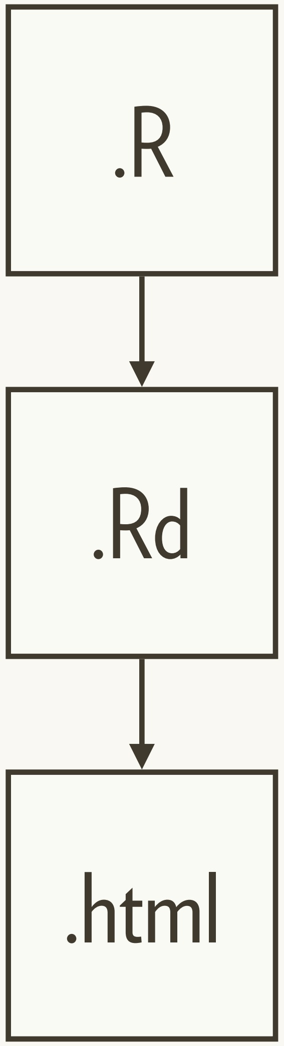
Documentation

January 2018

Hadley Wickham
@hadleywickham
Chief Scientist, **RStudio**

Function-level with roxygen2

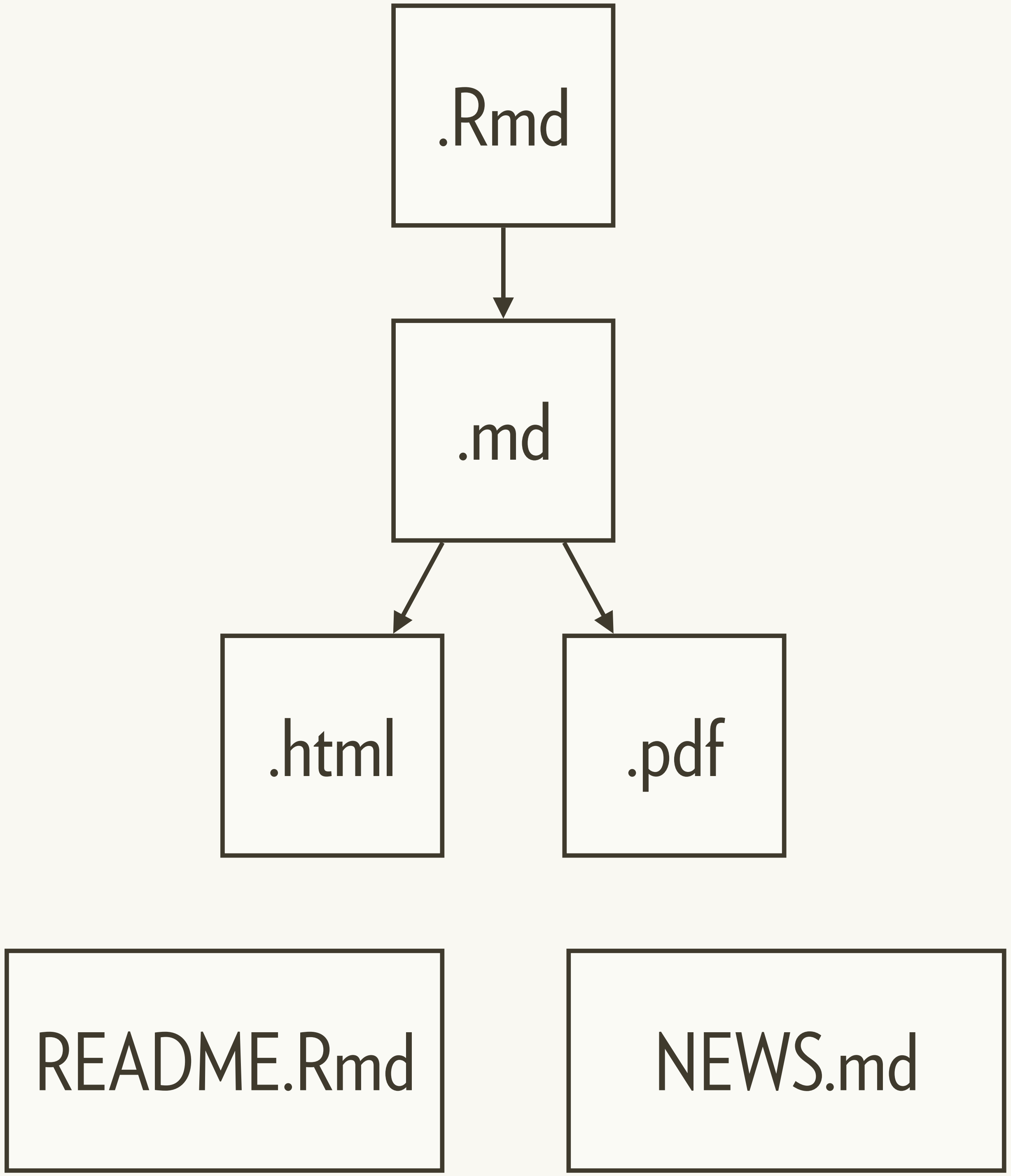


.R

.Rd

.html

Package-level with rmarkdown



.Rmd

.md

.html

.pdf

README.Rmd

NEWS.md

Markdown

I assume you are already familiar with it

Basic markdown formatting

~~~~
# This is a top level heading

This is some text. Make text _italic_ with single  underscores (or
stars). Make it **bold** with double stars (or underscores). Here is a
[link to a markdown guide](http://bit.ly/19fAexE).

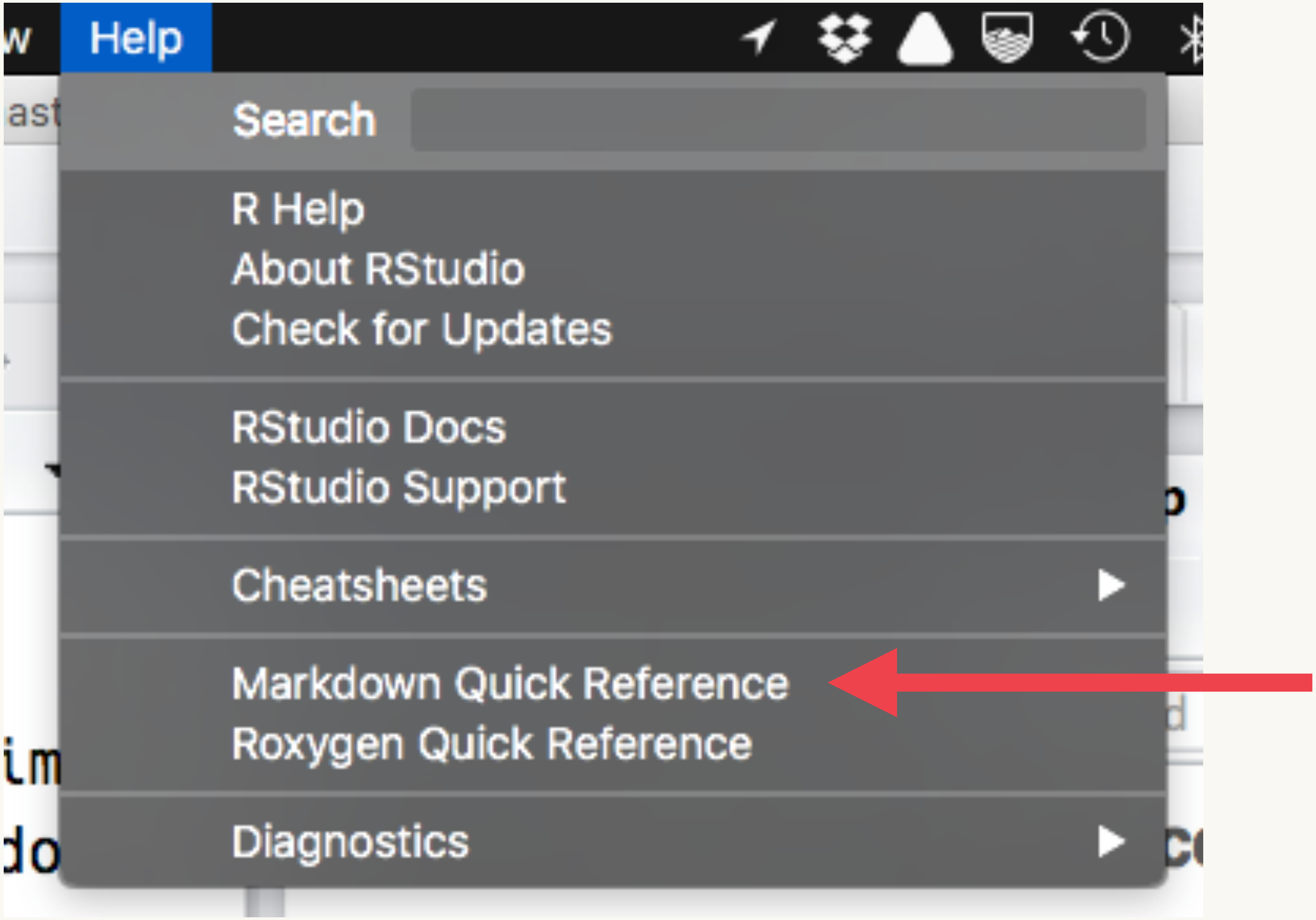
* This is a list

* This is another item

```R
# Some R code
f <- function() x + 1
```

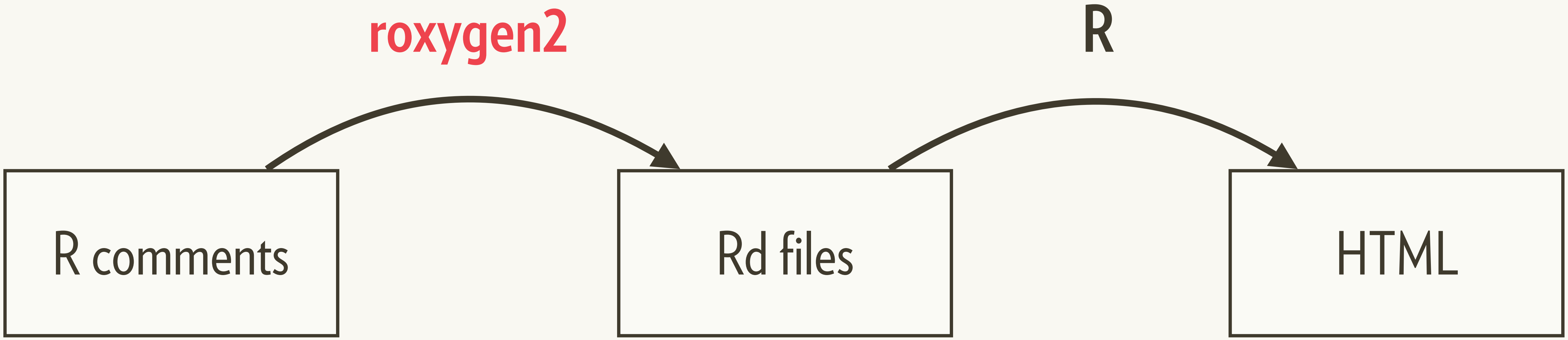
## This is a secondary heading

You can also do `inline code`, numbered lists and quotes and more.
~~~~


Help
Search
R Help
About RStudio
Check for Updates
RStudio Docs
RStudio Support
Cheatsheets
Markdown Quick Reference
Roxygen Quick Reference
Diagnostics

Function documentation with roxygen2

Roxygen2

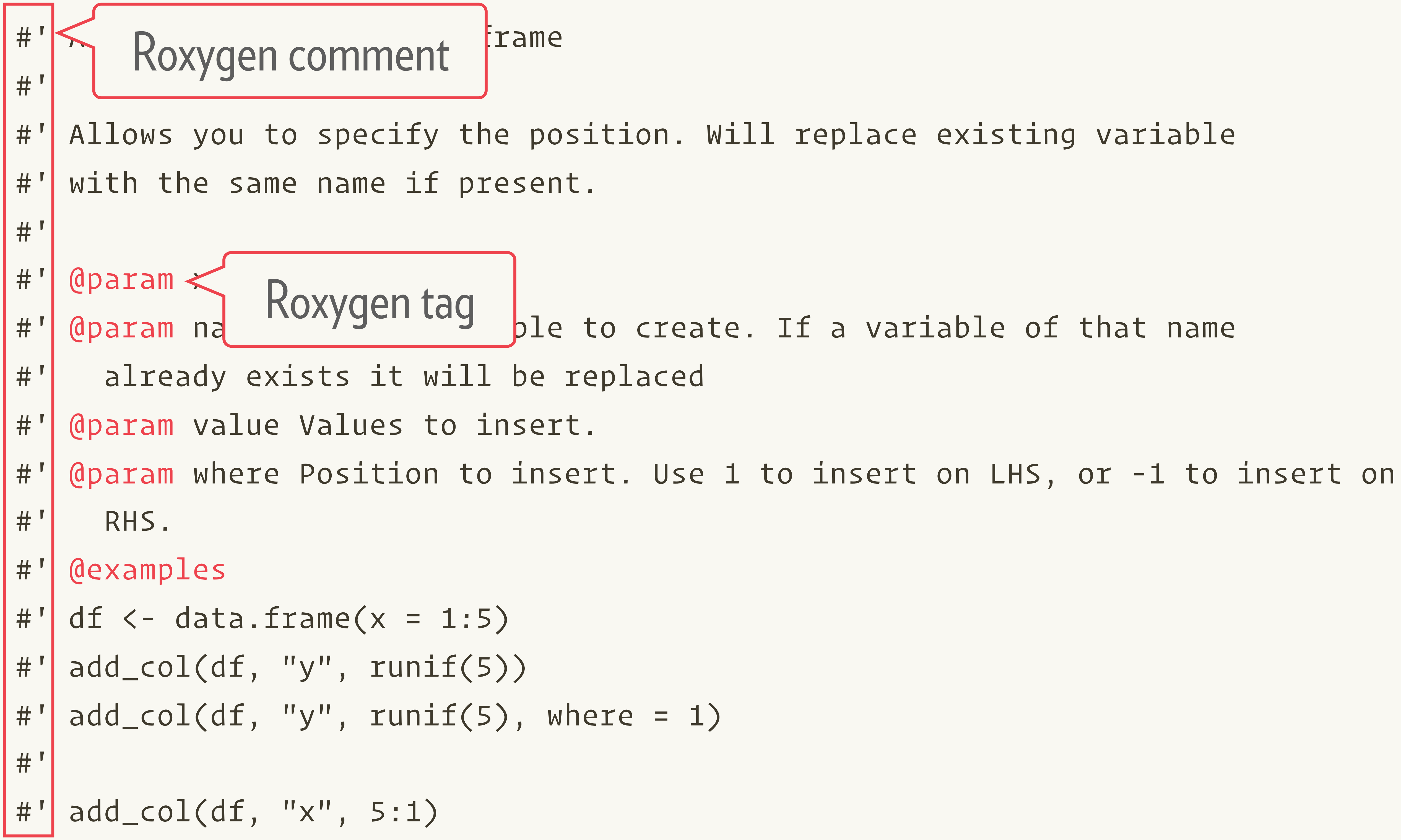


http://r-pkgs.had.co.nz/man.html

You write specially formatted comments in .R

```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)
```

You write specially formatted comments in .R



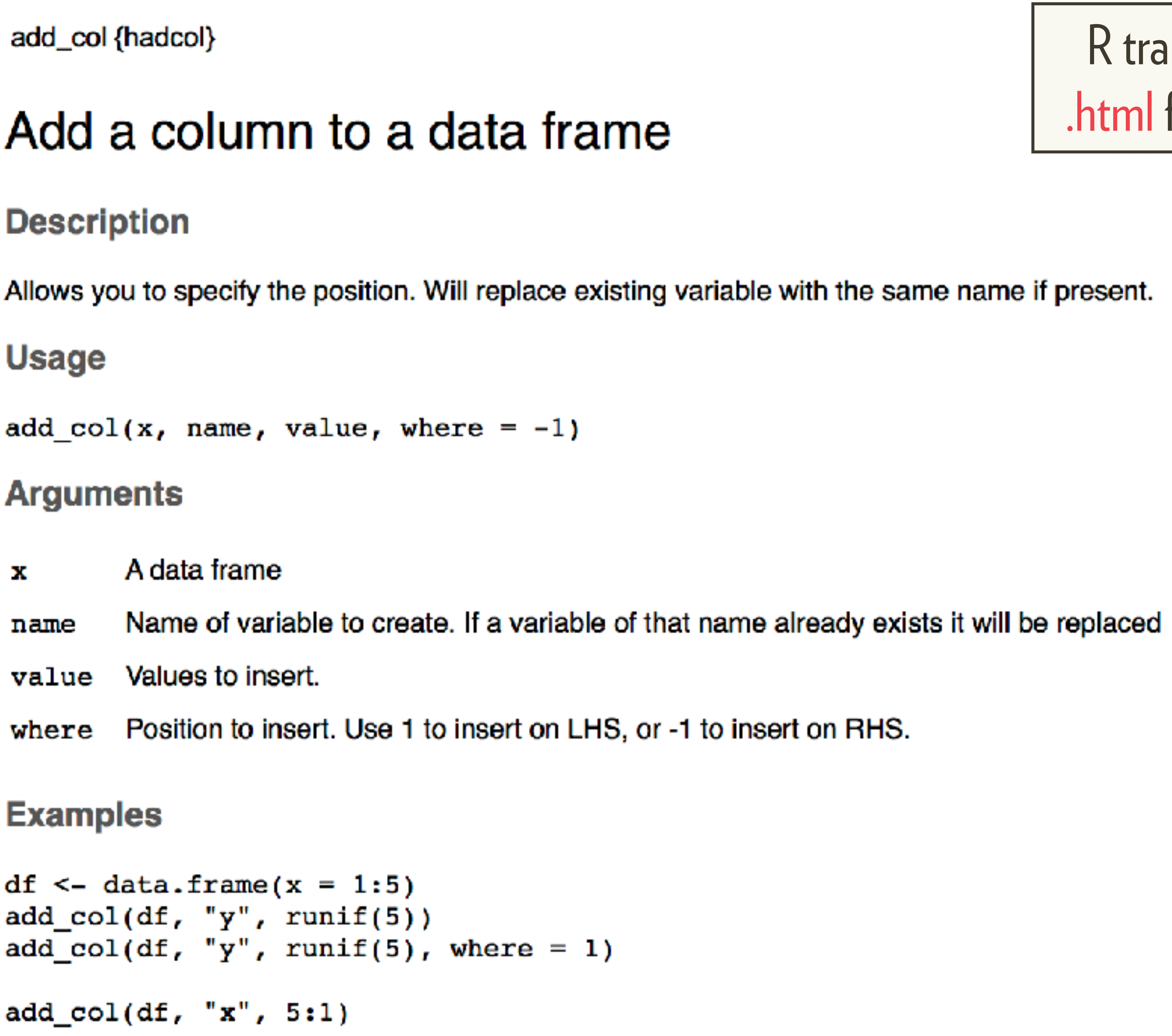
```
#' A                     frame
#'
#' Allows you to specify the position. Will replace existing variable
#' with the same name if present.
#'
#' @param x
#' @param na                ble to create. If a variable of that name
#'   already exists it will be replaced
#' @param value Values to insert.
#' @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)
```

Roxygen translates to .Rd

In almost all cases you can ignore these files

```
% 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, where = -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, or -1 to insert on
RHS.}
}
\description{
Allows you to specify the position. Will replace existing variable
with the same name if present.
}
```

R translates to .html for viewing



add_col {hadcol}

Add a column to a data frame

Description

Allows you to specify the position. Will replace existing variable with the same name if present.

Usage

```
add_col(x, name, value, where = -1)
```

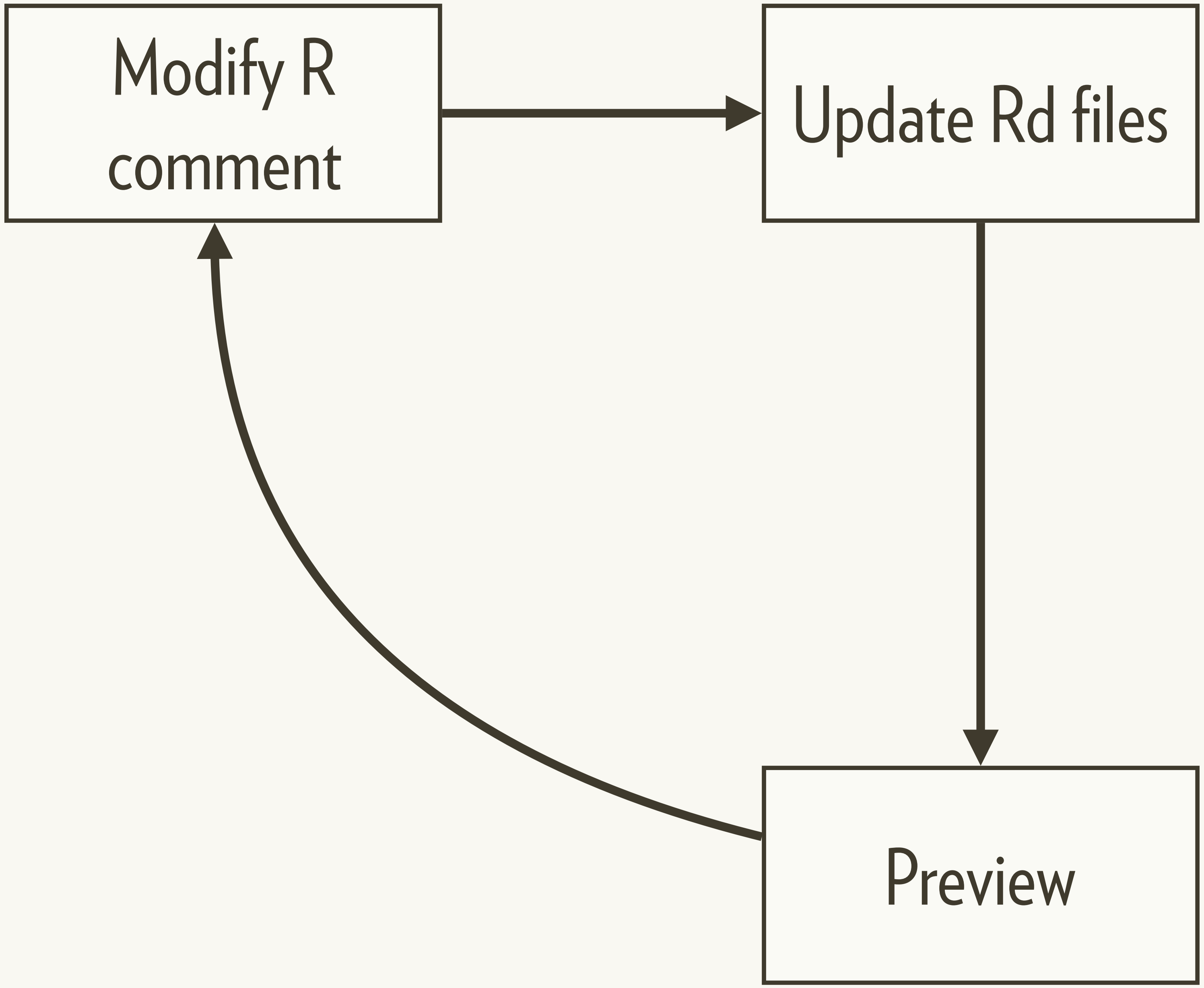
Arguments

`x`	A data frame
`name`	Name of variable to create. If a variable of that name already exists it will be replaced
`value`	Values to insert.
`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)
```

Documentation workflow



Modify R comment

Update Rd files

Preview

Cmd/Ctrl + Shift + D

```
devtools::document()
```

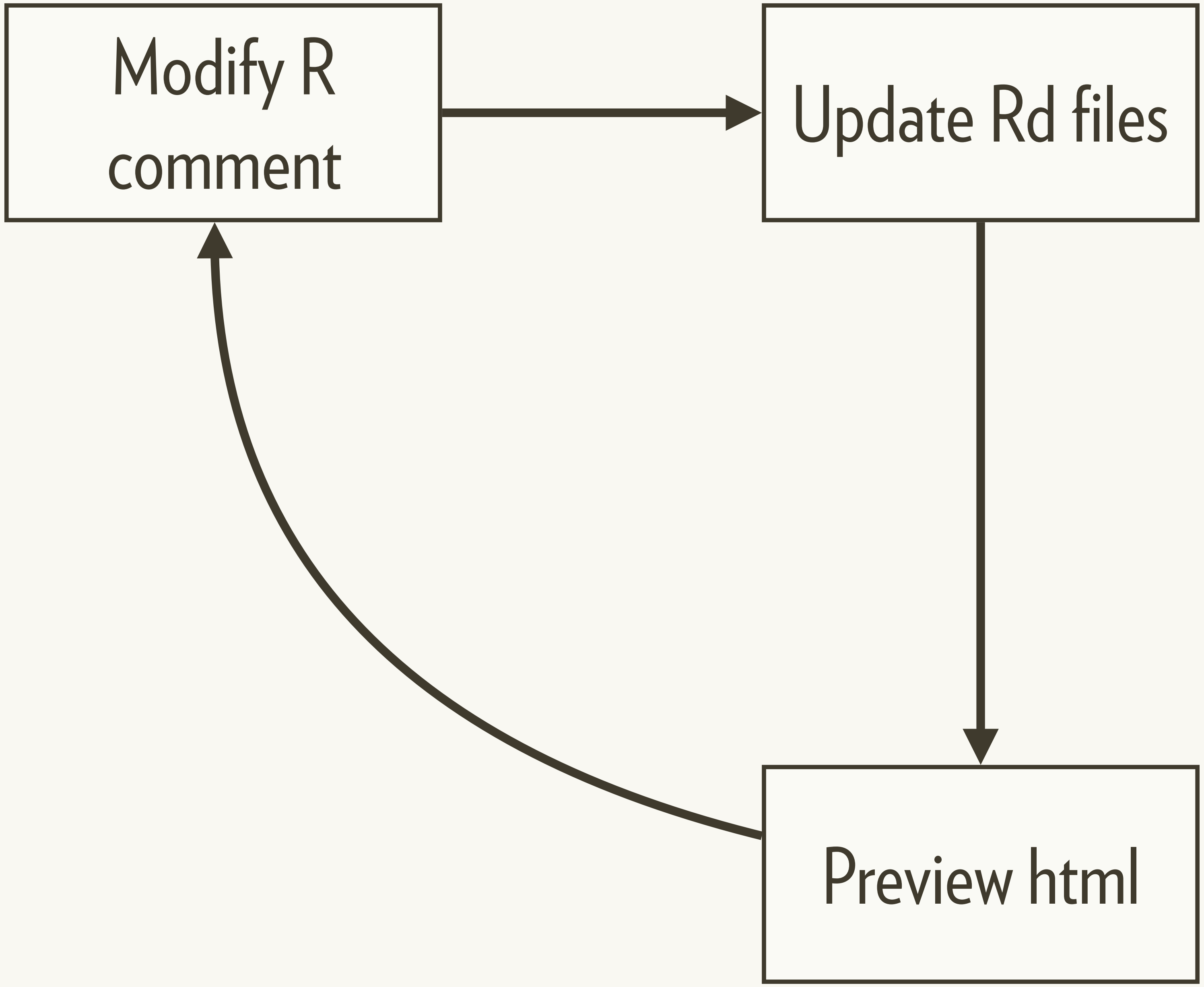
?topicname

Two caveats

1. You must have loaded the package with load_all() at least once.
   Check for message "Using development documentation…"

2. This technique only builds individual files so links do not work.

Documentation workflow



Modify R comment

Update Rd files

Preview html

Cmd/Ctrl + Shift + D

```
devtools::document()
```

NB: You must have loaded the package with load_all() at least once

?topicname

Only shows single file, so links do not work

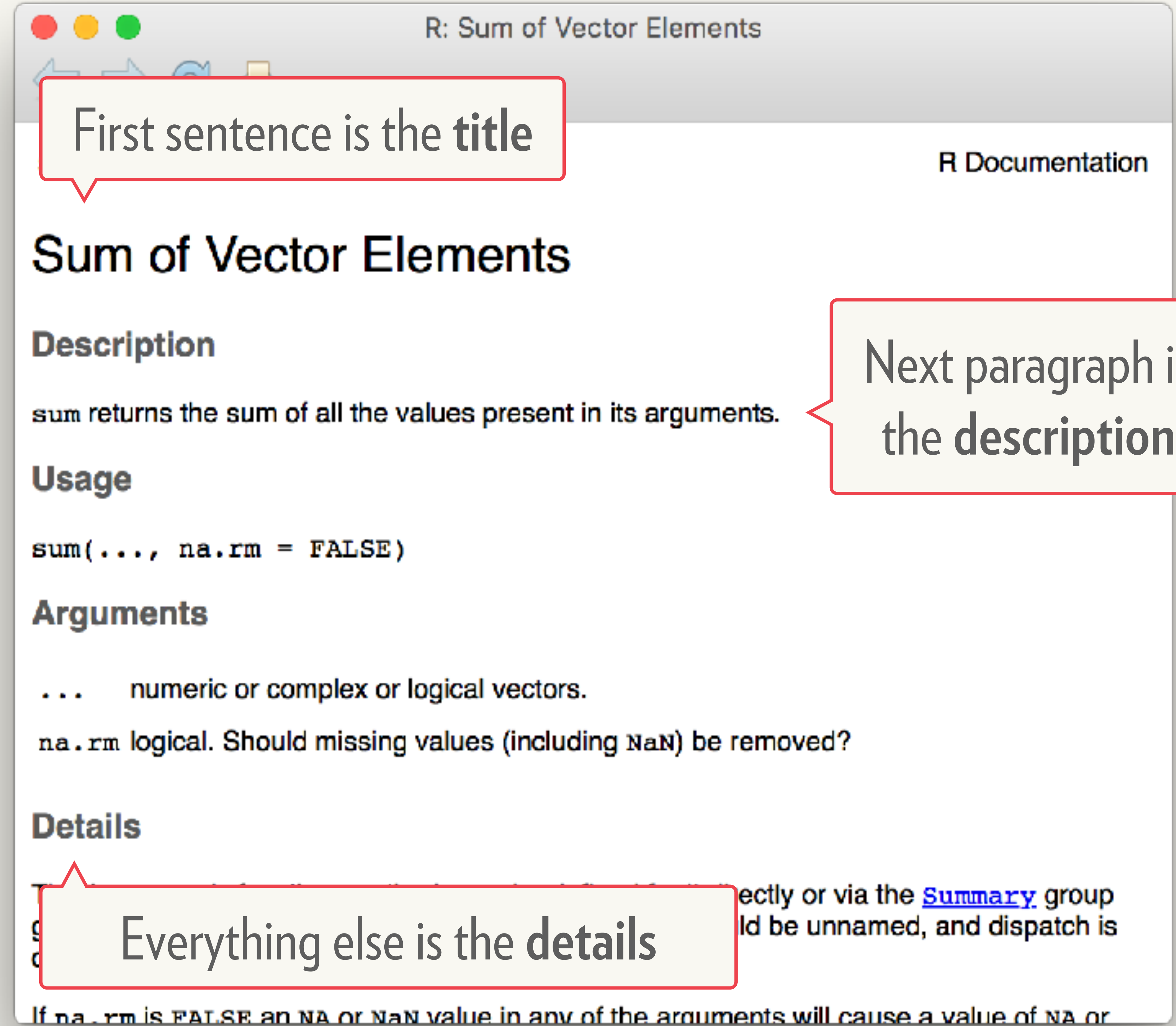
Change working directory/project to:

[hadcol]

Your turn

Fix the typos in the documentation for add_col.

Run the documentation workflow to check your work



R: Sum of Vector Elements

R Documentation

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

T ectly or via the Summary group g ld be unnamed, and dispatch is c

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

First sentence is the **title**

Next paragraph is the **description**

Everything else is the **details**

The description block

First sentence is the **title**

Next paragraph is the **description**

```
#' 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.
```

Everything else is the **details**

There are five tags you'll use for most functions

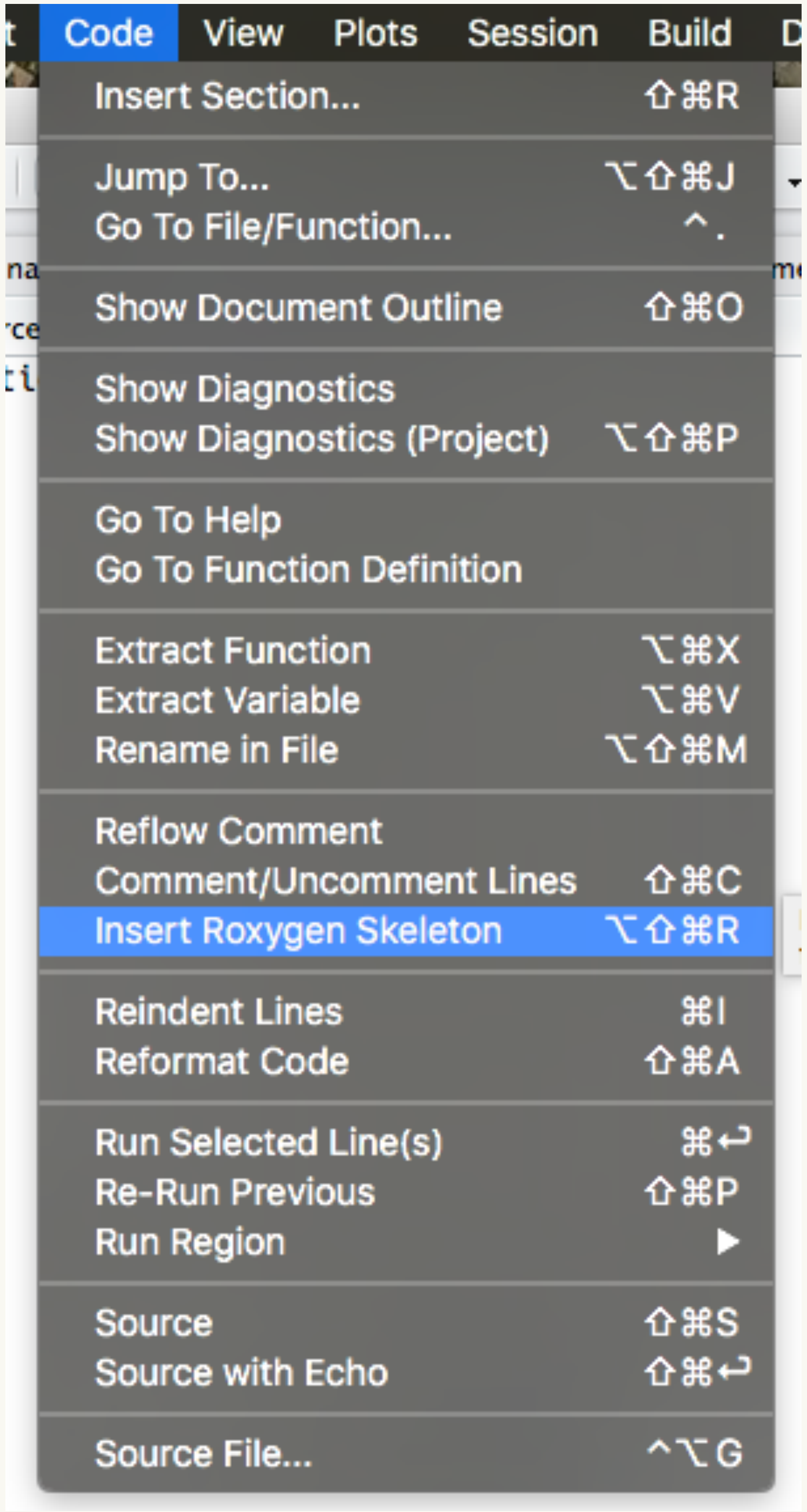
Tag	Purpose
@param arg	Describe inputs
@examples	Show how the function works. (Usual RStudio shortcuts work)
@seealso	Pointers to related functions
@return	Describe outputs (value)
@export	Is this a user-visible function?

Your turn

Document `add_cols()`.

(See next slide for hint)

RStudio helps you remember

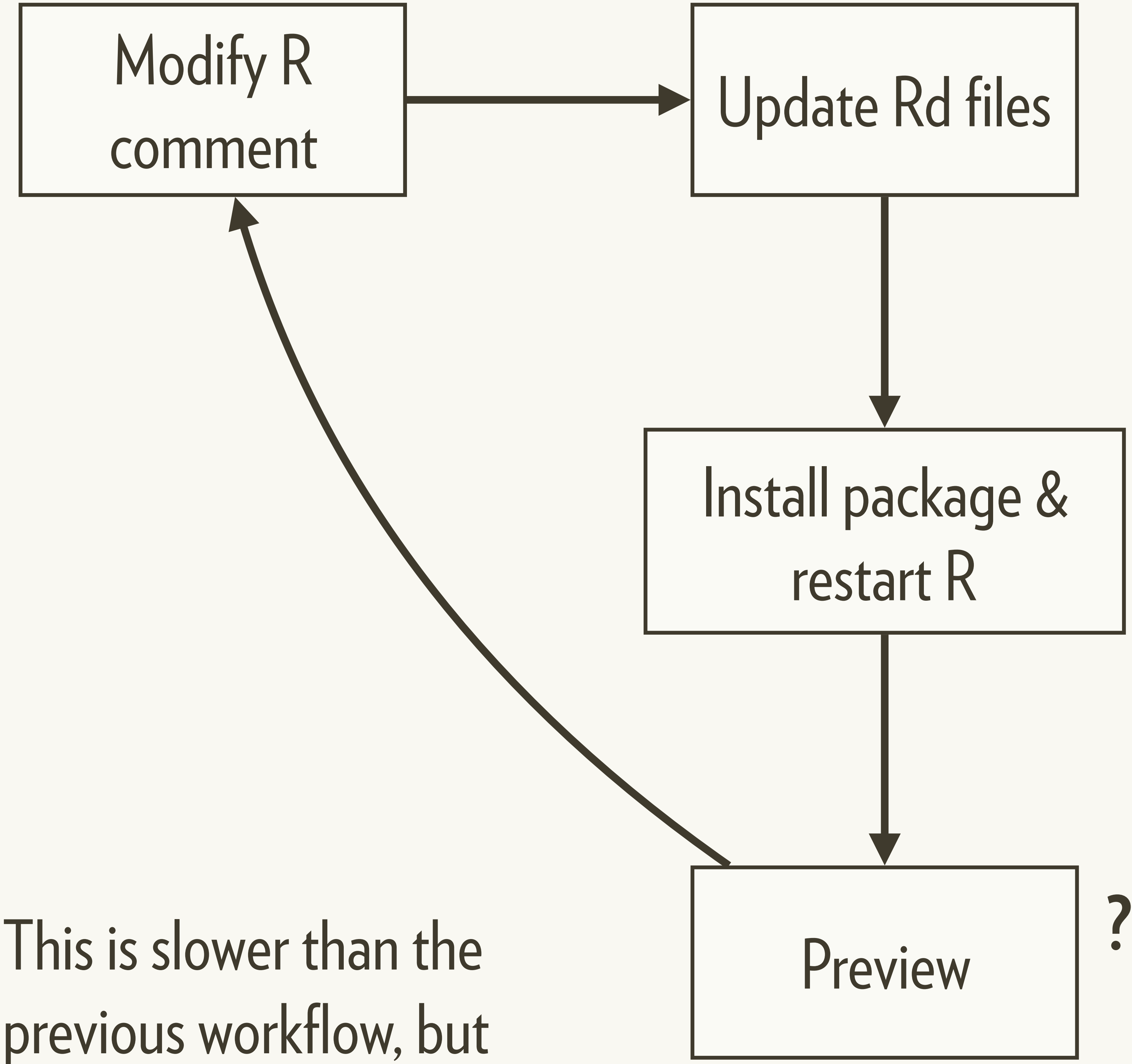


```r
#' Title
#'
#' @param x
#' @param y
#' @param z
#'
#' @return
#' @export
#'
#' @examples
fun <- function(x, y, z) {

}
```

Use markdown for formatting

```
# Activate by running
# use_roxygen_md()

**bold**, _italic_, `code`

* [func()]
* [pkg::func()]
* [link text][func()]
* [link text][pkg::func()]
```

Documentation workflow 2



Modify R comment

Update Rd files

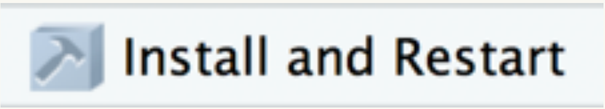
Install package & restart R

Preview

Cmd/Ctrl + Shift + D

```
devtools::document()
```

Cmd/Ctrl + Shift + B



?topicname

This is slower than the previous workflow, but there are fewer caveats

Your turn

Make real link to cbind()

Add a see also section (@seealso) to add_col() and add_cols() that links them together.

What happens if you add @family xyz to both?

roxygen2 comes with other tools to reduce duplication

```
# Document multiple functions in the same file
#' @rdname add_col

# Inherit the parameter descriptions from
# another function
#' @inheritParams add_col

# Inherit everything from another topic
#' @inherit add_col

# Inherit selectively
#' @inherit add_col parameters return references
#'   title description details
#'   sections seealso
```

Read online about how to document other objects

Data

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

Classes & methods

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

Packages

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

Namespace: imports

DESCRIPTION gives metadata about package

```
Package: colsum
Version: 0.0.0.9000
Title: What the Package Does (one line, title case)
Description: What the package does (one paragraph).
Authors@R: c(
    person("Hadley", "Wickham", , "hadley@rstudio.com"),
    person("RStudio", role = "cph")
  )
License: GPL-3
Encoding: UTF-8
LazyData: true
ByteCompile: true
Depends:
    purrr
Suggests:
    testthat
```

Depends is not ok for CRAN packages

```
Package: colsum
Version: 0.0.0.9000
Title: What the Package Does (one line, title case)
Description: What the package does (one paragraph).
Authors@R: c(
    person("Hadley", "Wickham", , "hadley@rstudio.com"),
    person("RStudio", role = "cph")
  )
License: GPL-3
Encoding: UTF-8
LazyData: true
ByteCompile: true
Depends:
    purrr
Suggests:
    testthat
```

Instead, need to use imports

```
Package: colsum
Version: 0.0.0.9000
Title: What the Package Does (one line, title case)
Description: What the package does (one paragraph).
Authors@R: c(
    person("Hadley", "Wickham", , "hadley@rstudio.com"),
    person("RStudio", role = "cph")
  )
License: GPL-3
Encoding: UTF-8
LazyData: true
ByteCompile: true
Imports:
    purrr
Suggests:
    testthat
```

But that requires we use ::

```
# Or you might want to use an infix function
`%>%` <- magittr::`%>%`

col_summary <- function(df, fun) {
  stopifnot(is.data.frame(df))

  df %>%
    purrr::keep(is.numeric) %>%
    purrr::modify(fun)
}
```

Instead can **import** functions into the package

```
#' @importFrom purrr keep modify
#' @importFrom magrittr %>%
col_summary <- function(df, fun) {
  stopifnot(is.data.frame(df))

  df %>%
    keep(is.numeric) %>%
    modify(fun)
}
```

Alternatively, create R/imports.R

```
# Imports belong to the package, not to
# individual functions, so you might want
# to recognise this by storing in a central
# location

#' @importFrom purrr keep map
#' @importFrom magrittr %>%
NULL
```

Importing everything from a package seems easy

```
#' @import purrr
col_summary <- function(df, fun) {
  stopifnot(is.data.frame(df))

  df %>%
    keep(is.numeric) %>%
    map_dfc(fun)
}
```

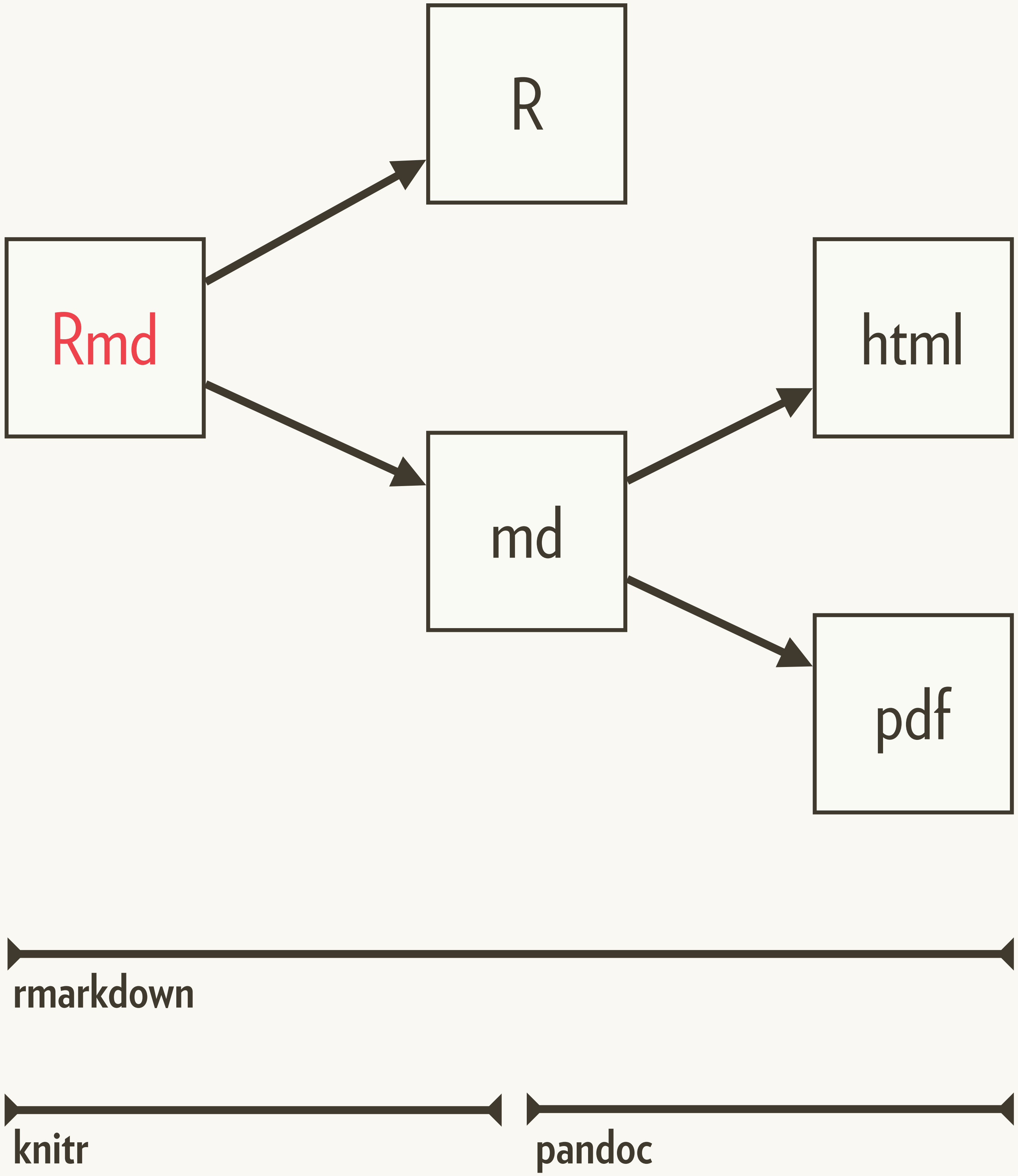
But is dangerous...

```
#' @import foo
#' @import bar
fun <- function(x) {
  fun1(x) + fun2(x)
}

# Works today
# But next year, bar package adds fun1 function
```

Description	NAMESPACE
Makes **package** available	Makes **function** available
Mandatory	Optional (can use :: instead)
use_package()	#’ @importFrom

Package documentation with rmarkdown

Vignettes



Lets you combine prose and code to explain your how you package works.

The hard part is the writing, not the technology!

http://r-pkgs.had.co.nz/vignettes.html

Easiest way to get started is with use_vignette()

```
usethis::use_vignette("name")

# Adds to DESCRIPTION
Suggests: knitr
VignetteBuilder: knitr

# Creates vignettes/
# Drafts vignettes/name.Rmd
```

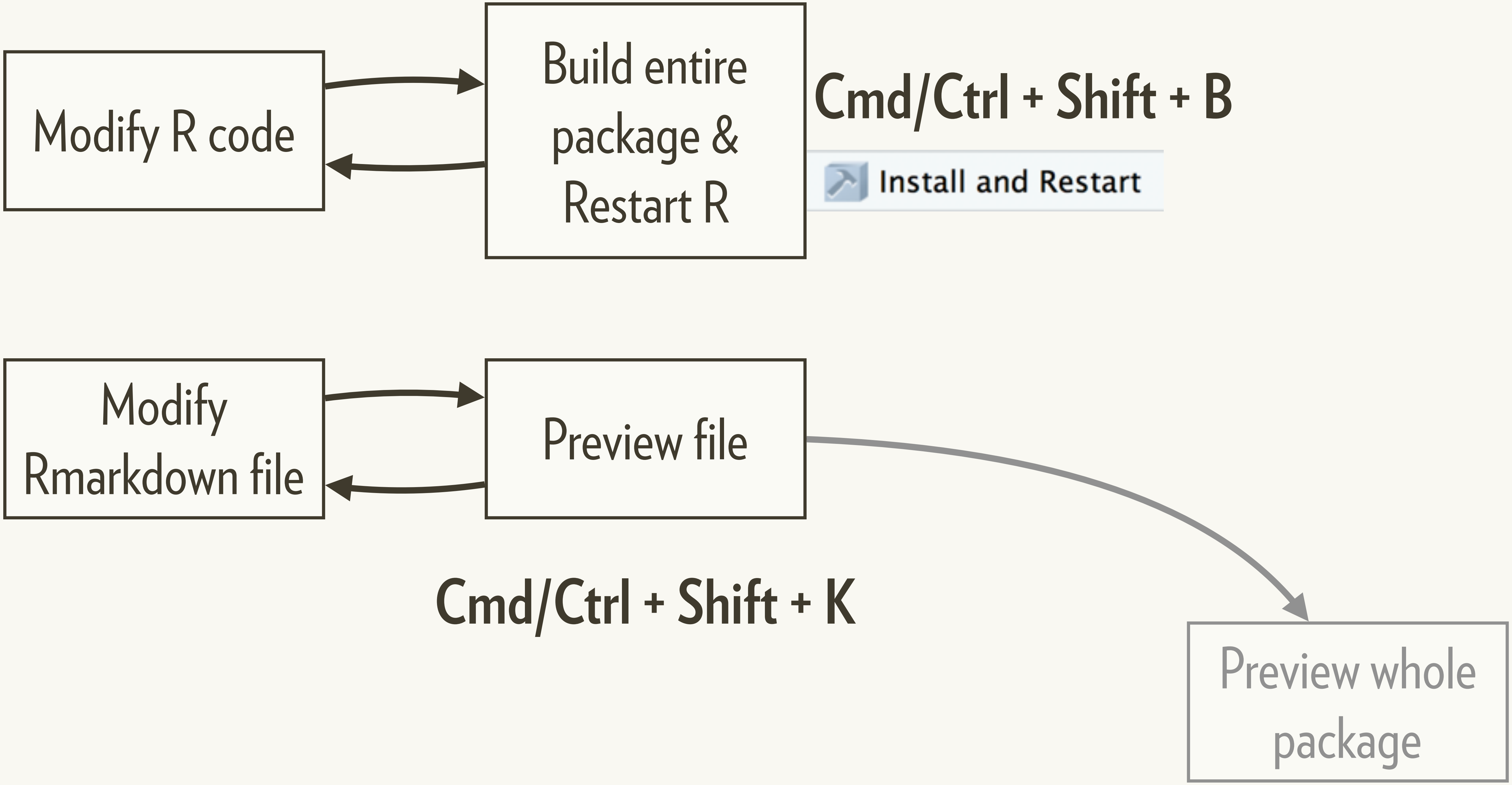
Vignette = Rmarkdown + special metadata

```
---
title: "Vignette Title"
author: "Vignette author"
date: "`r Sys.Date()`"
output: rmarkdown::html_vignette
vignette: >
  %\VignetteIndexEntry{Vignette Title}
  %\VignetteEngine{knitr::rmarkdown}
  %\VignetteEncoding{UTF-8}
---

Vignettes are long form documentation commonly included in packages.
Because they are part of the distribution of the package, they need to
be as compact as possible. The `html_vignette` output type provides a
custom style sheet (and tweaks some options) to ensure that the
resulting html is as small as possible. The `html_vignette` format:

...
```

Special output format for vignettes

Special metadata needed by R

Vignette workflow



Modify R code

Build entire package & Restart R

Cmd/Ctrl + Shift + B

Modify Rmarkdown file

Preview file

Cmd/Ctrl + Shift + K

Preview whole package

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

If sharing with others, include a README

```r
# Your choice: but often useful to include
# results of running code
usethis::use_readme_md()
usethis::use_readme_rmd()

# For public projects this should include a
# brief overview, instructions on how to
# install, and a few examples. For private
# projects, this is a great place to jot down
# notes
```

If evolving over time, note changes to API

```
usethis::use_news_md()

# For public projects, this should note
# important changes from perspective of
# the user. Most important to describe API
# changes. Less important for private projects
```