

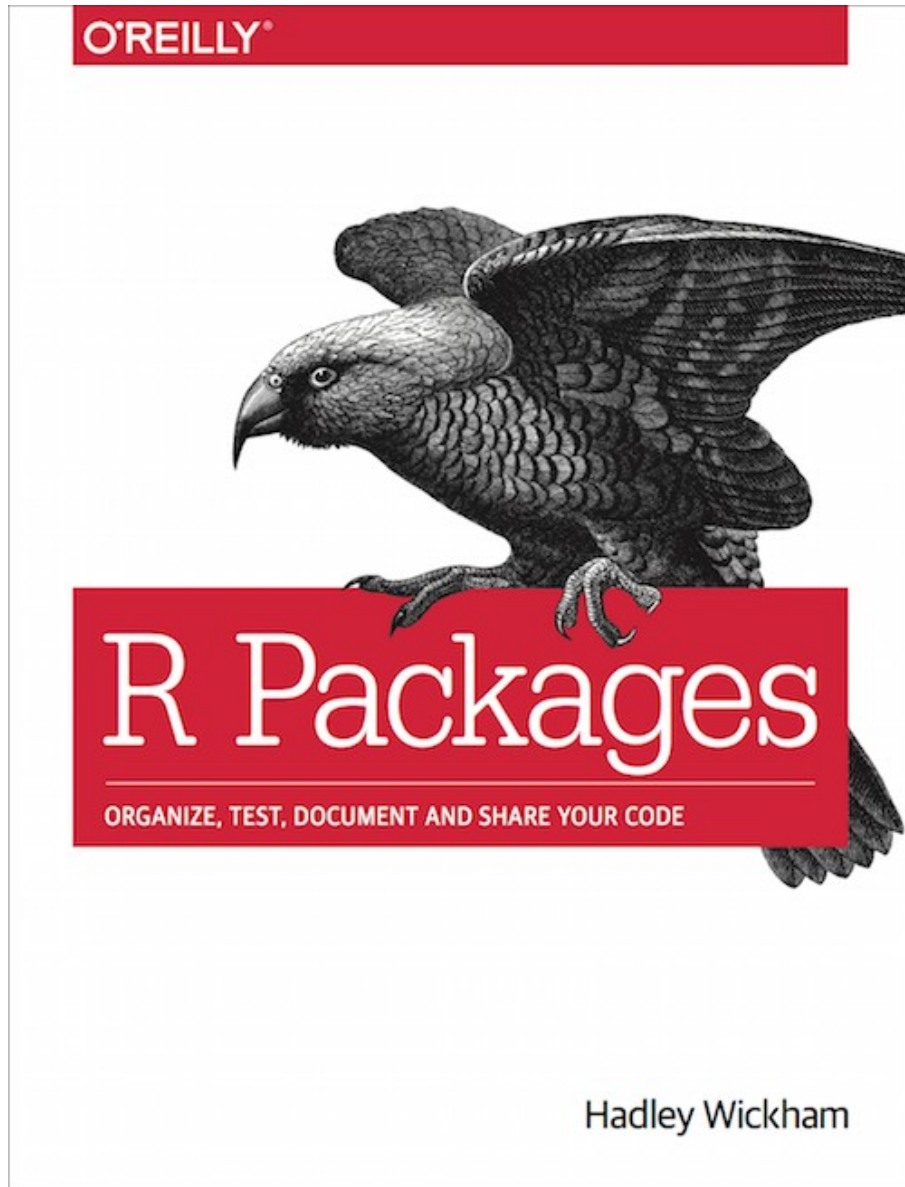


Pourquoi et comment mettre son code sous la forme de packages R ?



10 Mai 2023
Clémentine Decamps

Avant toute chose



<https://r-pkgs.org/>



Plan de la présentation



I. Pourquoi faire des packages ?

- définition
- avantages

II. Comment on fait

- le package
- la documentation
- les vignettes
- la diffusion



Un package c'est quoi déjà



- Des fonctions
- De la documentation (des fonctions, mais aussi des vignettes)
- Le tout dans un format que R comprend



Quand faire un package ?



- Si dans une analyse vous avez besoin de deux fois le même code (et vous copiez / collez) :

Quand faire un package ?

- Si dans une analyse vous avez besoin de deux fois le même code (et vous copiez / collez) :

→ **Vous devriez faire une fonction**

Quand faire un package ?

- Si dans une analyse vous avez besoin de deux fois le même code (et vous copiez / collez) :
 - **Vous devriez faire une fonction**
- Si entre deux analyses vous avez besoin de deux fois la même fonction (et vous copiez/collez) :

Quand faire un package ?

- Si dans une analyse vous avez besoin de deux fois le même code (et vous copiez / collez) :
 - **Vous devriez faire une fonction**
- Si entre deux analyses vous avez besoin de deux fois la même fonction (et vous copiez/collez) :
 - **Vous devriez faire un package**



Les avantages du package



Les avantages du package

→ **La reproductibilité !**

- Ranger son code
- Réutiliser facilement des fonctions
- Partager facilement des fonctions
- Minimise les erreurs

Les avantages du package

→ **La reproductibilité !**

- Ranger son code
- Réutiliser facilement des fonctions
- Partager facilement des fonctions
- Minimise les erreurs

→ **C'est facile donc pourquoi ne pas prendre l'habitude ?**



Plan de la présentation



I. Pourquoi faire des packages ?

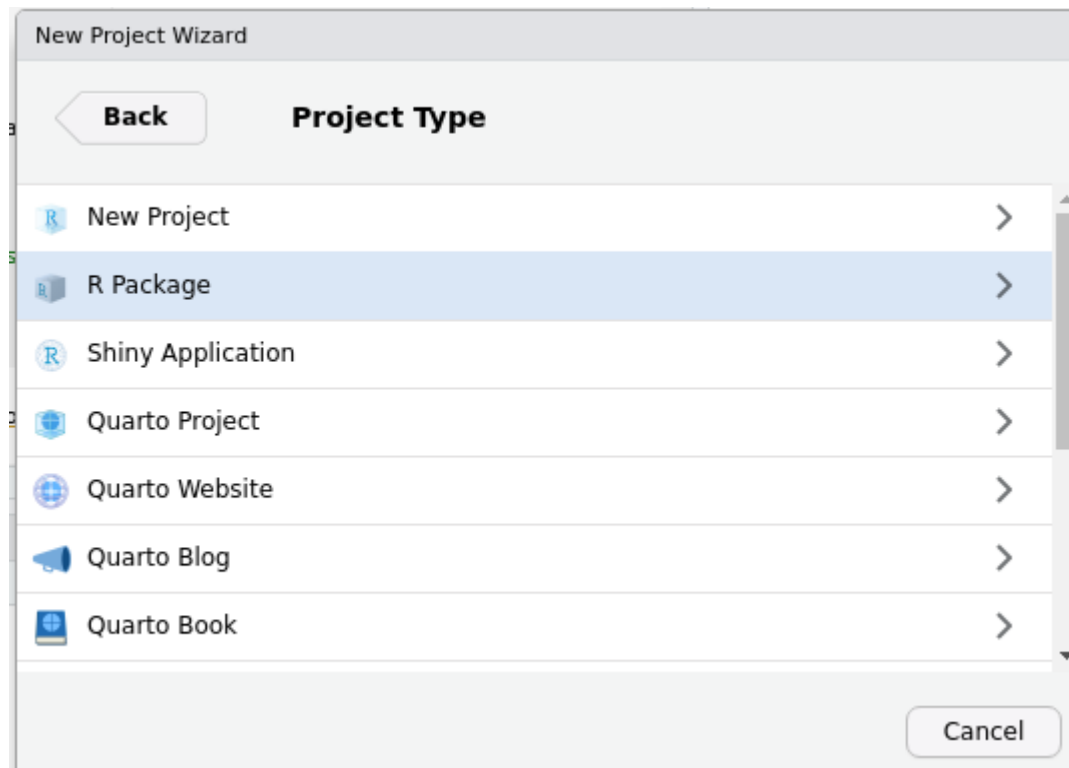
- définition
- avantages

II. Comment on fait

- le package
- la documentation
- les vignettes
- la diffusion

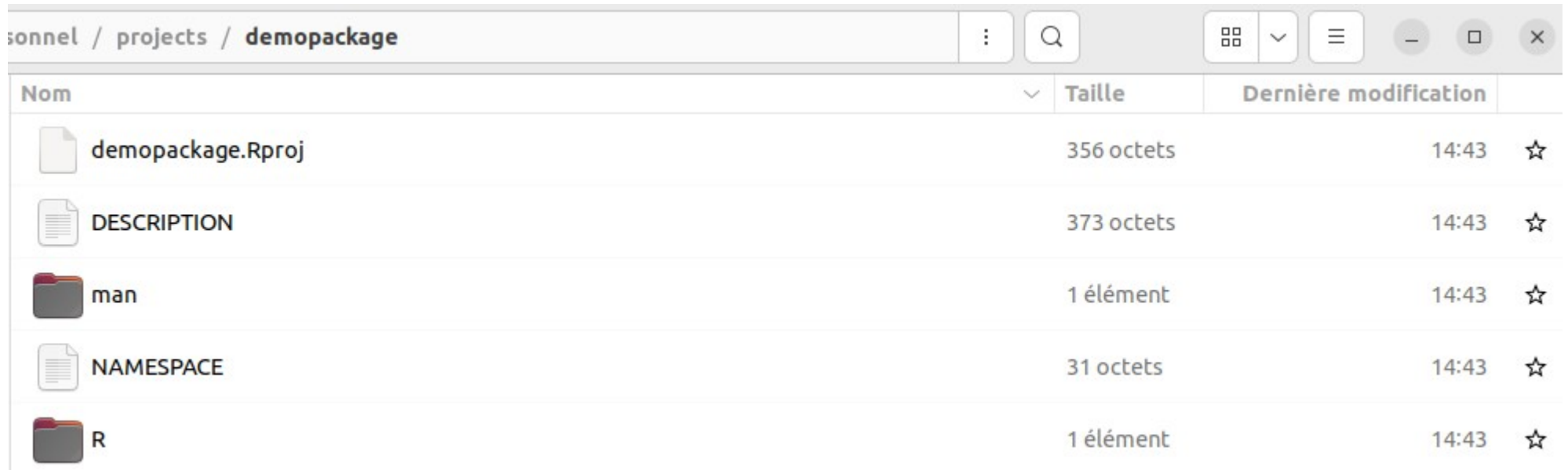
Comment créer le package

→ Rstudio, onglet package, et voilà.








Comment créer le package

→ Rstudio, onglet package, et voilà.

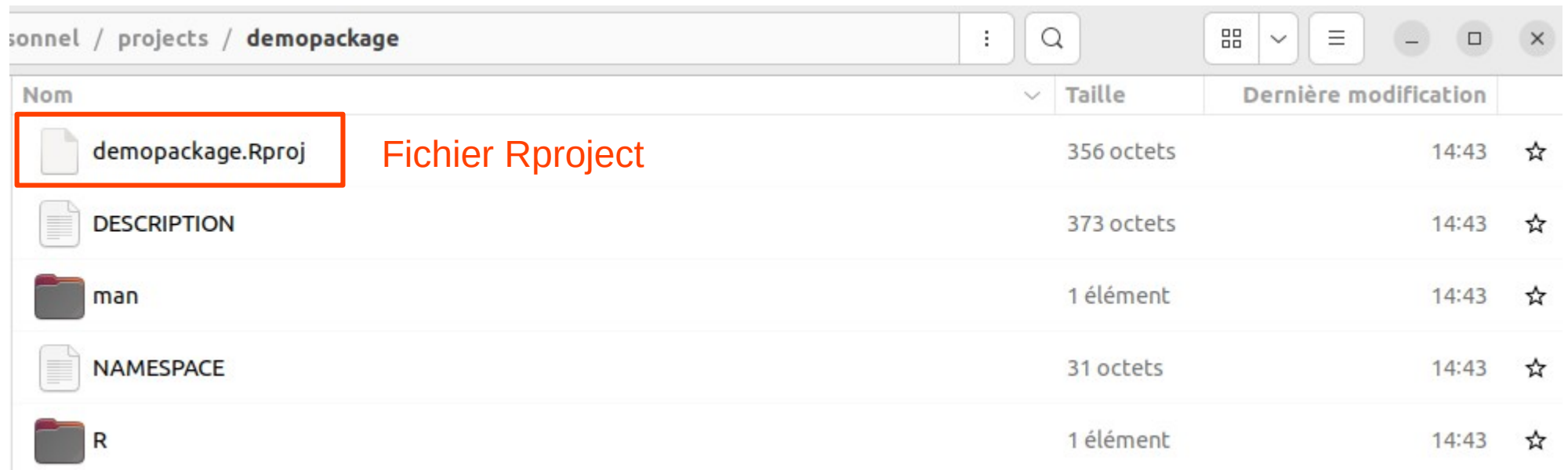


The screenshot shows the RStudio file explorer interface. The breadcrumb path is 'personnel / projects / demopackage'. The file list is as follows:






Nom	Taille	Dernière modification	
 demopackage.Rproj	356 octets	14:43	☆
 DESCRIPTION	373 octets	14:43	☆
 man	1 élément	14:43	☆
 NAMESPACE	31 octets	14:43	☆
 R	1 élément	14:43	☆

Comment créer le package

→ Rstudio, onglet package, et voilà.

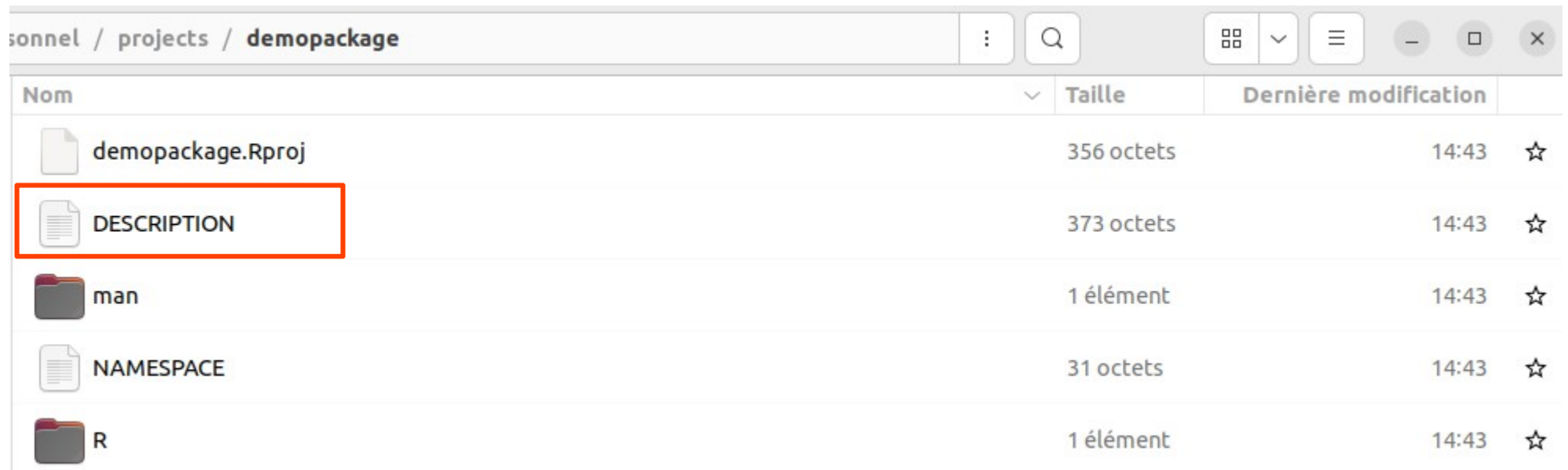


sonnel / projects / demopackage

Nom	Taille	Dernière modification
 demopackage.Rproj Fichier Rproject	356 octets	14:43 ☆
 DESCRIPTION	373 octets	14:43 ☆
 man	1 élément	14:43 ☆
 NAMESPACE	31 octets	14:43 ☆
 R	1 élément	14:43 ☆

Comment créer le package

→ Rstudio, onglet package, et voilà.

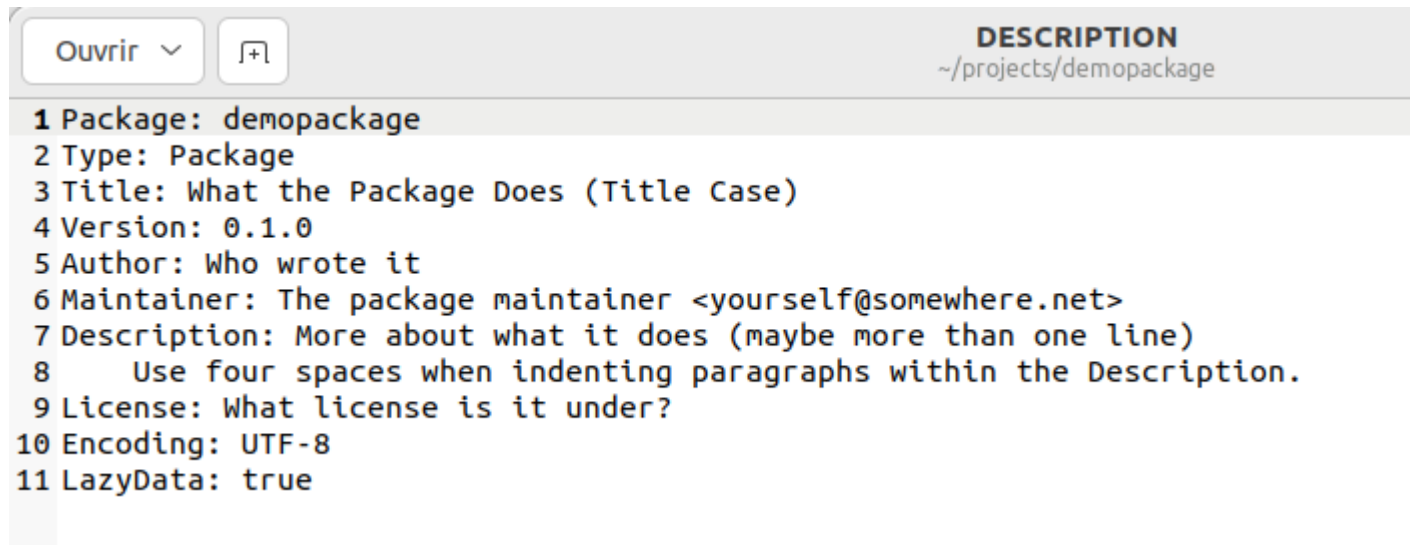


The screenshot shows the RStudio file explorer interface. The breadcrumb path is 'personnel / projects / demopackage'. The file list is as follows:

Nom	Taille	Dernière modification
demopackage.Rproj	356 octets	14:43 ☆
DESCRIPTION	373 octets	14:43 ☆
man	1 élément	14:43 ☆
NAMESPACE	31 octets	14:43 ☆
R	1 élément	14:43 ☆

Comment créer le package

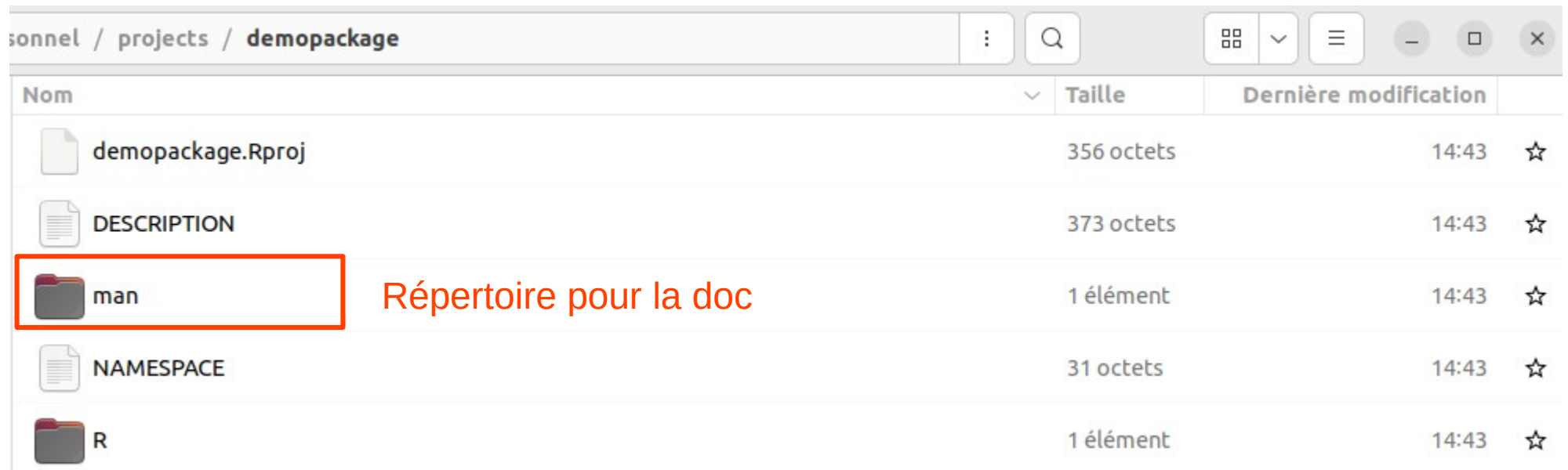
→ DESCRIPTION : tout est prérempli, il faut éditer avec vos informations



```
Ouvrir ▾ [+] DESCRIPTION
~/projects/demopackage
1 Package: demopackage
2 Type: Package
3 Title: What the Package Does (Title Case)
4 Version: 0.1.0
5 Author: Who wrote it
6 Maintainer: The package maintainer <yourself@somewhere.net>
7 Description: More about what it does (maybe more than one line)
8     Use four spaces when indenting paragraphs within the Description.
9 License: What license is it under?
10 Encoding: UTF-8
11 LazyData: true
```

Comment créer le package

→ Rstudio, onglet package, et voilà.

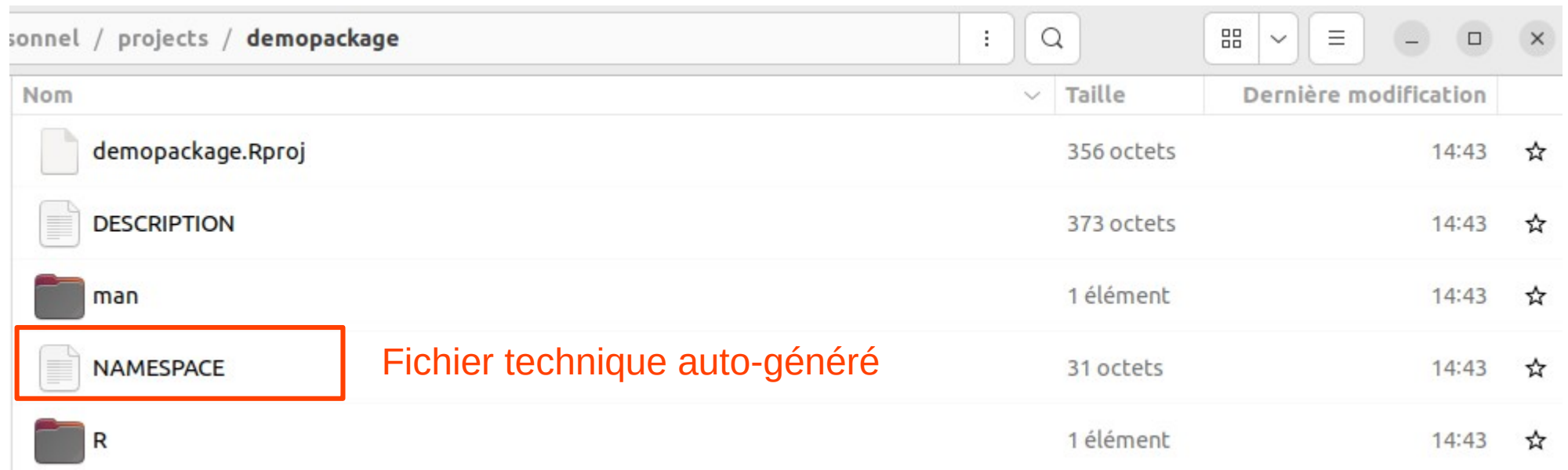


The screenshot shows the RStudio file explorer interface for a project named 'demopackage'. The breadcrumb path is 'personnel / projects / demopackage'. The file explorer displays a list of files and folders with columns for 'Nom', 'Taille', and 'Dernière modification'. The 'man' folder is highlighted with an orange border, and a red text annotation 'Répertoire pour la doc' is placed next to it.

Nom	Taille	Dernière modification
demopackage.Rproj	356 octets	14:43 ☆
DESCRIPTION	373 octets	14:43 ☆
man	1 élément	14:43 ☆
NAMESPACE	31 octets	14:43 ☆
R	1 élément	14:43 ☆

Comment créer le package

→ Rstudio, onglet package, et voilà.

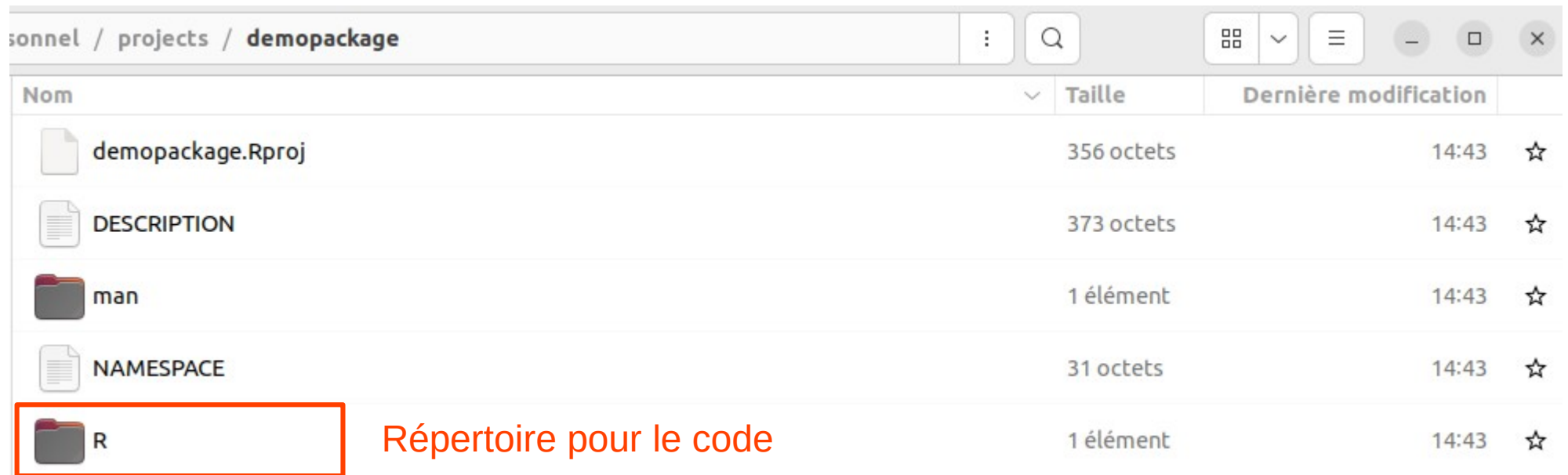


Nom	Taille	Dernière modification
demopackage.Rproj	356 octets	14:43 ☆
DESCRIPTION	373 octets	14:43 ☆
man	1 élément	14:43 ☆
NAMESPACE	31 octets	14:43 ☆
R	1 élément	14:43 ☆

Fichier technique auto-généré

Comment créer le package

→ Rstudio, onglet package, et voilà.



The screenshot shows the RStudio file explorer for a project named 'demopackage'. The interface includes a breadcrumb path 'personnel / projects / demopackage', a search bar, and window control buttons. The file list is organized into columns: 'Nom', 'Taille', and 'Dernière modification'. The files listed are 'demopackage.Rproj', 'DESCRIPTION', 'man', 'NAMESPACE', and 'R'. The 'R' folder is highlighted with a red box, and an orange text label 'Répertoire pour le code' points to it.

Nom	Taille	Dernière modification
demopackage.Rproj	356 octets	14:43 ☆
DESCRIPTION	373 octets	14:43 ☆
man	1 élément	14:43 ☆
NAMESPACE	31 octets	14:43 ☆
R	1 élément	14:43 ☆



Développement du package



Développement du package

- Package **devtools**

- `devtools::check` pour vérifier qu'on suit les règles de conformité

- `devtools::install` pour installer le package

- `devtools::build` pour créer la version compressée à partager

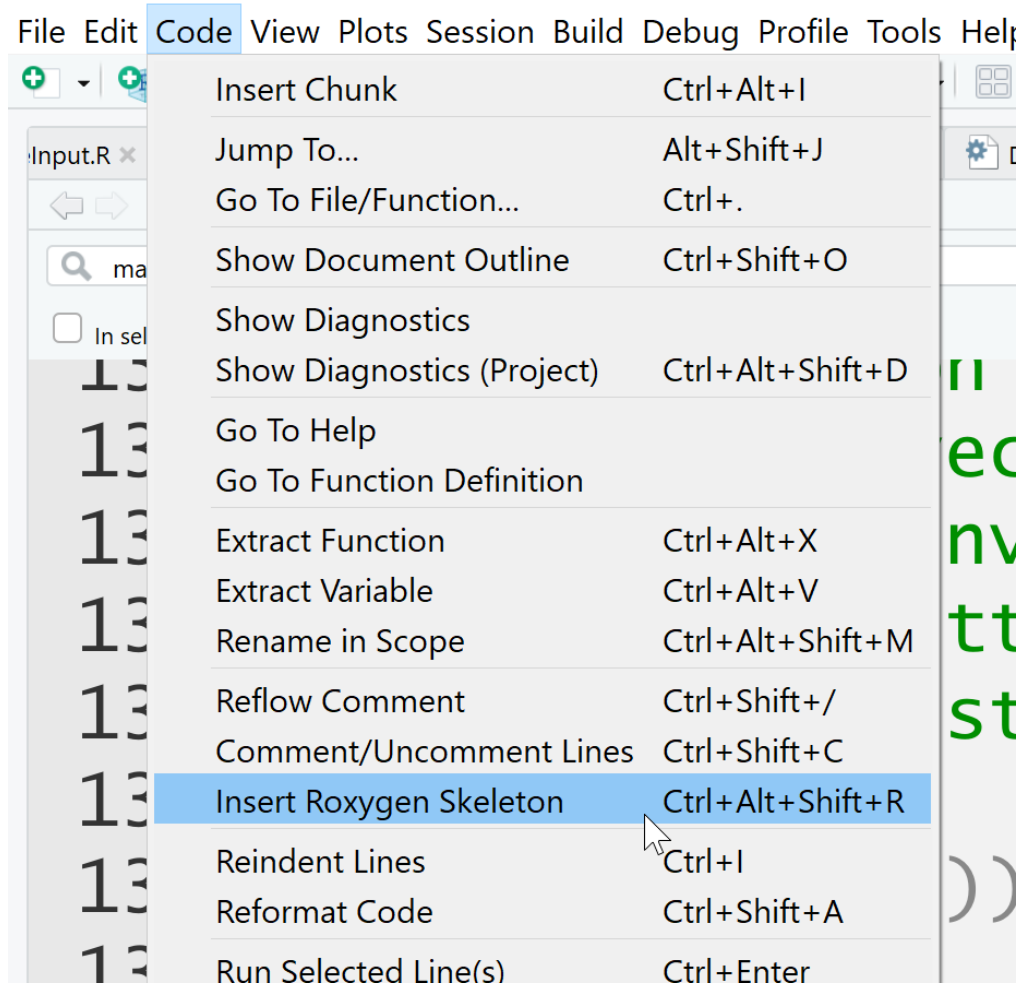
Développement du package

→ J'écris ma fonction

```
additionner.R x
Source on Save
1
2 additionne_ou_soustrait <- function(a, b, addition = T){
3   if (addition == T){
4     res = a + b
5   } else {
6     res = a - b
7   }
8   return(res)
9 }
```

Développement du package

→ Pour la documenter : [roxygen2](#)



Développement du package

→ Insérer le squelette Roxygen

```
additionner.R x
Source on Save
1
2 additionne_ou_soustrait <- function(a, b, addition = T){
3   if (addition == T){
4     res = a + b
5   } else {
6     res = a - b
7   }
8   return(res)
9 }
```



```
additionner.R* x
Source on Save
1
2 #' Title
3 #'
4 #' @param a
5 #' @param b
6 #' @param addition
7 #'
8 #' @return
9 #' @export
10 #'
11 #' @examples
12 additionne_ou_soustrait <- function(a, b, addition = T){
13   if (addition == T){
14     res = a + b
15   } else {
16     res = a - b
17   }
18   return(res)
19 }
20 |
```

Développement du package

→ Je documente ma fonction

```
1
2 #' Additionne ou soustrait
3 #'
4 #' @param a entier
5 #' @param b entier
6 #' @param addition boolean, si addition = TRUE la fonction additionne, sinon elle soustrait
7 #'
8 #' @return
9 #' @export
10 #'
11 #' @examples c = additionne_ou_soustrait(a = 4, b = 5)
12 additionne_ou_soustrait <- function(a, b, addition = T){
13   if (addition == T){
14     res = a + b
15   } else {
16     res = a - b
17   }
18   return(res)
19 }
```

Développement du package


→ Je documente ma fonction

```
1
2 #' Additionne ou soustrait
3 #'
4 #' @param a entier
5 #' @param b entier
6 #' @param addition boolean, si addition = TRUE la fonction additionne, sinon elle soustrait
7 #'
8 #' @return
9 #' @export
10 #'
11 #' @examples c = additionne_ou_soustrait(a = 4, b = 5)
12 additionne_ou_soustrait <- fonction(a, b, addition = T){
13   if (addition == T){
14     res = a + b
15   } else {
16     res = a - b
17   }
18   return(res)
19 }
```

```
.....
> devtools::document()
i Updating demopackage documentation
i Loading demopackage
Writing additionne_ou_soustrait.Rd
> ?additionne_ou_soustrait
.....
```

→ Permet de compiler la documentation

Développement du package



The screenshot shows an R IDE window with the following content:

```
additionne_ou_soustrait {demopackage} R Documentation
```

Additionne ou soustrait

Description

Additionne ou soustrait

Usage

```
additionne_ou_soustrait(a, b, addition = T)
```

Arguments

a	entier
b	entier
addition	boolean, si addition = TRUE la fonction additionne, sinon elle soustrait

Examples

```
c = additionne_ou_soustrait(a = 4, b = 5)
```

Développement du package

- Package **devtools**
 - `devtools::install()` pour installer le package
 - `devtools::check()` pour vérifier qu'on suit les règles de conformité
 - `devtools::build()` pour créer la version compressée à partager



Comment créer les vignettes



Et pour faire des vignettes ?

Comment créer les vignettes

Package `usethis`

→ `usethis::use_vignette("vignette_test")`

Génère le squelette pour les vignettes

```
1 ---
2 title: "vignette_test"
3 output: rmarkdown::html_vignette
4 vignette: >
5   %\VignetteIndexEntry{vignette_test}
6   %\VignetteEngine{knitr::rmarkdown}
7   %\VignetteEncoding{UTF-8}
8 ---
9
10 ```{r, include = FALSE}
11 knitr::opts_chunk$set(
12   collapse = TRUE,
13   comment = "#>"
14 )
15 ```
16
17 ```{r setup}
18 library(demopackage)
19 ```
20 |
```

Comment créer les vignettes

Package `usethis`

→ `usethis::use_vignette("vignette_test")`

On écrit la vignette

```
20
21 # Addition
22
23 Utilisez cette fonction pour faire des additions.
24
25 ```{r addition}
26 additionne_ou_soustrait(a = 1, b = 2, addition = TRUE)
27 ```
28
29
30 # Soustraction
31
32 Ou alors des soustractions.
33
34 ```{r addition}
35 additionne_ou_soustrait(a = 1, b = 2, addition = FALSE)
36 ```
--
```

→ `devtools::build_vignettes()`

Comment créer les vignettes

→ Et voilà on a notre vignette en html

vignette_test

```
library(demopackage)
```

Addition

Utilisez cette fonction pour faire des additions.

```
additionne_ou_soustrait(a = 1, b = 2, addition = TRUE)  
#> [1] 3
```

Soustraction

Ou alors des soustractions.

```
additionne_ou_soustrait(a = 1, b = 2, addition = FALSE)  
#> [1] -1
```

On peut aussi faire un readme github direct

→ `usethis::use_readme_rmd()`

Génère un document readme pour la package github du projet

demopackage

The goal of demopackage is to ...

Installation

You can install the development version of demopackage like so:

```
# FILL THIS IN! HOW CAN PEOPLE INSTALL YOUR DEV PACKAGE?
```

Example

This is a basic example which shows you how to solve a common problem:

```
library(demopackage)
## basic example code
```

What is special about using `README.Rmd` instead of just `README.md` ? You can include R chunks like so:

```
summary(cars)
#>      speed          dist
#>  Min.   : 4.0    Min.    : 2.00
#> 1st Qu.:12.0    1st Qu.: 26.00
#>  Median :15.0    Median : 36.00
#>   Mean  :15.4    Mean   : 42.98
#> 3rd Qu.:19.0    3rd Qu.: 56.00
#>   Max.  :25.0    Max.   :120.00
```

You'll still need to render `README.Rmd` regularly, to keep `README.md` up-to-date. `devtools::build_readme()` is handy for this. You could also use GitHub Actions to re-render `README.Rmd` every time you push. An example workflow can be found here: <https://github.com/r-lib/actions/tree/v1/examples>



Et ensuite ?





Et ensuite ?



→ Utiliser git :
usethis::use_git()

Et ensuite ?

- Utiliser git :
usethis::use_git()
- Faire des tests unitaires :
package **testthat** <https://testthat.r-lib.org/>

Et ensuite ?

- Utiliser git :
`usethis::use_git()`
- Faire des tests unitaires :
package **testthat** <https://testthat.r-lib.org/>
- Diffuser sur le CRAN ?
`usethis::use_release_issue()`

Sources

R package de Hadley Wickham et Jennifer Bryan :

<https://r-pkgs.org>

Cours « travail collaboratif en R » de Lino Galiana :

<https://linogaliana.gitlab.io/collaboratif/>

Vignette de Roxygen2 :

<https://cran.r-project.org/web/packages/roxygen2/vignettes/roxygen2.html>

Package de devtools :

<https://cran.r-project.org/web/packages/devtools/devtools.pdf>

Package usethis :

<https://usethis.r-lib.org/>