

# Documentación

*Julio 2019*

Traducido por Leonardo Collado-Torres

@fellgernon

lcolladotor@gmail.com

lcolladotor.github.io

Desarrollado por Charlotte Wickham para rstudio::conf(2019)

@cvwickham

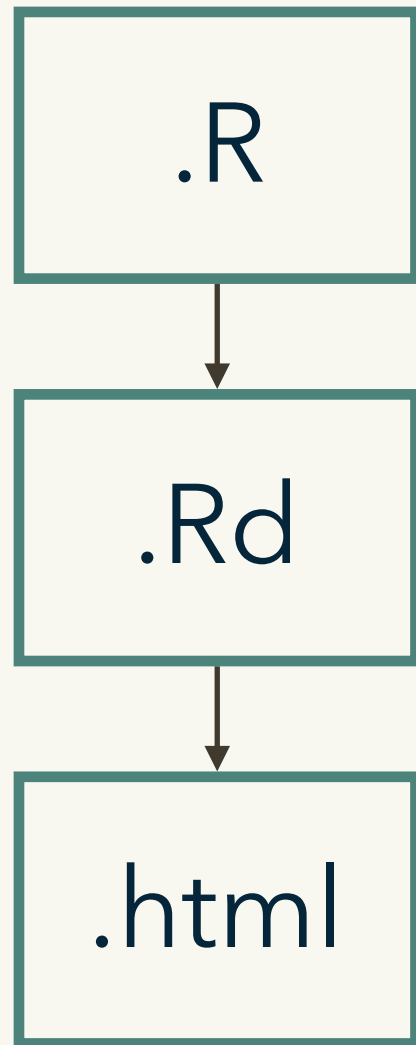
cwickham@gmail.com

cwick.co.nz

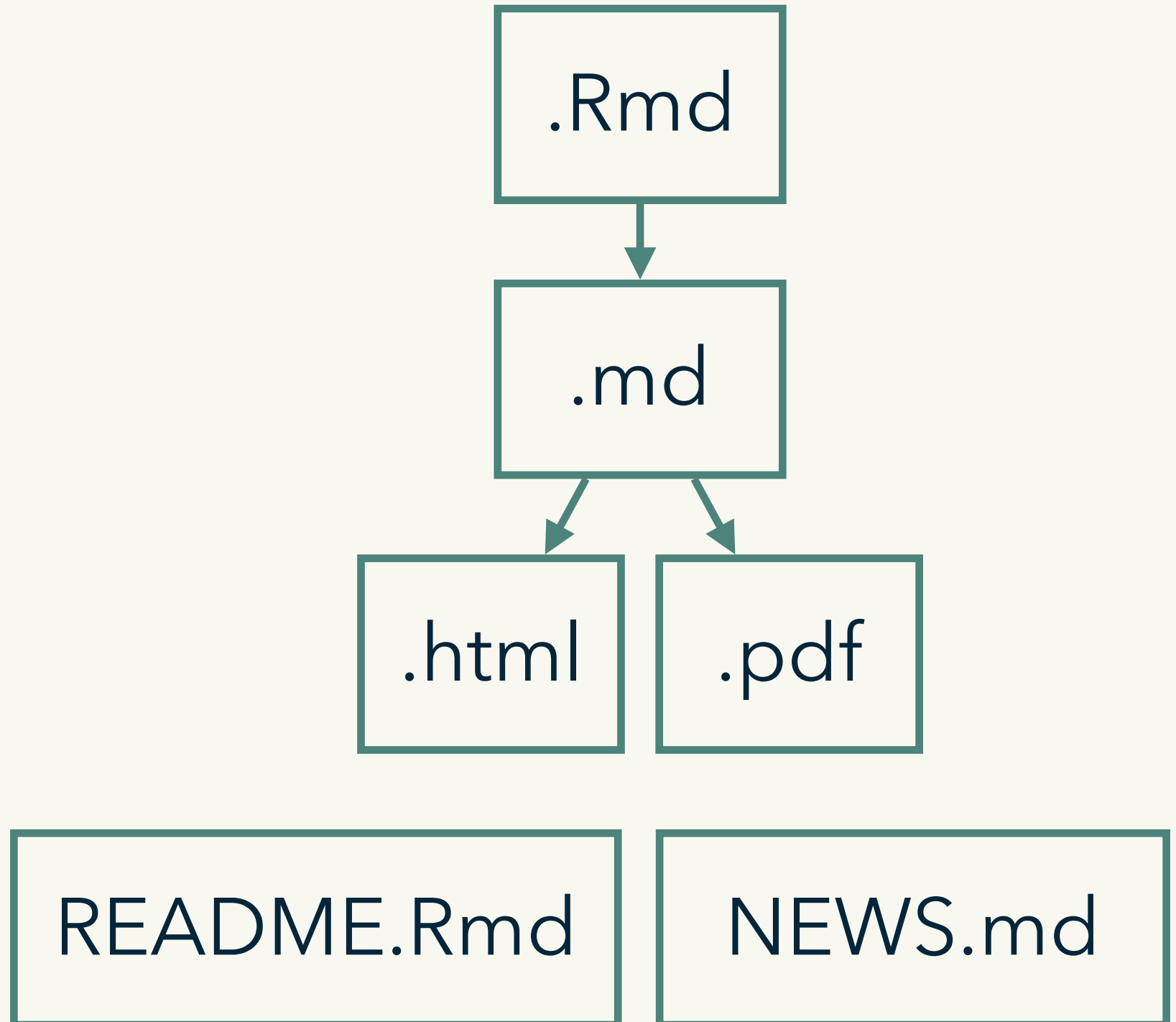
Adapted from *Tidy Tools* by Hadley Wickham



A nivel de función con  
**roxygen2**



A nivel de paquete  
con **rmarkdown**



# Markdown

Asumimos que ya lo conocen

# Formato básico de markdown

# Este es un título principal

Esto es un poco de texto. Pon en *itálico* el texto usando un solo guión bajo (o estrellas). Ponlo en **negritas** usando dos estrellas (o guiones bajos). Aquí esta la [liga a la guía de markdown](http://bit.ly/19fAexE).

- \* Esto es una lista (sin números)

- \* Este es otro elemento

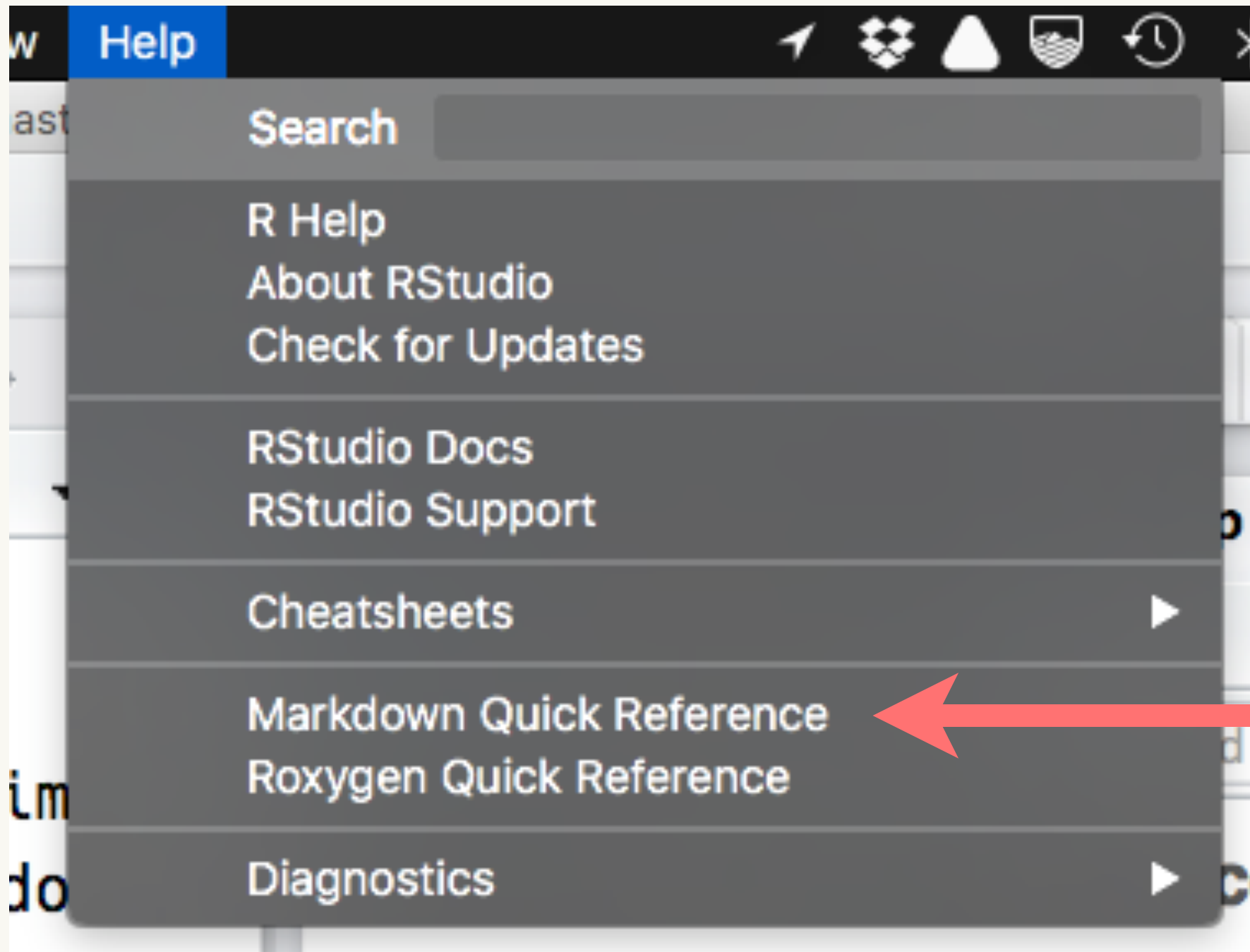
```
```R
```

```
# Código de R
```

```
f <- function() x + 1
```
```

## Este es un título secundario

También puedes escribir `código junto al texto`, listas enumeradas, citas y demás.



# Documentando funciones con roxygen2

# Roxxygen2

**roxxygen2**

**R**



<http://r-pkgs.had.co.nz/man.html>

Puedes escribir comentarios bajo un formato específico en .R

## En R/add\_col.R

## #' Agrega una columna a un data frame

# '

## #' Te permite especificar la posición. En caso de que ya exista la variable

```
#' reemplazará la existente por la nueva.
```

# '

```
#' @param x Un data frame
```

#' @param name Nombre de la variable a crear. Si una variable con ese nombre

#' ya existe será reemplazada.

```
#' @param value Valores a insertar.
```

```
#' @param where Posición en donde insertar. Usa 1 para insertar en el lado
```

```
#' izquierdo, o -1 para insertar en el lado derecho.
```

## #' @examples

```
#' df <- data.frame(x = 1:5)
```

```
#' add_col(df, "y", runif(5))
```

```
#' add_col(df, "y", runif(5), where = 1)
```

# '

```
#' add_col(df, "x", 5:1)
```



# Puedes escribir comentarios bajo un formato específico en .R

En R/add\_col.R

## #' Add a column to a data frame

# '

#' Allows you to specify the position. Will replace existing variable

#' with the same name if present.

# '

```
#' @param x A data frame
```

```
#' @param name Name of variable to create. If a variable of that name
```

```
#' already exists it will be replaced
```

```
#' @param value Values to insert.
```

#' @param where Position to insert. Use 1 to insert on LHS, or -1 to insert on

#' RHS.

## #' @examples

```
#' df <- data.frame(x = 1:5)
```

```
#' add_col(df, "y", runif(5))
```

```
#' add_col(df, "y", runif(5), where = 1)
```

# '

```
#' add_col(df, "x", 5:1)
```

Puedes escribir comentarios bajo un formato específico en `.R`

# Comentario de roxygen

```
#' Add a column to a data frame
#
# Allows you to specify the position. Will replace existing variable
# with the same name if present.
#
# @param x A data frame
# @param name Name of variable to create. If a variable of that name
# already exists it will be replaced
# @param value Values to insert.
# @param where Position to insert. Use 1 to insert on LHS.
# @examples
# df <- data.frame(x = 1:5)
# add_col(df, "y", runif(5))
# add_col(df, "y", runif(5), where = 1)
#
# add_col(df, "x", 5:1)
```

# Etiqueta de roxygen

# Roxygen los traduce a un **.Rd**

En `man/add_col.Rd`

```
% Generated by roxygen2: do not edit by hand
% Please edit documentation in R/add_col.R
\name{add_col}
\alias{add_col}
\title{Add a column to a data frame}
\usage{
  add_col(x, name, value, ncol(x) + 1)
}
\arguments{
  \item{x}{A data frame}

  \item{name}{Name of variable to create. If a variable of that name
already exists it will be replaced}

  \item{value}{Values to insert.}

  \item{where}{Position to insert. Use 1 to insert on LHS.}
}
\description{
  Allows you to specify the position. Will replace existing variable
with the same name if present.
}
```

En casi todos los  
casos puedes  
ignorar estos  
archivos

# Add a colum to a data frame

## Description

Similar to cbind, but allows you to specify the positoin. Will replac existing variable with the same name if present.

## Usage

```
add_col(x, name, value, where = ncol(x) + 1)
```

## Arguments

- x** A data frame
- name** Name of variabl to create. If variable of that name already exists
- value** Values to insert.
- where** position to insert. Use 1 to insert on LHS, or -1 to insert on RHS.

## Details

The columns in

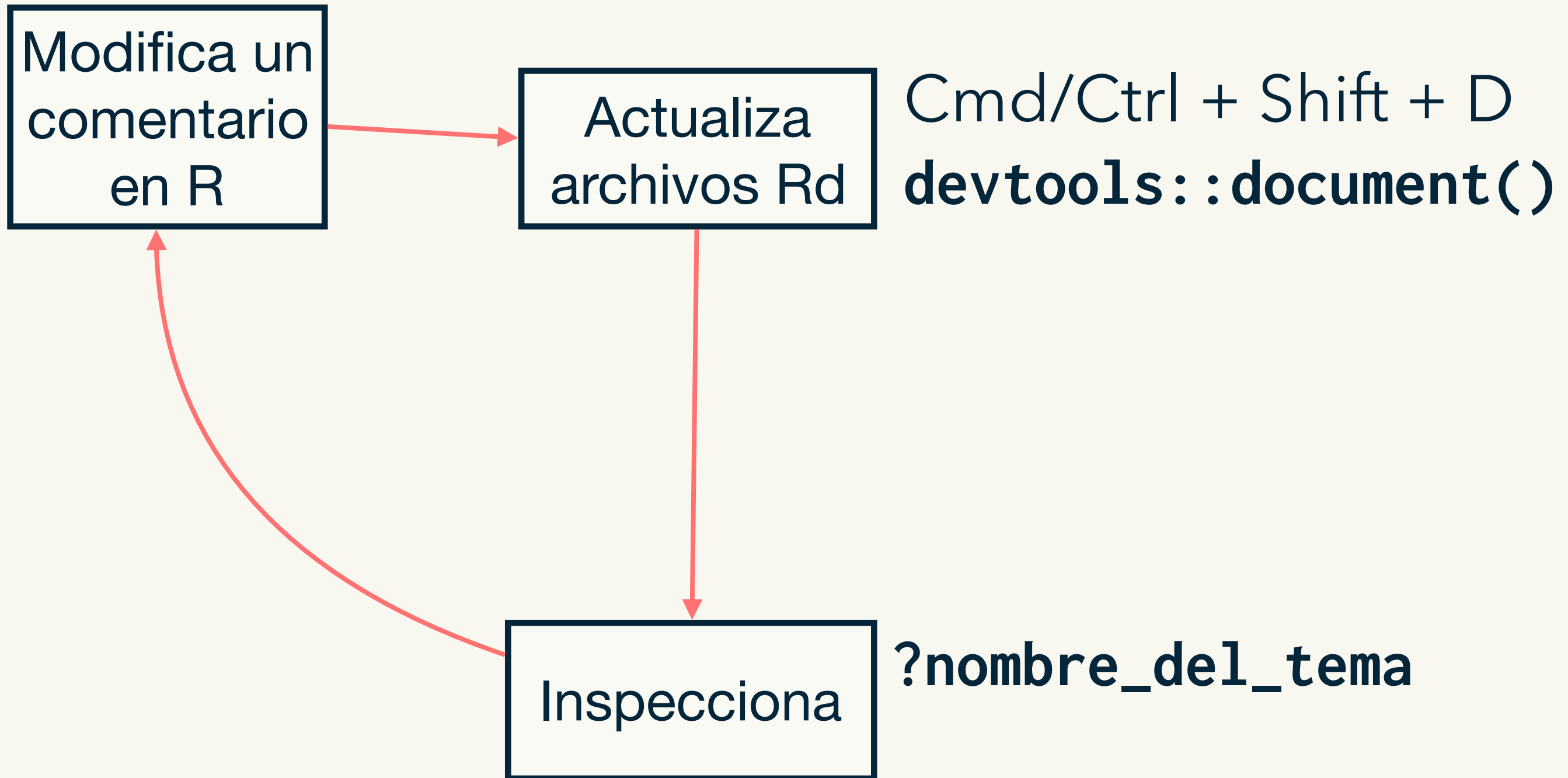
## Examples

```
df <- data.frame(x = 1:5)
add_col(df, "y", runif(5))
add_col(df, "y", runif(5), where = 1)

add_col(df, "x", 5:1)
```

R los traduce en un  
.html para que los  
puedas ver

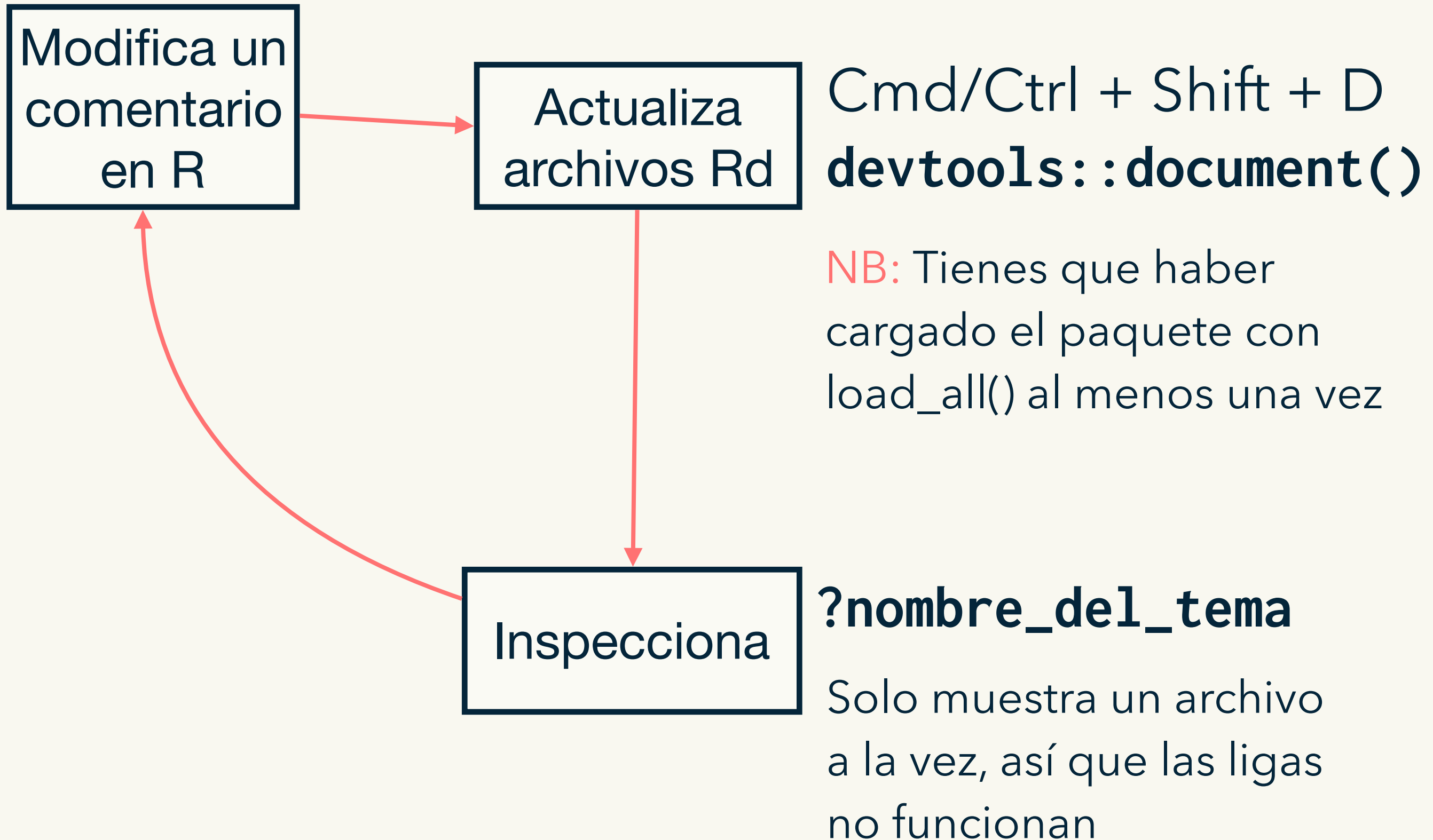
# Flujo de trabajo para documentar



# Dos advertencias

1. Tienes que haber cargado el paquete al menos una vez con `load_all()`.  
Verifica que salga el mensaje "Using development documentation..."
2. Esta técnica solo construye archivos individuales así las ligas no funcionan.

# Flujo de trabajo para documentar



Cambia al proyecto:

[hadcol-doc]



# Tu turno

Arregla los errores de dedo en la documentación de `add_col()`.

Utiliza el flujo de trabajo para documentar para inspeccionar el resultado de tu trabajo.

R: Sum of Vector Elements

La primera oración es el **título**

## Sum of Vector Elements

### Description

sum returns the sum of all the values present in its arguments.

### Usage

```
sum(..., na.rm = FALSE)
```

### Arguments

... numeric or complex or logical vectors.

na.rm logical. Should missing values (including NaN) be removed?

### Details

This is a generic function: methods can be defined for it directly or via the [Summary](#) group generic. For this to work properly, the arguments ... should be unnamed, and dispatch is on the first argument.

If na.rm is FALSE an NA or NaN value in any of the arguments will cause a value of NA or

El siguiente párrafo es la **descripción**

Todo lo demás son los **detalles**

# El párrafo de la descripción

La primera oración es el **título**

```
#' Sum of vector elements
#'  
#' \code{sum} returns the sum of all the values present in its arguments.  
#'  
#' This is a generic function: methods can be defined for it directly or via the  
#' \code{\link{Summary}} group generic. For this to work properly, the arguments  
#' \code{...} should be unnamed, and dispatch is on the first argument.
```

El siguiente párrafo  
es la **descripción**

Todo lo demás son los **detalles**

Hay cinco **etiquetas** que usarás para la mayoría de las funciones

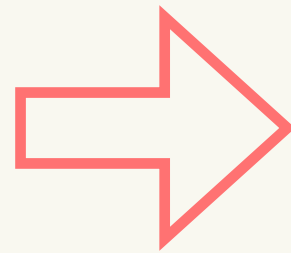
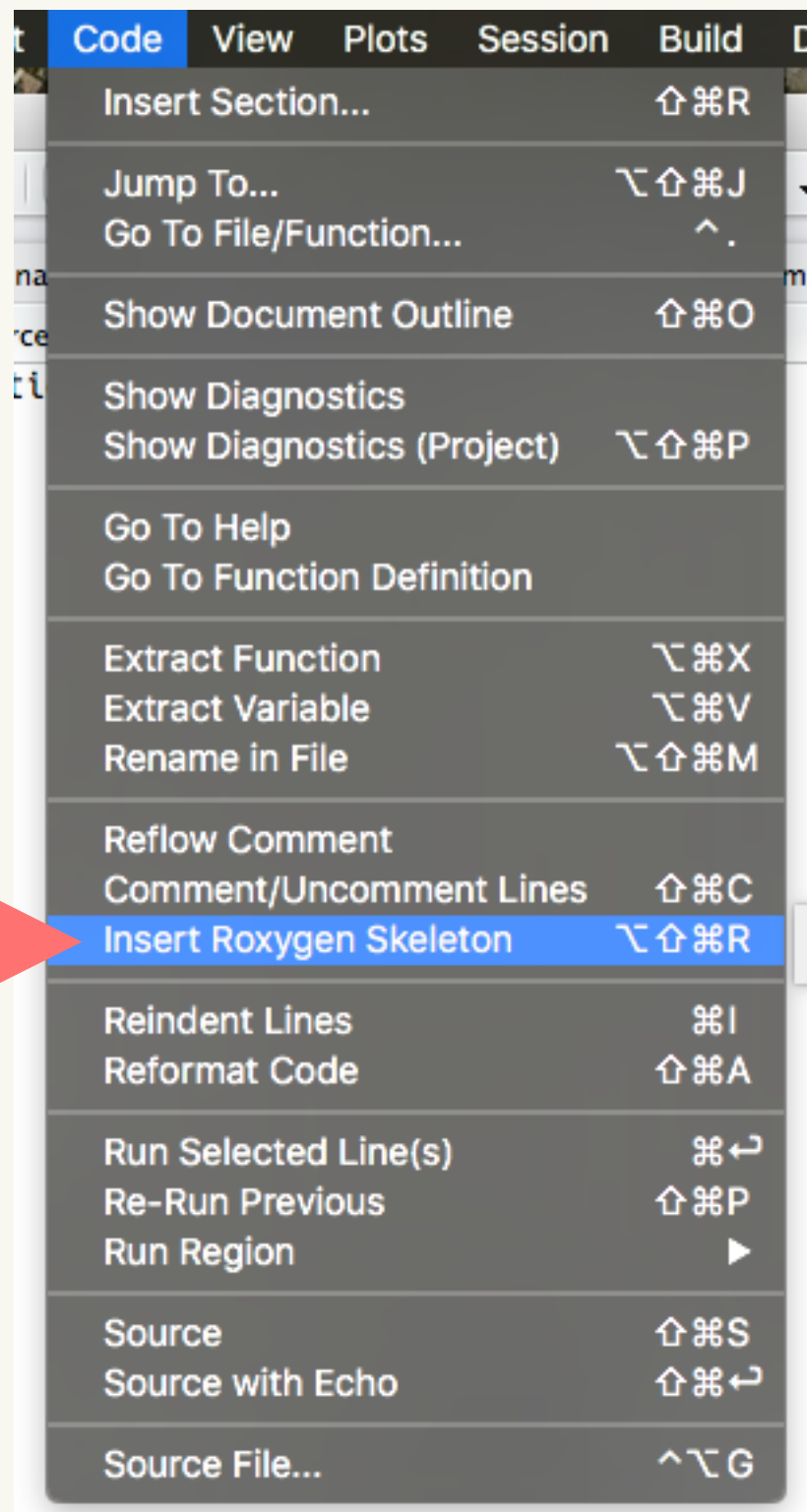
| Etiqueta   | Propósito   |
|------------|---|
| @param arg | Describe valores de entrada   |
| @examples  | Muestra cómo funciona la función.<br>(Atajos de RStudio usualmente funcionan) |
| @seealso   | Ligas a funciones relacionada   |
| @return    | Describe valores de salida  |
| @export    | ¿Esta función es visible para el usuario?                                     |

# Tu turno

Documenta `insert_into()`.

(Checa la próxima diapositiva  
para una pista)

# RStudio te ayuda a recordar



```
#' Título
#'
#' @param x
#' @param y
#' @param z
#'
#' @return
#' @export
#'
#' @examples
fun <- function(x, y, z) {
}
```

# Usa markdown para el formato

```
# En un nuevo paquete, actívalo al correr  
# usethis::use_roxygen_md()
```

```
**negritas**, _itálico_, `código`
```

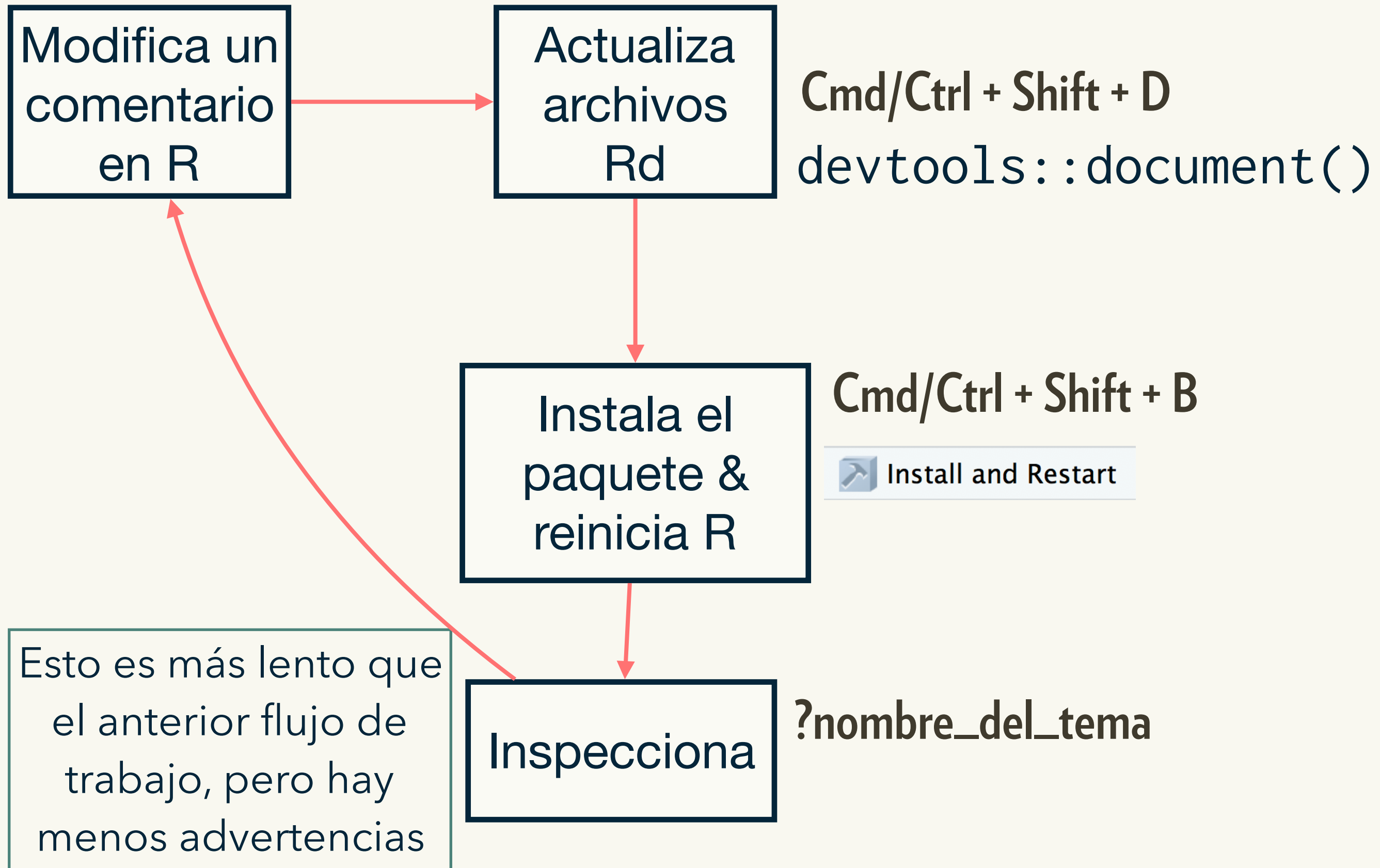
```
* [función()]
```

```
* [paquete::función()]
```

```
* [texto de la liga][función()]
```

```
* [texto de la liga][paquete::función()]
```

# Flujo de trabajo 2 para documentar





# Tu turno

En `add_col.R`

Crea una liga real hacia `cbind()`

Agrega una sección *ver también* (`@seealso`) a `add_col()` y `insert_into()` que ligue a las funciones.

¿Qué pasa si agregas `@family xyz` a las dos funciones?

roxygen2 viene con otras herramientas para reducir copias

```
# Documenta múltiples funciones en el mismo archivo
```

```
#' @rdname add_col
```

```
# Hereda las descripciones de parámetros de
```

```
# otra función
```

```
#' @inheritParams add_col
```

```
# Hereda todo de otra función
```

```
#' @inherit add_col
```

```
# Hereda selectivamente
```

```
#' @inherit add_col parameters return references
```

```
#' title description details
```

```
#' sections seealso
```

Lee en línea sobre como documentar otros objetos

## Datos

<http://r-pkgs.had.co.nz/data.html#documenting-data>

## Clases & métodos

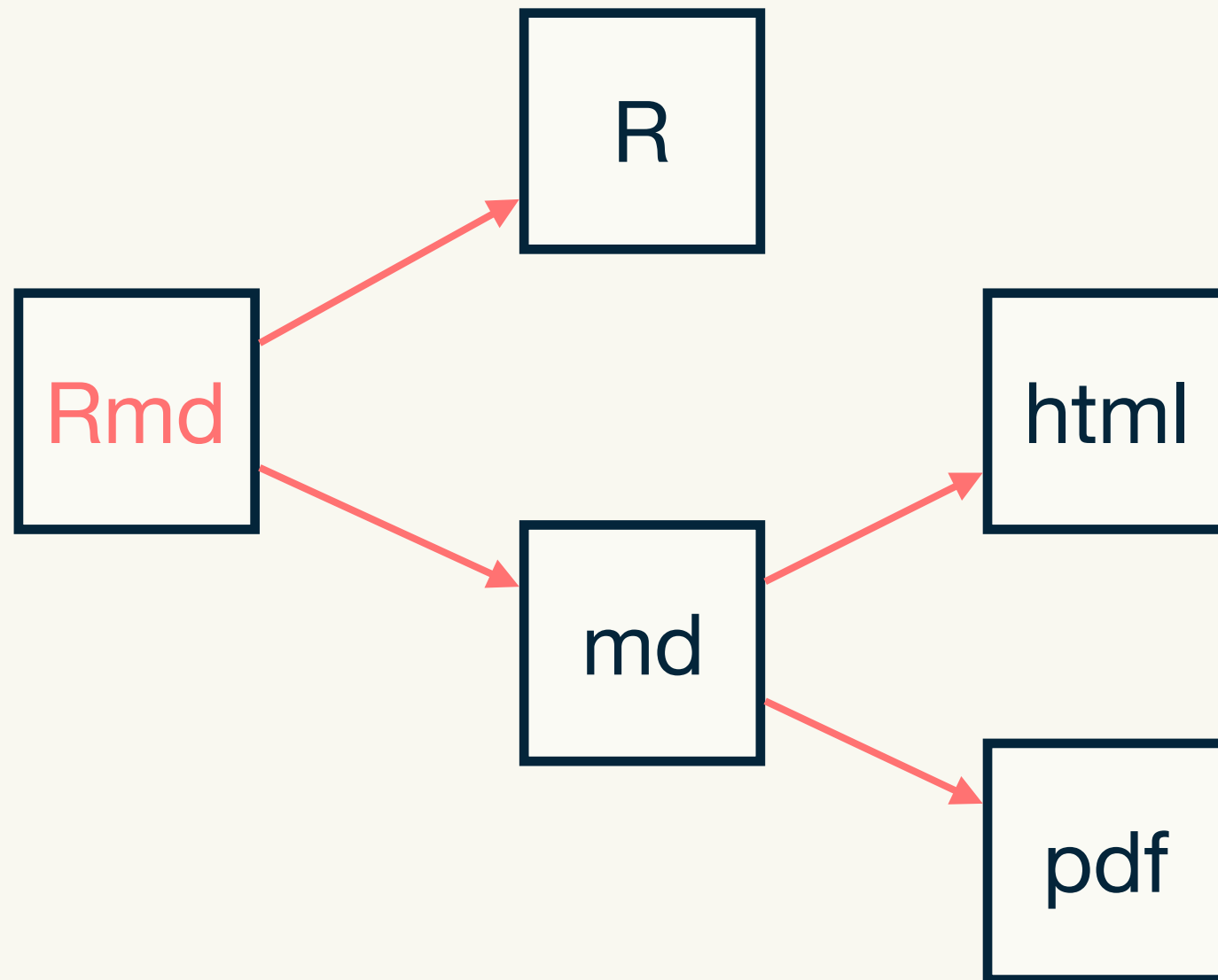
<http://r-pkgs.had.co.nz/man.html#man-classes>

## Paquetes

<http://r-pkgs.had.co.nz/man.html#man-packages>

# Documentación de paquetes con rmarkdown

# Vignettes



Te permiten combinar prosa y código para explicar como funciona tu paquete de R.

¡La parte difícil es escribir, no la tecnología!

La forma más sencilla de empezar es usando `use_vignette()`

```
usethis::use_vignette("nombre")
```

```
# Agrega lo siguiente al archivo DESCRIPTION
```

```
Suggests: knitr
```

```
VignetteBuilder: knitr
```

```
# Crea el directorio vignettes/
```

```
# Inicial el borrador en vignettes/nombre.Rmd
```

# Vignette = Rmarkdown + información especial

---

```
title: "Título de la vignette"
author: "Autor de la vignette"
date: "`r Sys.Date()`"
output: rmarkdown::html_vignette
vignette: >
```

Formato especial para  
vignettes

```
%\VignetteIndexEntry{Título de la vignette}
%\VignetteEngine{knitr::rmarkdown}
%\VignetteEncoding{UTF-8}
```

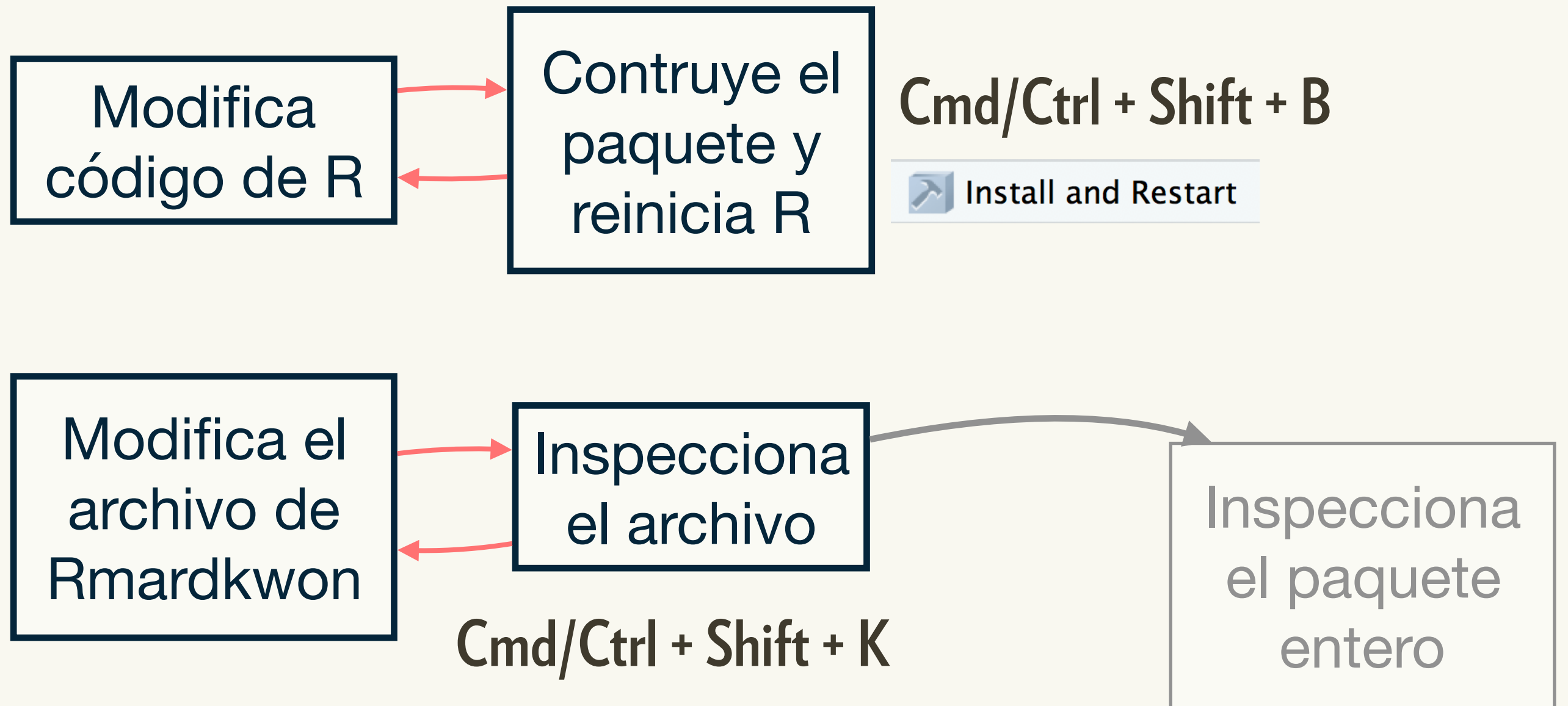
Información especial  
que require R

---

Vignettes son un tipo de documentación larga típicamente incluida en los paquetes. Porque son distribuidas con los paquetes, tienen que ser lo más compactas posibles. El formato de salida de `html\_vignette` provee un templado específico (y cambia algunas opciones) para asegurarse que el archivo html sea lo más pequeño posible. El formato `html\_vignette`:

...

# Flujo de trabajo para vignettes



```
devtools::install(build_vignettes = TRUE)
```

```
browseVignettes()
```



# Si lo vas a compartir con otros, usa un README

```
# Es tu decisión: pero comúnmente es útil mostrar  
# resultados de código [Leo: es la opción _rmd() ]  
usethis::use_readme_md()  
usethis::use_readme_rmd()
```

```
# Para proyectos públicos este archivo debería  
# incluir un pequeño resumen, instrucciones de cómo  
# instalarlo, y algunos ejemplos. Para proyectos  
# privados, este es un excelente lugar para  
# escribir tus notas
```

Si cambia con el tiempo, documenta los cambios a tu API

```
usethis::use_news_md()
```

```
# Para proyectos públicos, aquí deberían  
# venir documentados los cambios importantes  
# desde el punto de vista del usuario. Es  
# crítico que describas los cambios al API.  
# Esto es menos importante para proyectos  
# privados.
```



Adapted from *Tidy Tools* by Hadley Wickham

This work is licensed as

Creative Commons  
Attribution-ShareAlike 4.0  
International

To view a copy of this license, visit  
[https://creativecommons.org/  
licenses/by-sa/4.0/](https://creativecommons.org/licenses/by-sa/4.0/)