

Documentation

Jan 2019

Charlotte Wickham

@cvwickham

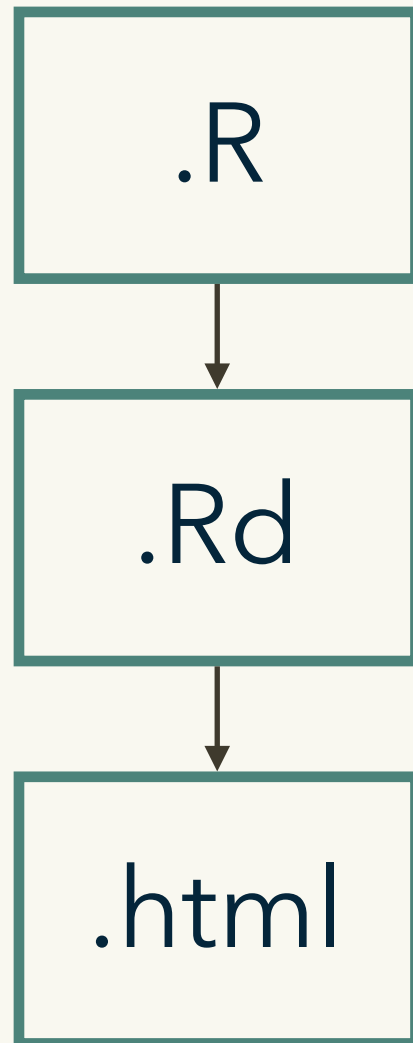
cwickham@gmail.com

cwick.co.nz

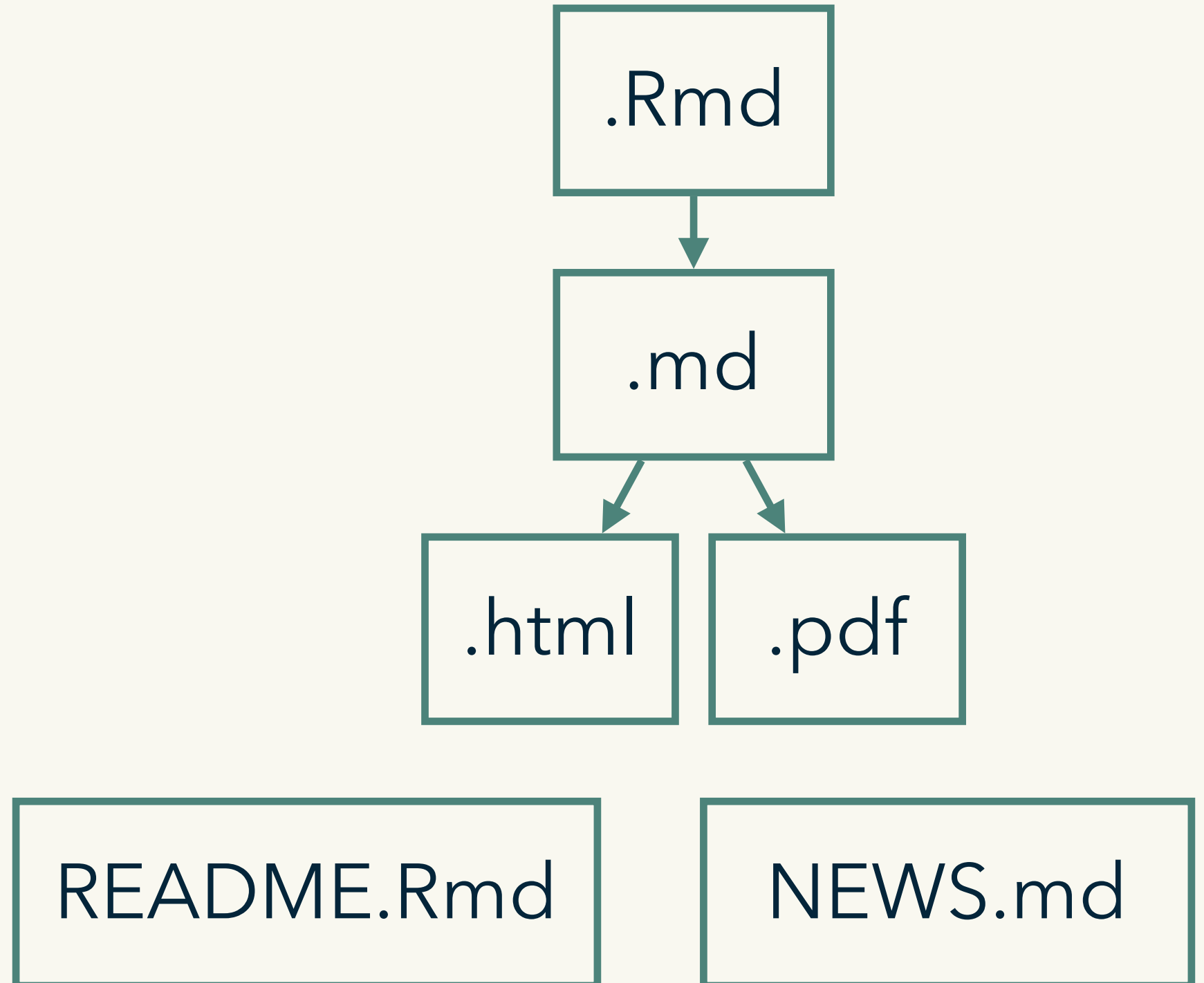
Adapted from *Tidy Tools* by Hadley Wickham



Function-level with **roxygen2**



Package-level with **rmarkdown**



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
```

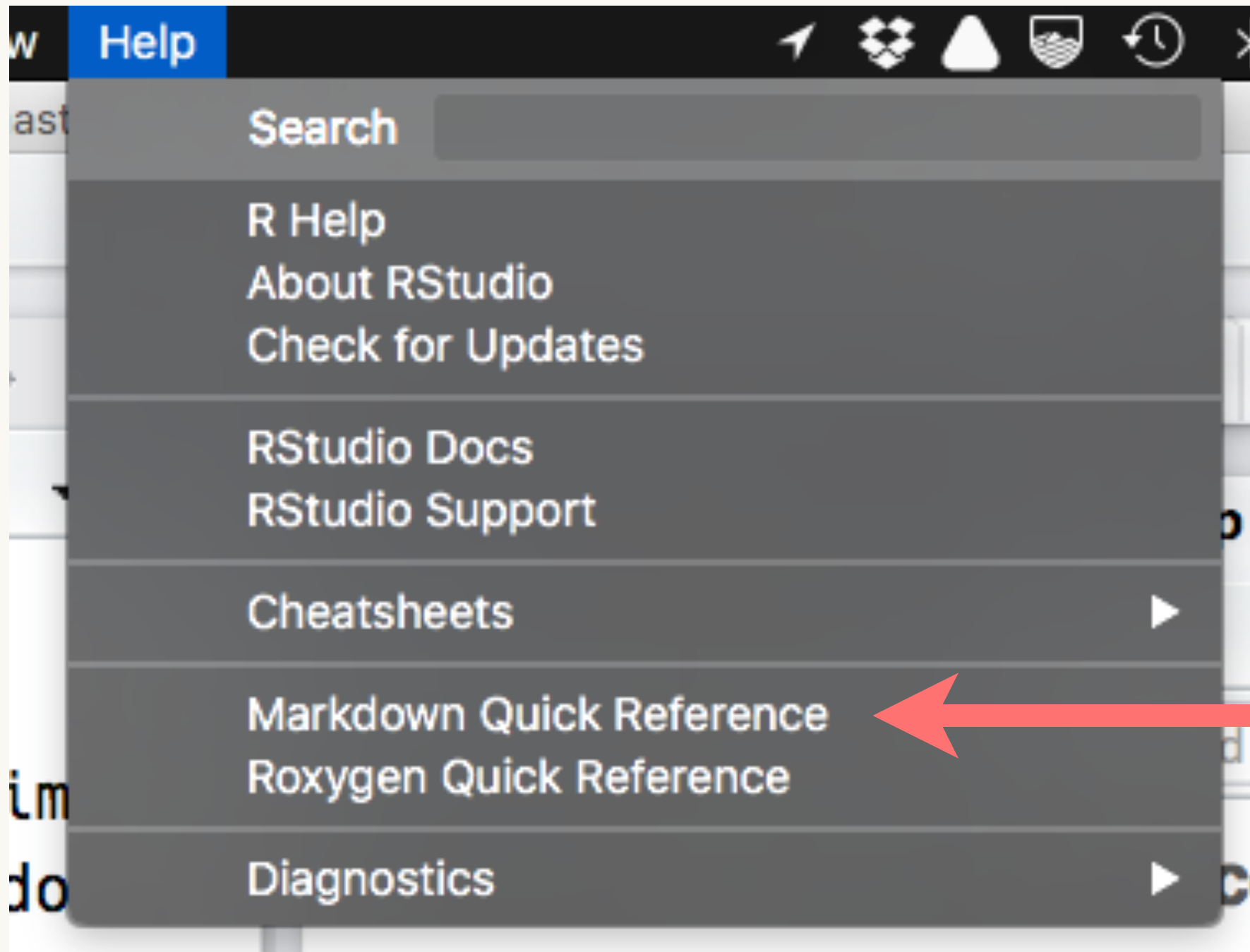
```
# Some R code
```

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

```
` `` `
```

This is a secondary heading

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



Function documentation with roxygen2

Roxxygen2

roxxygen2

R



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

You write specially formatted comments in .R

In 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)
```


You write specially formatted comments in `.R`

Roxygen comment

```
#' 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)
```

Roxygen tag

Roxygen translates to **.Rd**

In `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.
}
```

*In almost all cases
you can ignore
these files*

```
add_col {hadcol}
```

R Documentation

R translates to
[.html](#) for viewing

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

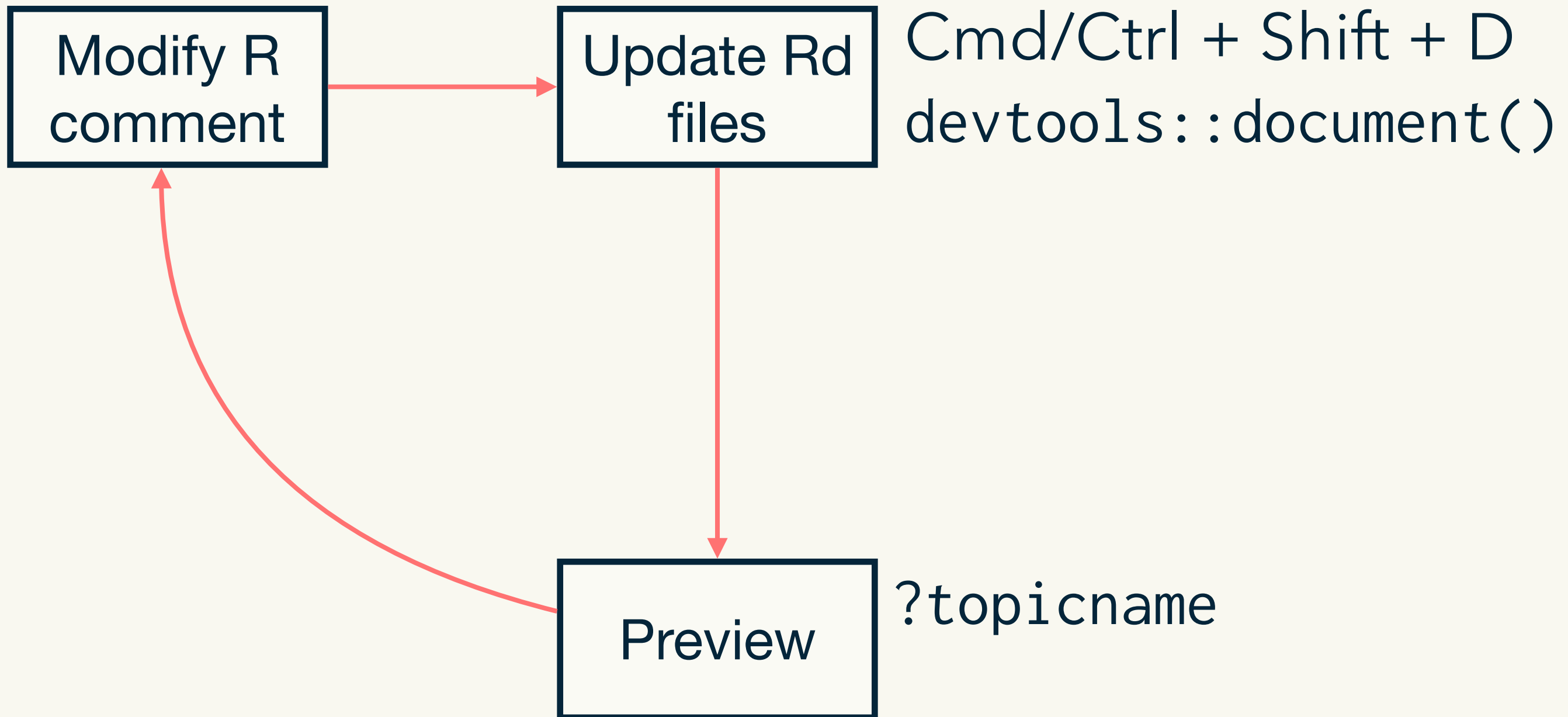
The columns in

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)
```

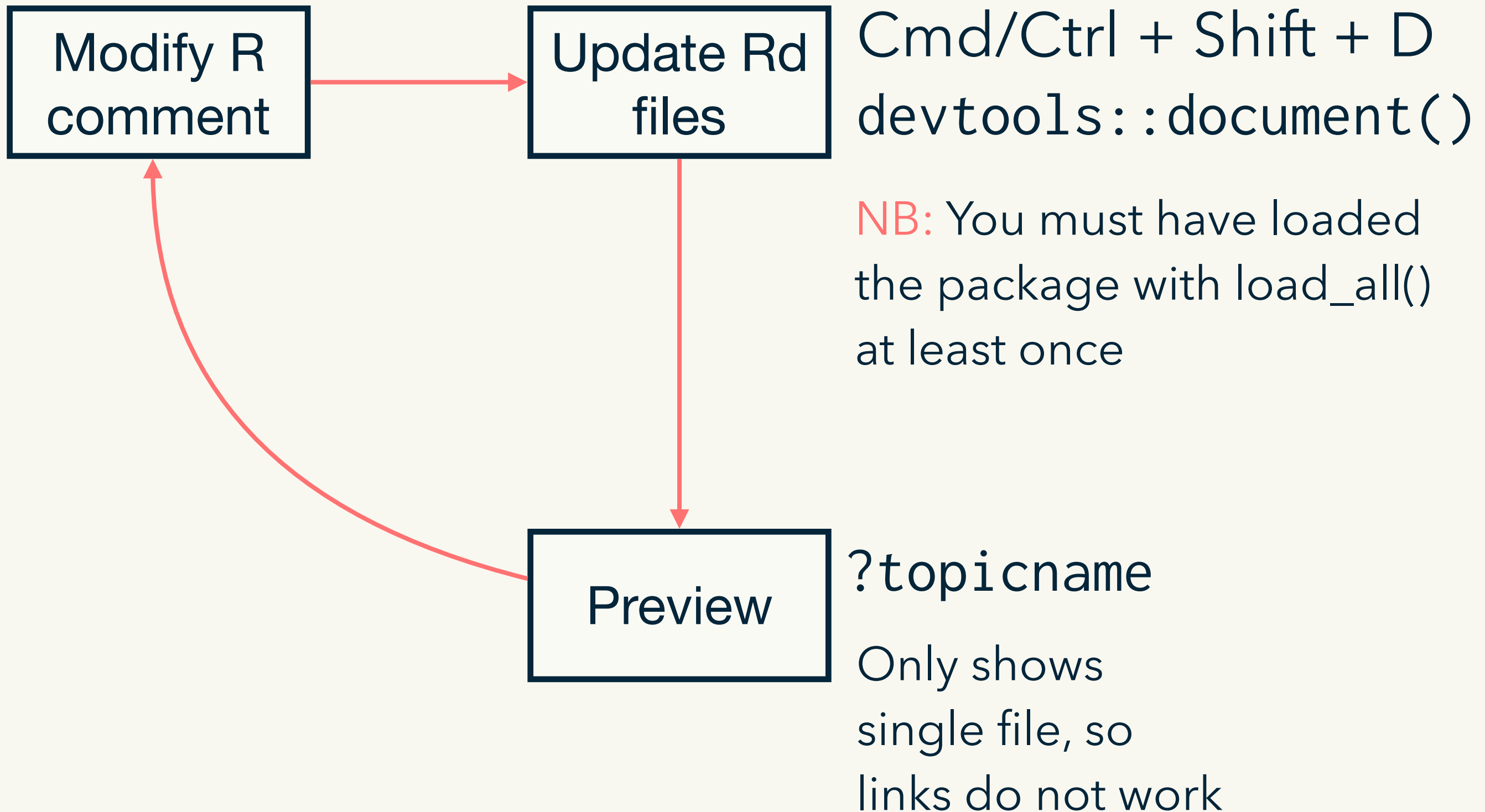
Documentation workflow



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



Change project to:

[hadcol-doc]

Your turn

Fix the typos in the documentation for `add_col()`.

Run the documentation workflow to check your work.

R: Sum of Vector Elements

sum {base}

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

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

First sentence is the **title**

Next paragraph is the **description**

Everything else is the **details**

The description block

First sentence is the **title**

```
#' 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.
```

Next paragraph is
the **description**

Everything else is the
details

There are five **tags** you'll use for most functions

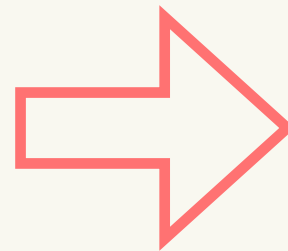
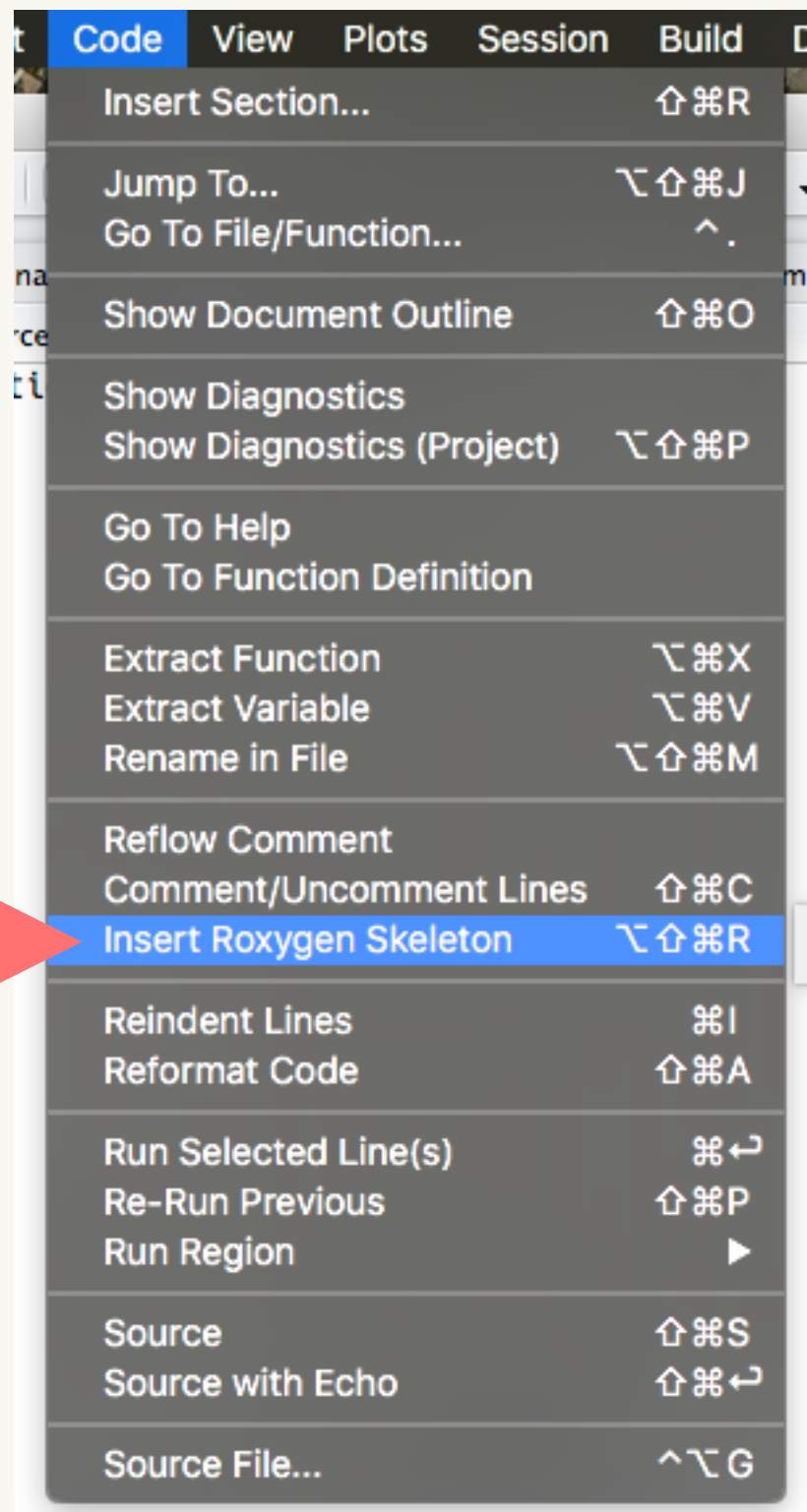
Tag	Purpose
@param arg	Describe inputs
@examples	Show how the function works. (Usual RStudio shortcuts work)
@seealso	Pointers to related functions
@return	Describe outputs (value)
@export	Is this a user-visible function?

Your turn

Document `insert_into()`.

(See next slide for hint)

RStudio helps you remember



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

Use markdown for formatting

```
# In new package, activate by running  
# usethis::use_roxygen_md()
```

****bold****, *_italic_*, ``code``

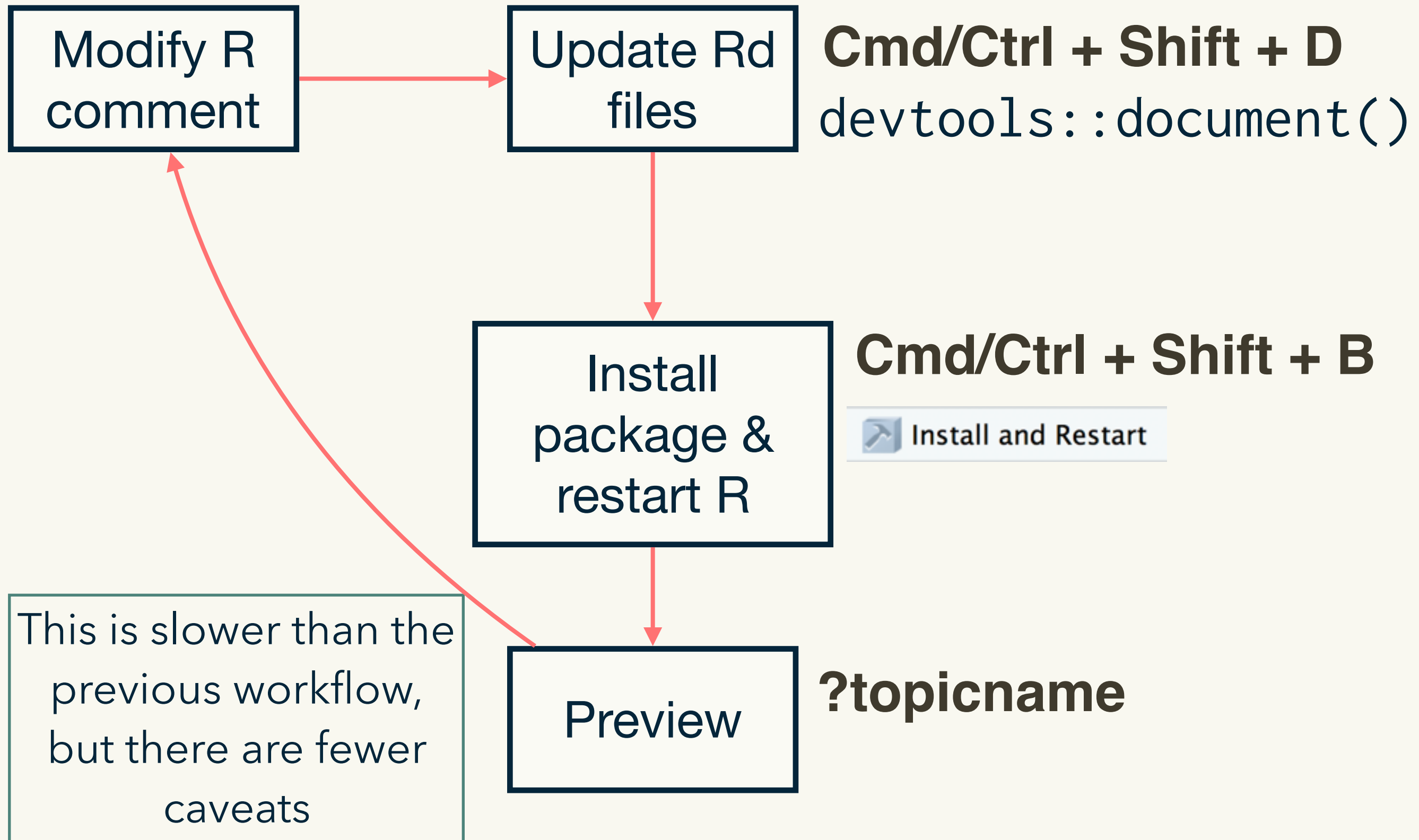
* [func()]

* [pkg::func()]

* [link text][func()]

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

Documentation workflow 2



Your turn

Make real link to `cbind()`

Add a see also section
(`@seealso`) to `add_col()` and
`insert_into()` 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

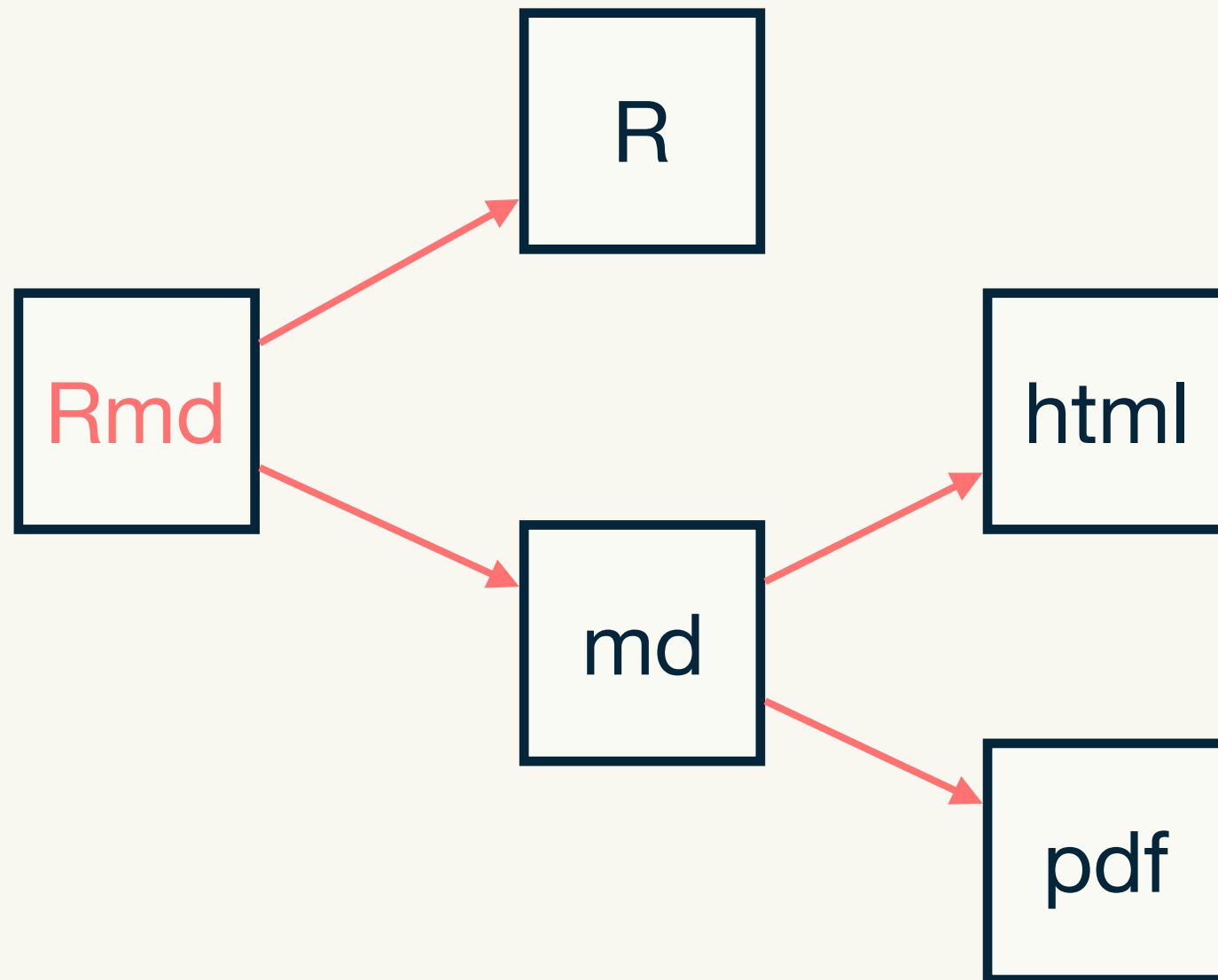
<http://r-pkgs.had.co.nz/man.html#man-classes>

Packages

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

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!

rmarkdown

knitr

pandoc

<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: >
```

Special output format
for vignettes

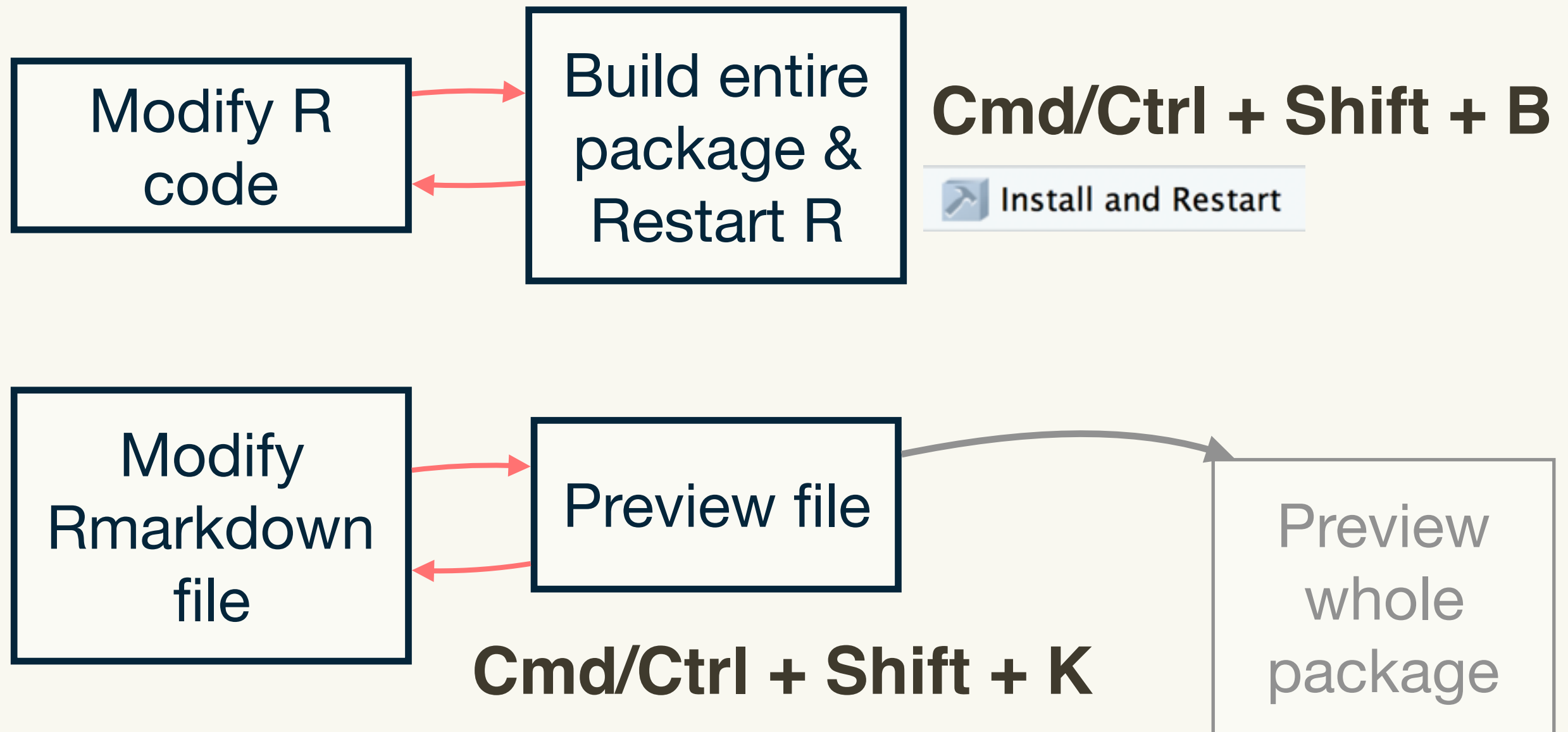
```
%\VignetteIndexEntry{Vignette Title}
%\VignetteEngine{knitr::rmarkdown}
%\VignetteEncoding{UTF-8}
```

Special metadata
needed by R

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:

...

Vignette workflow



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

```
browseVignettes()
```

If sharing with others, include a README

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
```


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/)