

# Document and share

**January 2020**

Charlotte Wickham

@cvwickham

Adapted from *Tidy Tools* by Hadley Wickham



# Overview

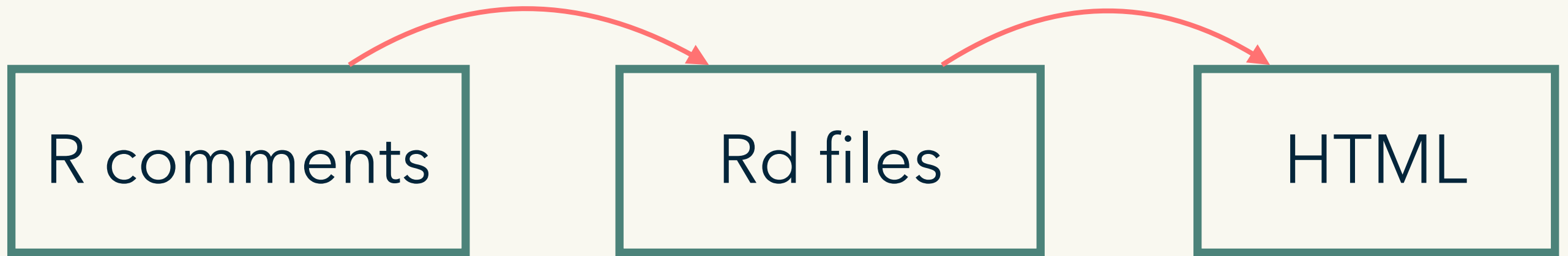
1. Function docs
2. Other docs
3. R CMD check
4. Continuous integration
5. CRAN submission

Function docs 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))
```

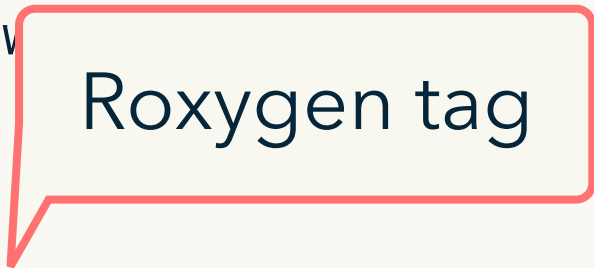
```
#' add_col(df, "y", runif(5), where = 1)
```

```
#'
```

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

You write specialy formatted comments in `.R`

## Roxygen comment

```
#' Add a column to a data frame
#
#
# Allow the position. Will replace existing variable
# with  Roxygen tag 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 almost all cases  
you can ignore  
these files*

```
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.
```

```
}
```

```
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)
```



**Easy(er)**

R comments



Rd files

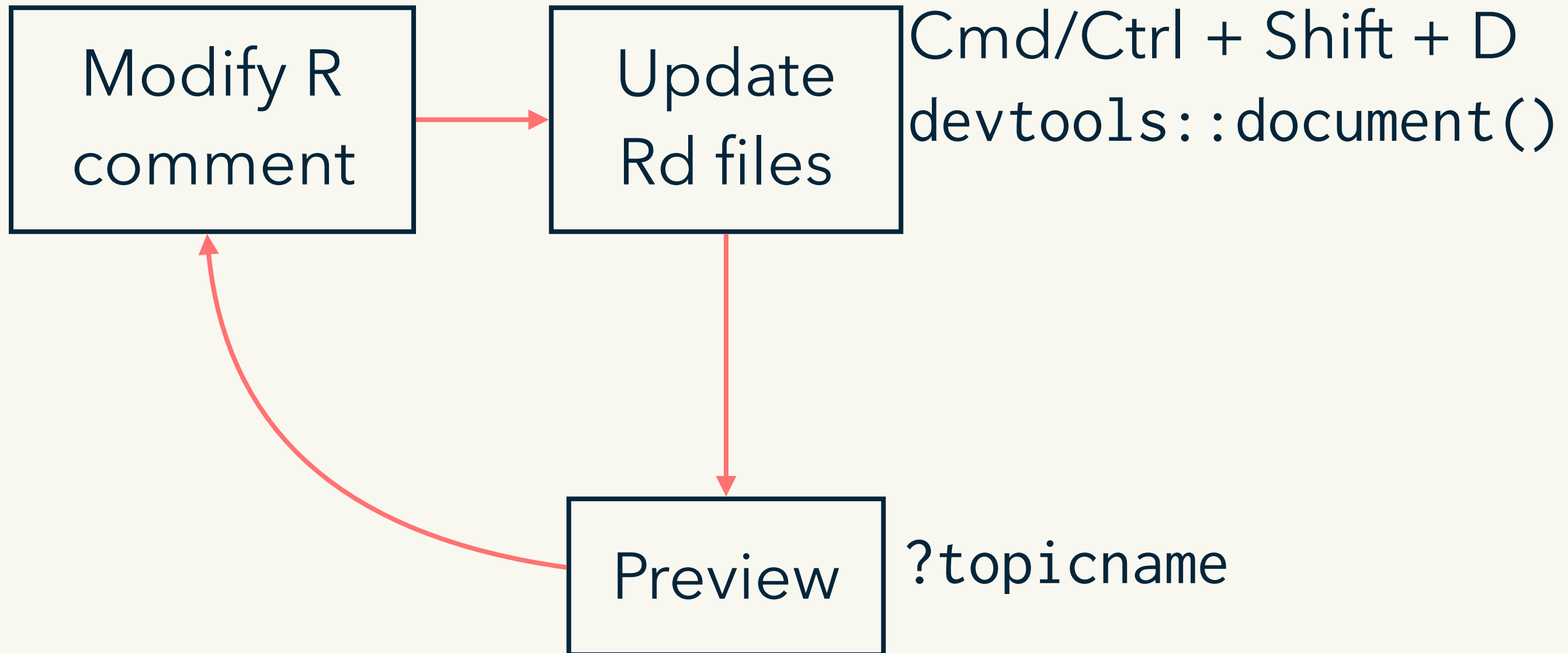


HTML

**Hard**

Writing text that  
other humans can  
understand

# 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.

Change project to:

[fordogs]

```
create_from_github("hadley/fordogs", fork = FALSE)
```

# Your turn

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

Run the documentation workflow to check your work.

Get the package with:

```
usethis::create_from_github("hadley/fordogs",  
fork = FALSE, protocol = "https")
```

R: Sum of Vector Elements

First sentence is the **title**

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

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
```

```
#'
```

```
#' `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 [Summary] group generic. For this to work properly,
```

```
#' the arguments `...` 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
@returns	Describe outputs (value)
@export	Is this a user-visible function?

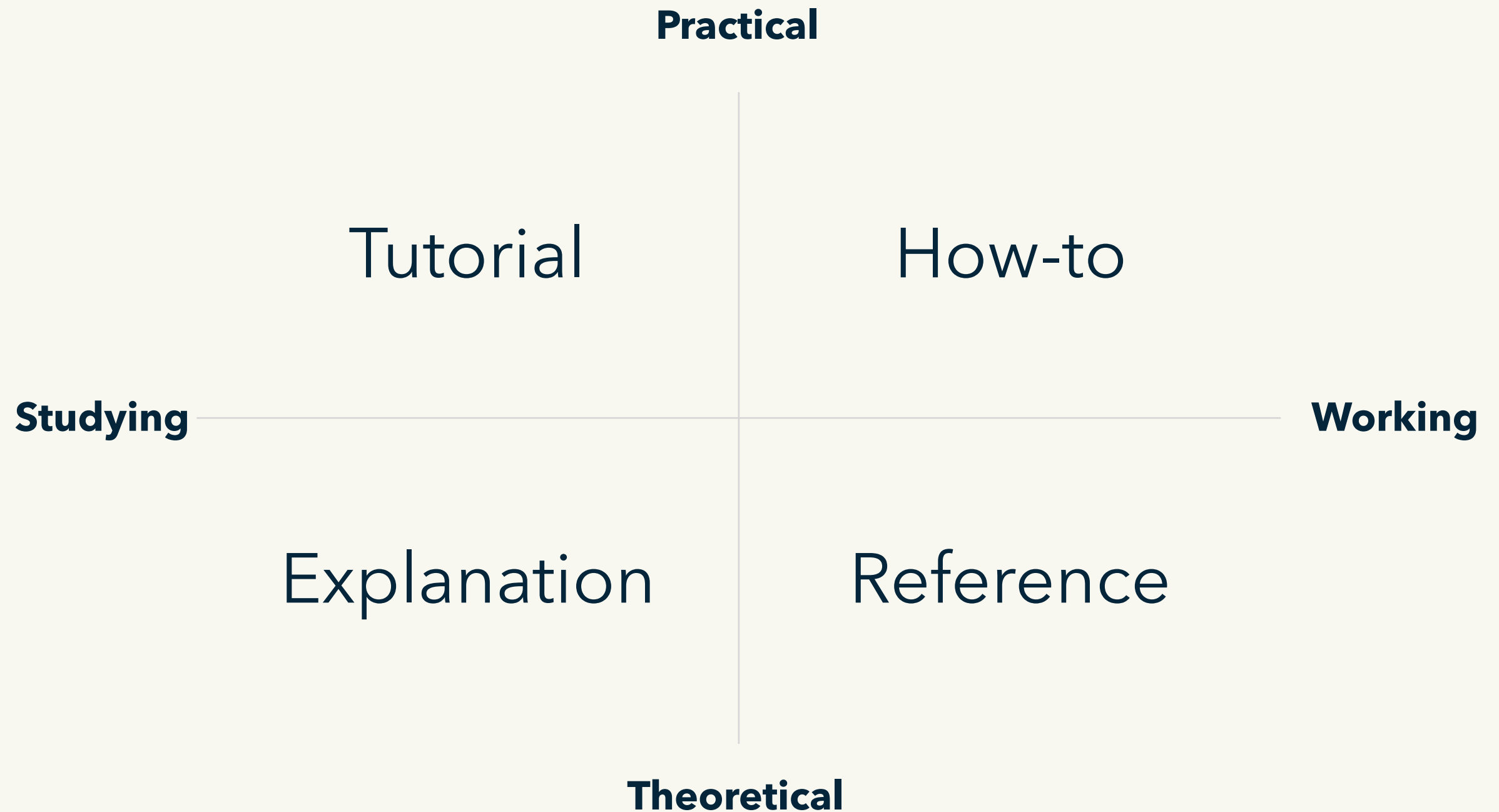
<https://roxygen2.r-lib.org/articles/rd.html>

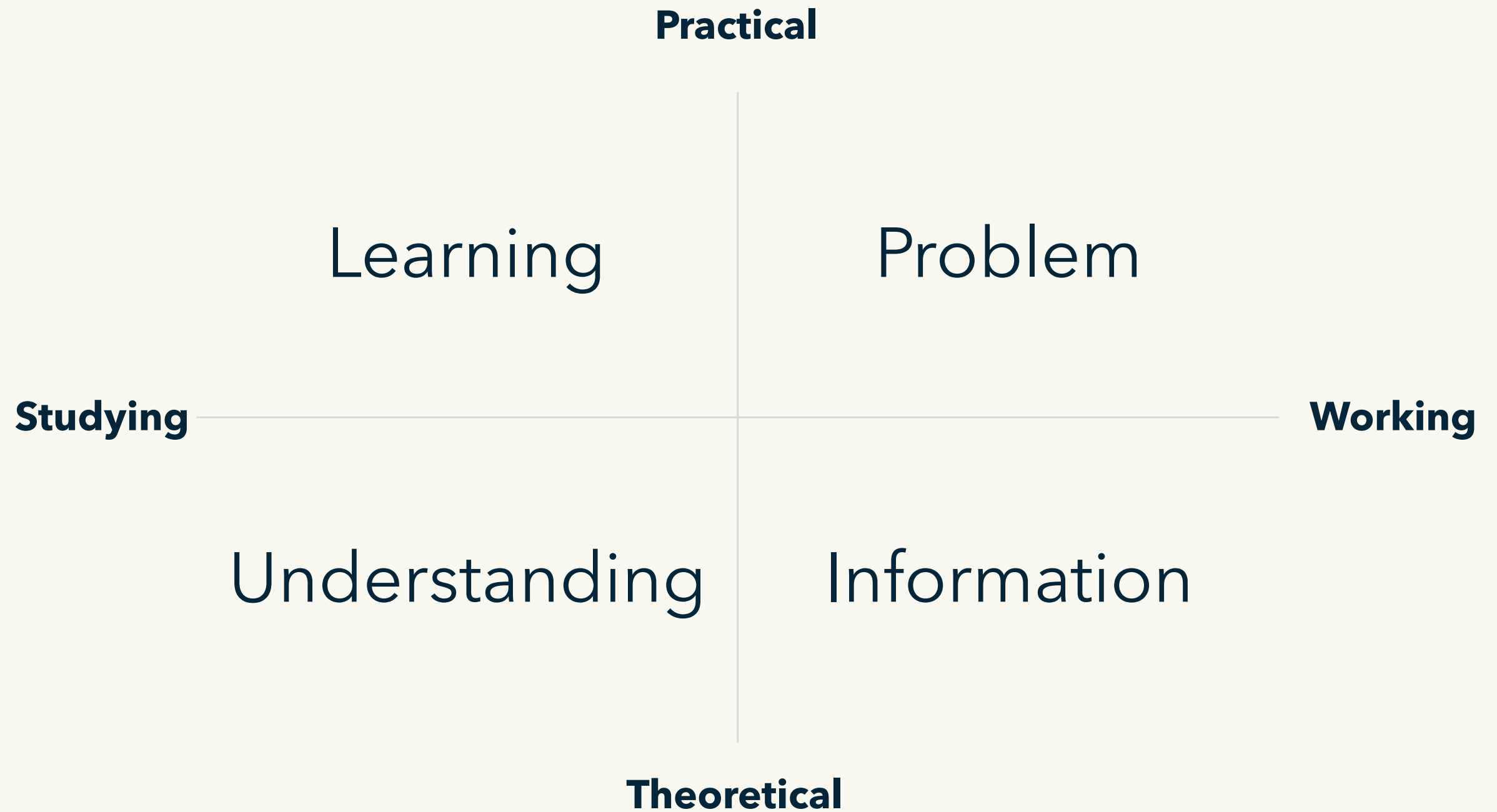


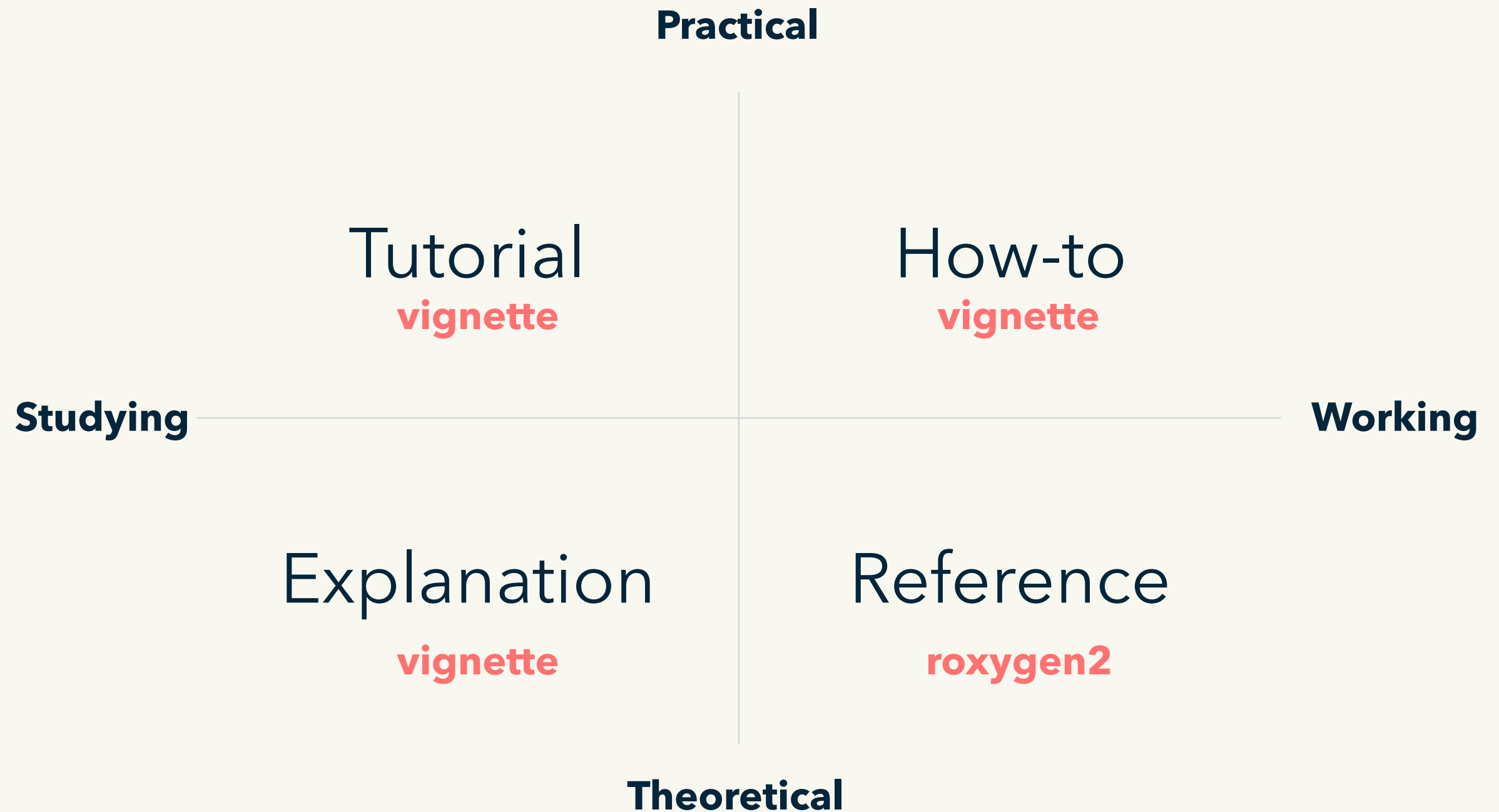
# Your turn

Complete the documentation for `fbind()`.

A good minimum is to describe the types of the input and output (we'll talk a lot more about types tomorrow).

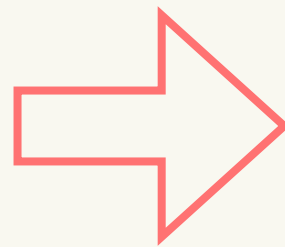
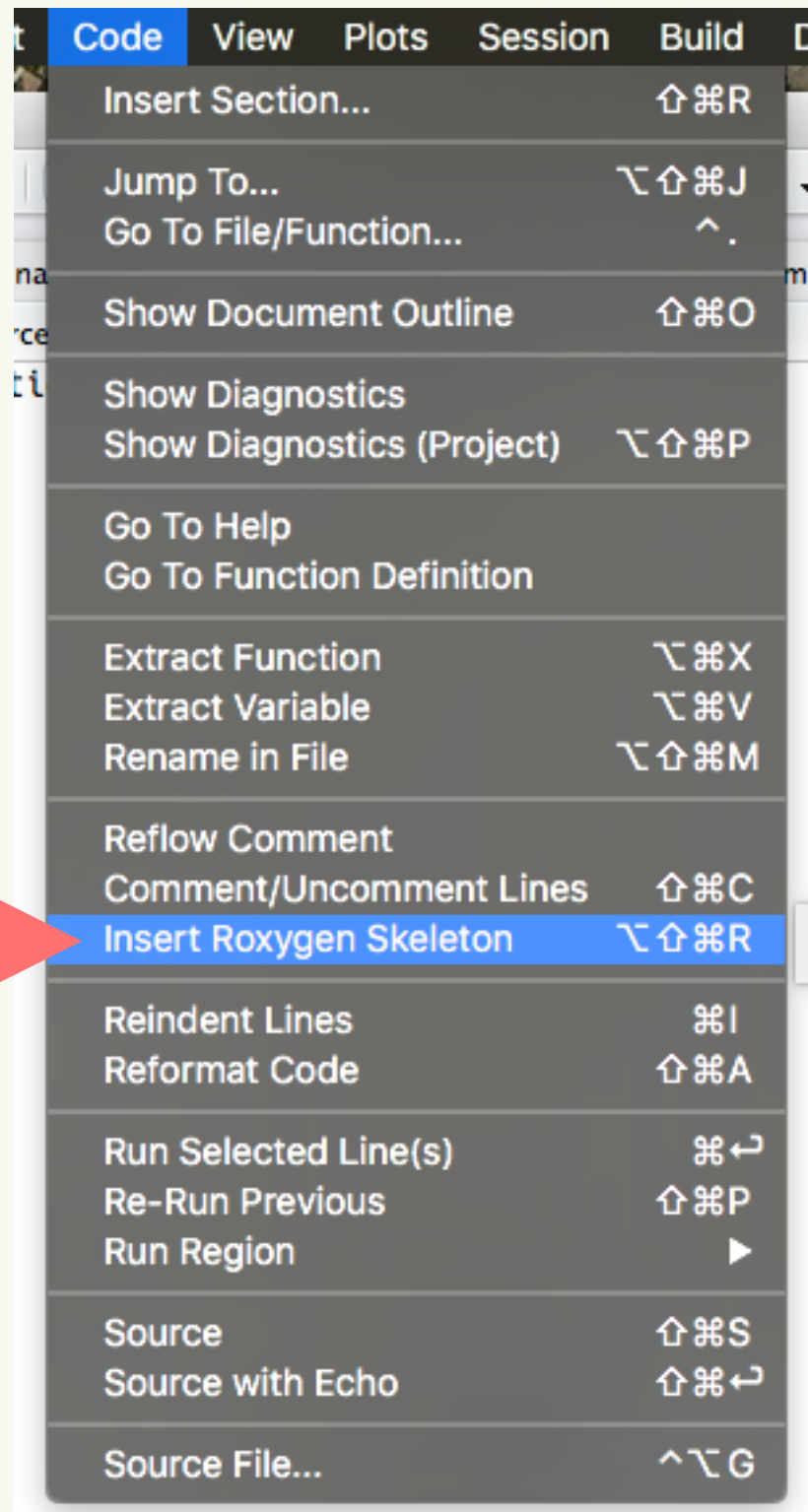






<https://www.divio.com/blog/documentation/>

# RStudio helps you remember



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

# Your turn

Document `fdist()`.

```
fdist(factor(rpois(50, 5)))
```

```
fdist(factor(rpois(500, 5)))
```

```
fdist(factor(round(rnorm(100, 5))))
```

```
fdist(ggplot2::diamonds$cut)
```

```
fdist(ggplot2::diamonds$cut, sort = TRUE)
```

# Use markdown for formatting

```
# In fordogs package, activate by running  
# usethis::use_roxygen_md()  
# (Activated by default in new packages)
```

***code***

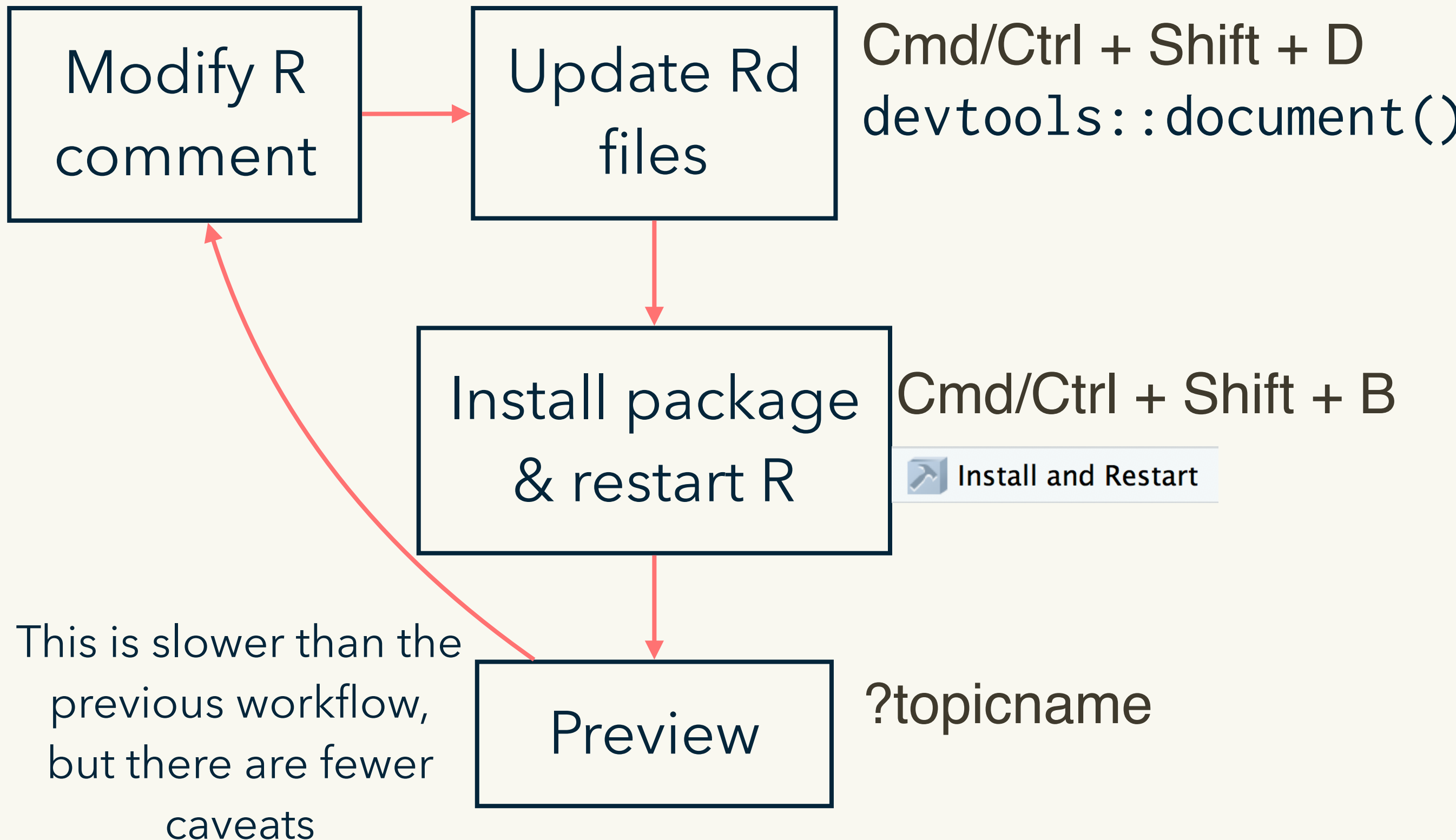
\* [func()]

\* [pkg::func()]

\* [link text][func()]

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

# Documentation workflow 2





# Package documentation with rmarkdown

# Use vignettes for broader topics

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

```
# Adds to DESCRIPTION
```

```
Suggests: knitr
```

```
VignetteBuilder: knitr
```

```
# Creates vignettes/
```

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

# Vignette = Rmarkdown + special metadata

---

```
title: "Vignette Title"  
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:

...

# Big picture in README

```
usethis::use_readme_rmd()
```

```
# * 2-3 paragraph overview
```

```
# * Installation instructions
```

```
# * Usage example, with pointer to vignettes
```

If evolving over time, note changes to API

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

```
# * what's new
```

```
# * what's changed
```

```
# * what's gone away
```

```
#
```

```
# More info at
```

```
# http://style.tidyverse.org/news.html
```

# Turn into a website with pkgdown



```
use_pkgdown()  
use_pkgdown_travis()
```

# Continuous integration (CI)

`use_git()`

`use_github()`

`use_travis()`

`use_coverage()`

`use_pkgdown_travis()`

`use_github_actions()` # dev version



R CMD check

# Automated checking

Runs automated checks for common problems in R packages.

Useful for local packages, even with some false positives.

If you want to submit to CRAN, you **must** pass R CMD check cleanly.

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



To avoid frustration run  
early and run often

“If it hurts,  
do it more often”

— Martin Fowler

==> R CMD build rv2

- \* checking for file 'rv2/DESCRIPTION' ... OK
- \* preparing 'rv2':
- \* checking DESCRIPTION meta-information ... OK
- \* installing the package to build vignettes
- \* creating vignettes ... OK
- \* checking for LF line-endings in source and make files
- \* checking for empty or unneeded directories
- \* building 'rv2\_0.1.tar.gz'

==> R CMD check rv2\_0.1.tar.gz

- \* using log directory '/Users/hadley/Documents/courses/13-devtools/rv2.Rcheck'
- \* using R version 3.0.2 (2013-09-25)
- \* using platform: x86\_64-apple-darwin10.8.0 (64-bit)
- \* using session charset: UTF-8
- \* checking for file 'rv2/DESCRIPTION' ... OK
- \* this is package 'rv2' version '0.1'
- \* checking package namespace information ... OK
- \* checking package dependencies ... OK
- \* checking if this is a source package ... OK
- \* checking if there is a namespace ... OK
- \* checking for executable files ... OK
- \* checking for hidden files and directories ... OK
- \* checking for portable file names ... OK

- \* checking for sufficient/correct file permissions ... OK
- \* checking whether package 'rv2' can be installed ... OK\* checking installed package size ... OK
- \* checking package directory ... OK
- \* checking DESCRIPTION meta-information ... OK
- \* checking top-level files ... OK
- \* checking for left-over files ... OK
- \* checking index information ... OK
- \* checking package subdirectories ... OK
- \* checking R files for non-ASCII characters ... OK
- \* checking R files for syntax errors ... OK
- \* checking whether the package can be loaded ... OK
- \* checking whether the package can be loaded with stated dependencies ... OK
- \* checking whether the package can be unloaded cleanly ... OK
- \* checking whether the namespace can be loaded with stated dependencies ... OK
- \* checking whether the namespace can be unloaded cleanly ... OK
- \* checking loading without being on the library search path ... OK
- \* 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 ... OK
- \* checking Rd metadata ... OK
- \* checking Rd cross-references ... OK
- \* checking for missing documentation entries ... OK
- \* checking for code/documentation mismatches ... OK

- \* checking Rd \usage sections ... OK
- \* checking Rd contents ... OK
- \* checking for unstated dependencies in examples ... OK
- \* checking installed files from 'inst/doc' ... OK
- \* checking files in 'vignettes' ... OK
- \* checking examples ... OK
- \* checking for unstated dependencies in tests ... OK
- \* checking tests ...
  - Running 'testthat.R' OK
- \* checking for unstated dependencies in vignettes ...  
OK
- \* checking package vignettes in 'inst/doc' ... OK
- \* checking running R code from vignettes ...
  - 'c1t.Rmd' ... OK
- OK
- \* checking re-building of vignette outputs ... OK
- \* checking PDF version of manual ... OK

R CMD check succeeded

# Types of problem

## ERROR

Must fix!

## WARNING

Fix if submitting to CRAN

## NOTE

Fix if submitting to CRAN

*It is possible to submit with a NOTE, but it's best avoided*

	Local	CRAN
ERROR	✓	✓
WARNING		✓
NOTE		✓



# Cmd/Ctrl + Shift + E

devtools::check()

# If you don't understand an error,

# google it!

CRAN

# Submission to CRAN

```
usethis::use_release_issue()
```

```
devtools::release()
```

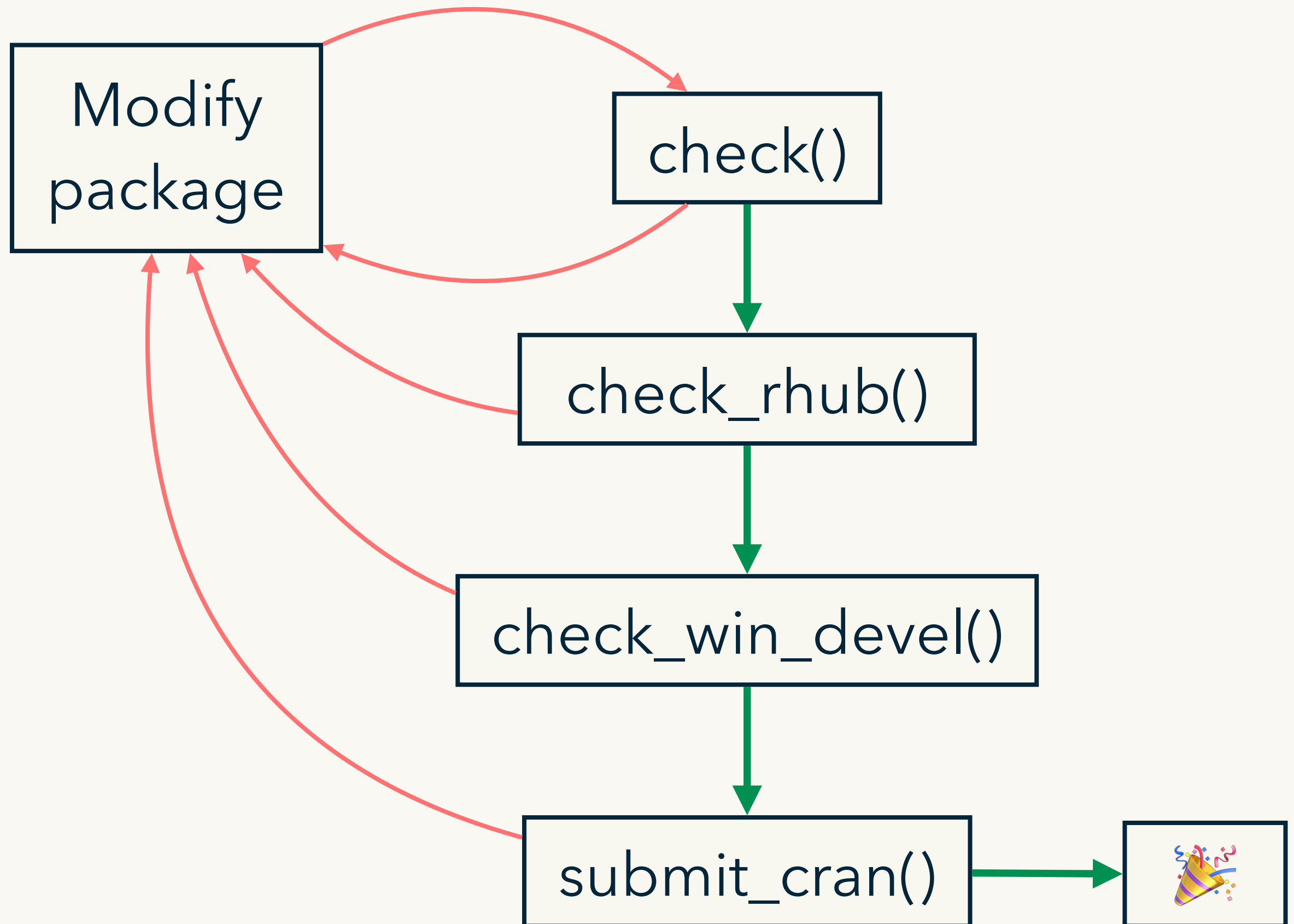
```
# Particularly important for larger packages
```

```
# where release process might take weeks.
```

```
# But also useful for smaller packages,
```

```
# and you should feel free to tweak for your
```

```
# needs
```



# cran-comments.md

Goal is to document  
your process

## Test environments

- \* local OS X install (R-release)
- \* win-builder (R-release, R-devel)

## R CMD check results

0 errors | 0 warnings | 1 note

- \* This is a new release.

There's always one note  
for a new submission

# If your submission fails

Do not despair! It happens to everyone, even R-core and tidyverse developers.

If it's from the CRAN robot, just fix the problem & resubmit.

If it's from a human, **do not respond** to the email and **do not argue**. Instead update cran-comments.md & resubmit.

# For resubmission:

This is a resubmission. Compared to the last submission, I have:

- \* First change.
- \* Second change.
- \* Third change

---

## Test environments

- \* local OS X install, R 3.2.2
- \* win-builder (devel and release)

## R CMD check results

...

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/)