric" || class(col.sq) != "numeric") {  
    stop("Please, enter integers for arguments col.dd1 and col.sq.")  
  }  
  if (class(row.fact) != "numeric" || class(base.comp) != "numeric") {  
    stop("Please, enter integers for arguments row.fact and base.comp.")  
  }  
  if (!(row.fact %in% 1:length(mod.anova[[1]])) || sum(base.comp %in% 1:length(mod.anova[[1]]  
    length(base.comp))) {  
    stop("row.fact or elements of base.comp are out of bounds.")  
  }  
  CMfact <- mod.anova[[col.sq]][[row.fact]]/mod.anova[[col.dd1]][[row.fact]]  
  CMcomp <- 0  
  if (length(base.comp) == 1) {  
    CMcomp <- mod.anova[[col.sq]][[base.comp]]/mod.anova[[col.dd1]][[base.comp]]  
    F.fact <- CMfact/CMcomp  
    dd1Denom <- mod.anova[[col.dd1]][[base.comp]]  
  } else {  
    rows.comp <- base.comp  
    CMdenom <- rep(0, length(base.comp))  
    frac <- rep(0, length(base.comp))  
    for (i in 1:length(base.comp)) {  
      if (base.comp[i] < 0) {  
        rows.comp[i] <- -base.comp[i]  
      }  
    }  
    for (i in 1:length(rows.comp)) {
```

Il faut regrouper les fonctions par catégories

Outils d'aide au développement des scripts

- Les raccourcis :
 - **Ctrl/Cmd + Shift + L** équivaut à **devtools::load.all()**
 - **F2** et **Ctrl + .** pour la navigation entre fonctions
- **formatR::tidy_dir()**
- **lintr::lint_package()**

Jeu de données du package

- 3 manières d'insérer le jeu de données dans un package

Pour rendre le jeu de données disponible à l'utilisateur → Dossier DATA

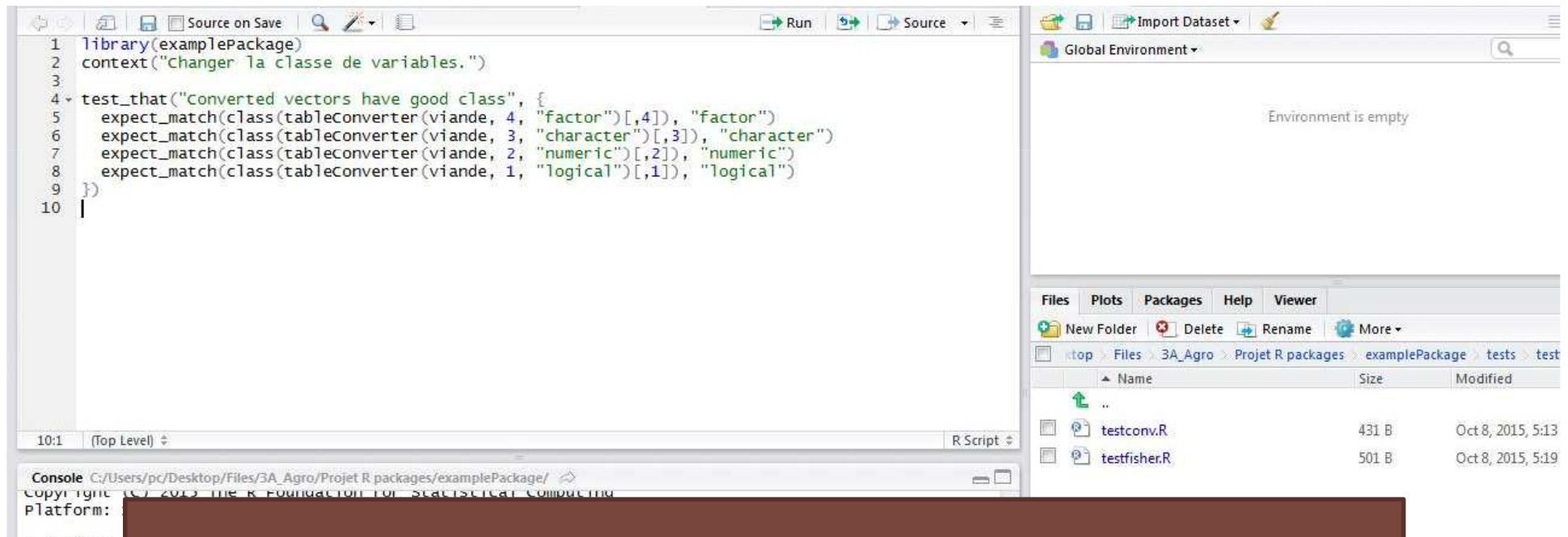
Non disponible à l'utilisateur mais nécessaire pour les fonctions → Dossier R

Jeu de données bruts → Dossier INST

- Les jeux de données sont sous format rda
- Utilisation de `devtools::use_data()`

Tests

- Partie vitale de la création du package
- **devtools::use_testthat()** → créer le dossier tests et met à l'intérieur un fichier **testthat.R** et un dossier **testthat**



The screenshot shows an R IDE with a script editor on the left and a file explorer on the right. The script editor contains the following R code:

```
1 library(examplePackage)
2 context("changer la classe de variables.")
3
4 test_that("Converted vectors have good class", {
5   expect_match(class(tableconverter(viande, 4, "factor"))[,4]), "factor")
6   expect_match(class(tableconverter(viande, 3, "character"))[,3]), "character")
7   expect_match(class(tableconverter(viande, 2, "numeric"))[,2]), "numeric")
8   expect_match(class(tableconverter(viande, 1, "logical"))[,1]), "logical")
9 })
10 |
```

The file explorer on the right shows the following file structure:

- Files
- Plots
- Packages
- Help
- Viewer

Name	Size	Modified
..		
testconv.R	431 B	Oct 8, 2015, 5:13
testfisher.R	501 B	Oct 8, 2015, 5:19

The console at the bottom shows the following output:

```
C:/Users/pc/Desktop/Files/3A_Agro/Projet R packages/examplePackage/
Copyright (C) 2015 THE R FOUNDATION FOR STATISTICAL COMPUTING
Platform:
```

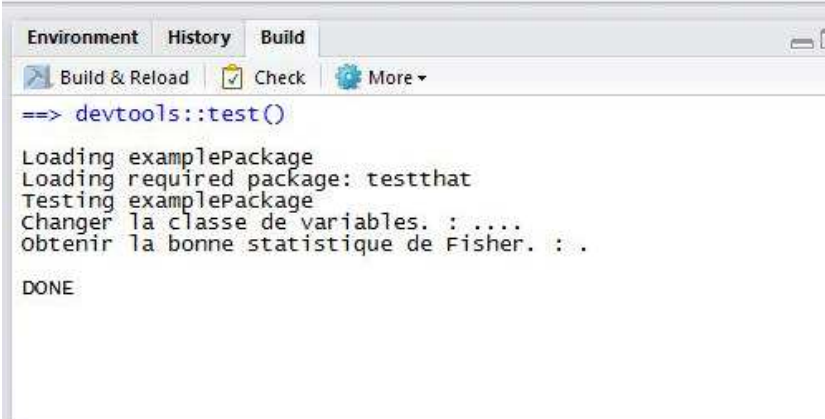
Les tests permettent de faire marcher les fonctions sur un exemple

Tests

- Le raccourci :

Ctrl/Cmd + Shift + T ou **devtools::test()**

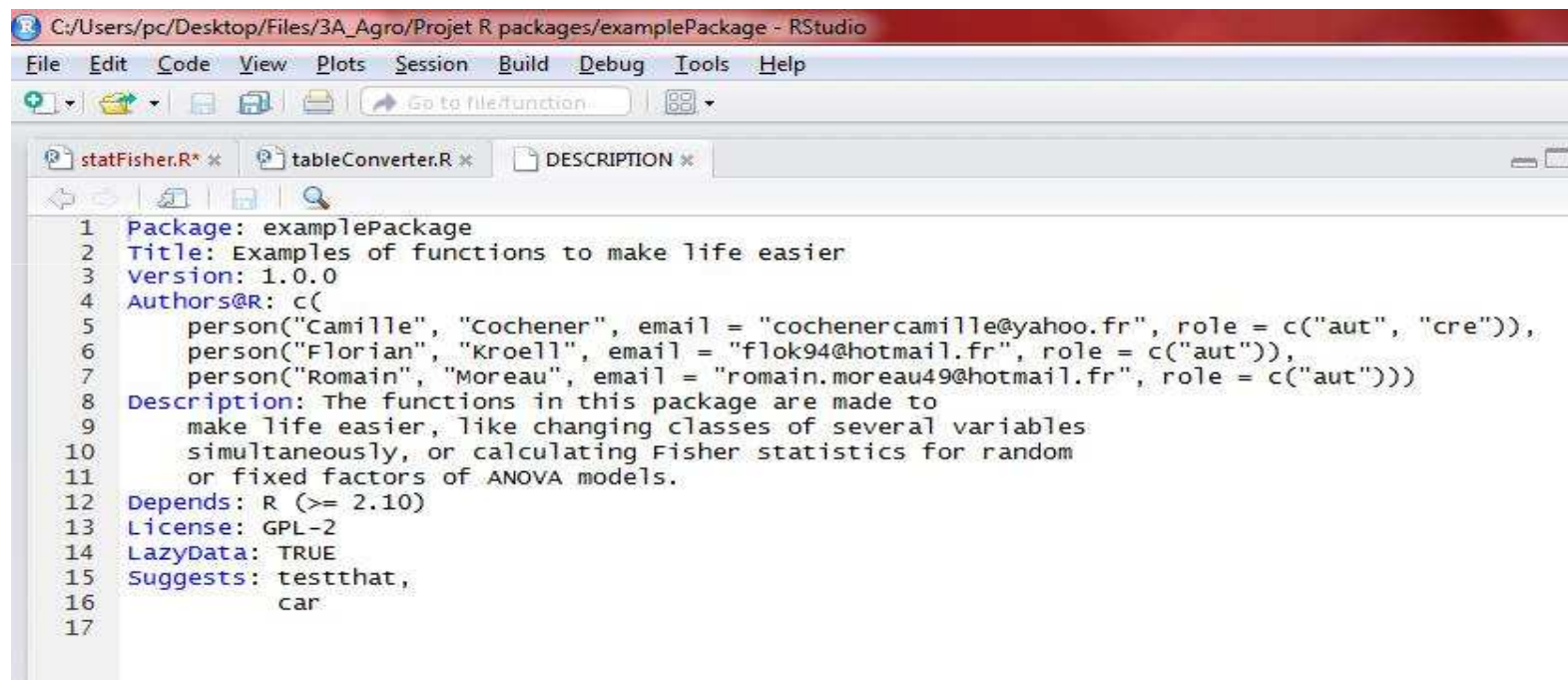
- **expect_equal()**
- **expect_warning()**
- ...



```
Environment History Build
Build & Reload Check More
==> devtools::test()
Loading examplePackage
Loading required package: testthat
Testing examplePackage
Changer la classe de variables. :.....
obtenir la bonne statistique de Fisher. :.
DONE
```

Description (package metadata)

- Dans notre exemple...



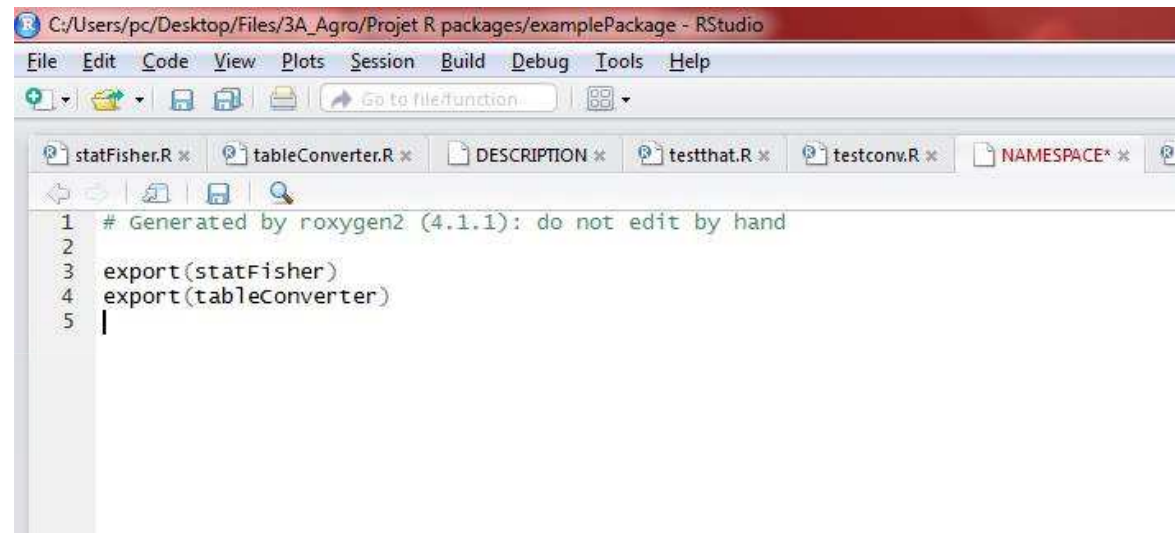
```
1 Package: examplePackage
2 Title: Examples of functions to make life easier
3 Version: 1.0.0
4 Authors@R: c(
5   person("Camille", "Cochener", email = "cochenercamille@yahoo.fr", role = c("aut", "cre")),
6   person("Florian", "Kroell", email = "flok94@hotmail.fr", role = c("aut")),
7   person("Romain", "Moreau", email = "romain.moreau49@hotmail.fr", role = c("aut")))
8 Description: The functions in this package are made to
9   make life easier, like changing classes of several variables
10  simultaneously, or calculating Fisher statistics for random
11  or fixed factors of ANOVA models.
12 Depends: R (>= 2.10)
13 License: GPL-2
14 LazyData: TRUE
15 Suggests: testthat,
16   car
17
```

Remplissage « à la main »
Description générale du package

Namespace

- `export()`
- `exportPattern()`
- `exportClasses()`
- `S3method()`

- `import()`
- `importFrom()`
- `importClassesFrom()`
- `useDynLib()`



```
C:/Users/pc/Desktop/Files/3A_Agro/Projet R packages/examplePackage - RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
statFisher.R x tableConverter.R x DESCRIPTION x testthat.R x testconv.R x NAMESPACE* x
1 # Generated by roxygen2 (4.1.1): do not edit by hand
2
3 export(statFisher)
4 export(tableConverter)
5 |
```

Création par Roxygen

- ✓ Ordinateur configuré
- ✓ Programme qui tourne
- ✓ Arborescence créée
- ✓ Fichiers DESCRIPTION et NAMESPACE remplis

...

Il manque la **DOCUMENTATION**

Documentation (étape longue)

Documentation des fonctions

The screenshot displays the R IDE interface. On the left, the source code for the function `statFisher` is shown in an R script editor. The code includes comments for documentation, such as `@param`, `@return`, `@export`, and `@examples`. The function itself calculates the Fisher's test statistic and returns a numeric list. The right pane shows the generated R documentation file `statFisher.Rd`, which is rendered in a structured format with sections for **Description**, **Usage**, **Arguments**, **Value**, and **Examples**. The console at the bottom shows the execution of the function and the generation of the documentation file.

```
1 #' statFisher
2 #' Calcule la statistique de Fisher a partir d'une base de comparaison donnee.
3 #'
4 #'
5 #' @param mod.anova : data.frame. Modele anova issu de la fonction anova() ou Anova()
6 #' @param col.dd1 : numeric. Indice de la colonne des degres de liberte
7 #' @param col.sq : numeric. Indice de la colonne des somme des carres des ecarts
8 #' @param row.fact : numeric. Indice de la ligne du facteur a tester
9 #' @param base.comp : numeric vector. Indice(s) de(s) ligne(s) des facteurs de la base de compa
10 #'
11 #' @return numeric list. Statistique de Fisher et les 2 ddl de la loi de fisher associee au tes
12 #'
13 #' @export
14 #'
15 #' @examples
16 #' # Creation du modele lineaire
17 #'
18 #' library(car)
19 #'
20 #' mod <- lm(Note ~ Fournisseur + Seance + Consommateur +
21 #'           Fournisseur:Seance + Fournisseur:Consommateur + Seance:Consommateur,
22 #'           data = viande)
23 #' mod.anova <- Anova(mod, type="III")
24 #'
25 #' # Effet du facteur Consommateur
26 #'
27 #' F.conso <- statFisher(mod.anova, col.dd1=2, col.sq=1, row.fact=4, base.comp=7)
28 #' 1-pf(F.conso[[1]], F.conso[[2]], F.conso[[3]])
29 #'
30 #' # Effet du facteur Fournisseur
31 #'
32 #' F.fourni <- statFisher(mod.anova, col.dd1=2, col.sq=1, row.fact=2, base.comp=c(5,6,8))
33 #' 1-pf(F.fourni[[1]], F.fourni[[2]], F.fourni[[3]])
34 #'
35 #'
```

statFisher (examplePackage) R Documentation

Description

Calcule la statistique de Fisher a partir d'une base de comparaison donnee.

Usage

```
statFisher(mod.anova, col.dd1, col.sq, row.fact, base.comp)
```

Arguments

`mod.anova` : data.frame. Modele anova issu de la fonction `anova()` ou `Anova()`

`col.dd1` : numeric. Indice de la colonne des degres de liberte

`col.sq` : numeric. Indice de la colonne des somme des carres des ecarts

`row.fact` : numeric. Indice de la ligne du facteur a tester

`base.comp` : numeric vector. Indice(s) de(s) ligne(s) des facteurs de la base de comparaison

Value

numeric list. Statistique de Fisher et les 2 ddl de la loi de fisher associee au test

Examples

```
# Creation du modele lineaire
```

- Dans le dossier **man** sous format **.rd**
- Autres tags : `@section`, `@seealso`, `@references`, etc.
- `\url{}`

Documentation (étape longue)

Documentation du package et des jeux de données

```
1 #' exemplePackage : le package qui vous facilite la vie.
2 #'
3 #' Ce package contient 2 fonctions d'aide a l'analyse de modeles anova, ainsi qu'a la manipulati
4 #'
5 #' @details \itemize{
6 #'   \item Nom : exemplePackage
7 #'   \item Version: 1.0.0
8 #'   \item License: GPL-2}
9 #'
10 #' @author Camille Cochener, Florian Kroell et Romain Moreau
11 #'
12 #' @seealso \code{\link{statFisher}}, \code{\link{tableConverter}}
13 #'
14 #' @references \url{http://r-pkgs.had.co.nz/} pour la creation de packages.
15 #'
16 #' @doctype package
17 #' @name exemplePackage
18 NULL
19
20 #' Evaluation sensorielle de viandes provenant de 4 fournisseurs
21 #'
22 #' Jeu de donnees contenant le resultat d'une experience sensorielle
23 #' regroupant 4 fournisseurs, 5 consommateurs sur 15 seances.
24 #'
25 #' @format data.frame avec 300 lignes et 4 colonnes :
26 #' \describe{
27 #'   \item{Fournisseur}{Numero du fournisseur (1 a 4)}
28 #'   \item{Consommateur}{Numero du consommateur (1 a 5)}
29 #'   \item{Seance}{Numero de la seance (1 a 15)}
30 #'   \item{Note}{Note attribuee (0 a 10)}
31 #' }
32 #' @source \url{http://emily.perso.math.cnrs.fr/Anova/Anova_files/viande.txt}
33 "viande"
34
```

Intégration de la documentation au package avec :
Ctrl + Shift + D ou `devtools::document()`

Autres composantes optionnelles

- Vignettes (`vignettes/`) : `devtools::use_vignette()`
- Code compilé (C++) : `devtools::use_rcpp()`

Vérification et publication

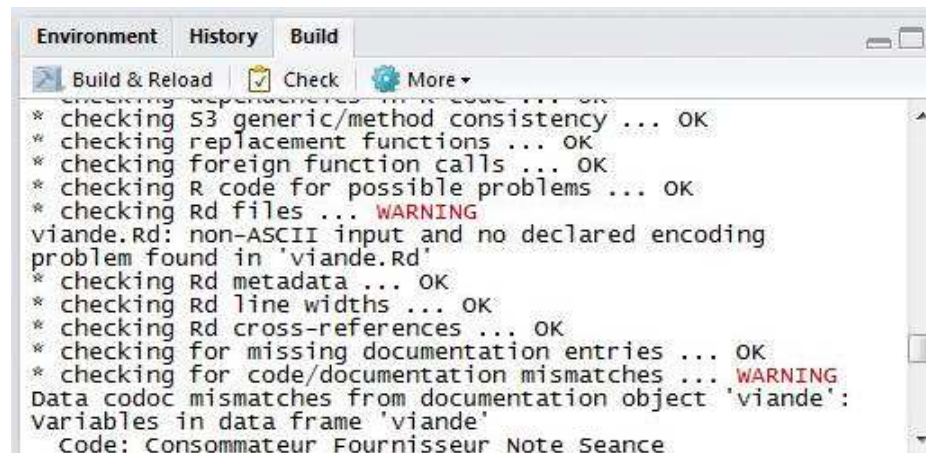
- Vérification :

Ctrl + Shift + E

Rend des errors (à réparer)

Rend des warnings (obligation de réparer si publication sur le CRAN)

Rend des notes



```
Environment History Build
Build & Reload Check More
* checking dependencies in R code ... OK
* checking S3 generic/method consistency ... OK
* checking replacement functions ... OK
* checking foreign function calls ... OK
* checking R code for possible problems ... OK
* checking Rd files ... WARNING
viande.Rd: non-ASCII input and no declared encoding
problem found in 'viande.Rd'
* checking Rd metadata ... OK
* checking Rd line widths ... OK
* checking Rd cross-references ... OK
* checking for missing documentation entries ... OK
* checking for code/documentation mismatches ... WARNING
Data codoc mismatches from documentation object 'viande':
Variables in data frame 'viande'
Code: Consommateur Fournisseur Note Seance
```

- Publication :

Donner la version du package (ex: 1.0)

Vérifier que le package respecte bien les règles du CRAN

Créer des fichiers README.md et NEWS.md

devtools::release() pour soumettre le package au CRAN

Conclusion

- Création d'un package **longue** et **fastidieuse**
- **Rstudio** permet de faciliter les choses
- Proposition du package au CRAN peut être rejetée
- Il faut être **persévérant** !



Merci de votre attention !

Bibliographie

- Agrocampus Ouest (NC). Construire un package R. 4p. http://www.agrocampus-ouest.fr/math/livreR/faire_pkg_R.pdf.
- Genolini C. (NC). Construire un package. 53 p. <https://cran.r-project.org/doc/contrib/Genolini-ConstruireUnPackage.pdf>.
- Wickham H. (NC). R packages. <http://r-pkgs.had.co.nz>.