

Document and share

September 2020

Charlotte Wickham & Sara Altman

Adapted from *Tidy Tools* by Hadley Wickham



Getting help

slido: Ask a question in the Q&A at anytime. Vote on questions.



In Breakout Rooms:

1. Ask your roommates
2. If your room is stuck, change your status to “Send Help” in the Google Doc

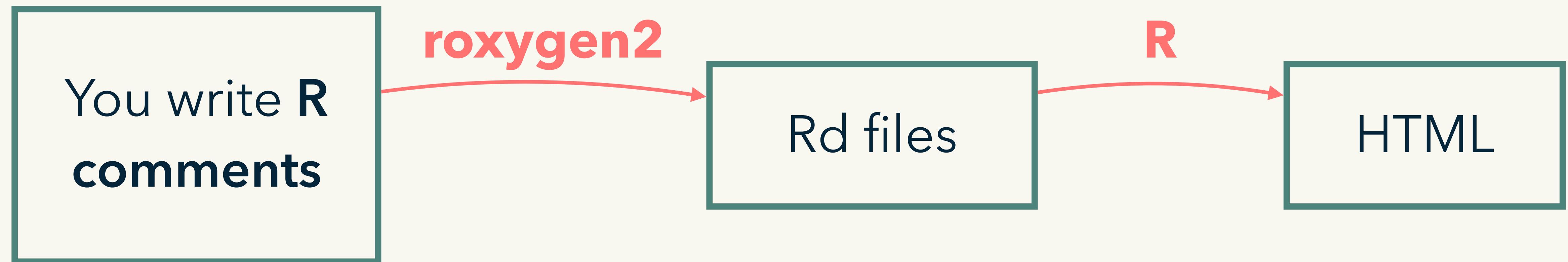
Zoom chat: Reserved for urgent technical matters (e.g., “we can’t hear you”)

Overview

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

Function docs with roxygen2

Roxygen2



You write specially formatted comments in .R

In R/add_col.R

```
'# Add a column to a data frame
#
#' @description Add a column to a data frame in a specified position.
#
#
#' @param x A data frame.
#' @param name Name of the new column.
#' @param value Vector of new value(s).
#' @param where Position in existing data frame to insert new values.
#
#' @return Data frame
#' @export
#
#' @examples
#' df <- data.frame(a = 1:3)
#' add_col(df, name = "x", value = c("a", "b", "c"))
#' add_col(df, name = "x", value = c("a", "b", "c"), where = 2)
add_col <- function(x, name, value, where = ncol(x) + 1) {
  if (name %in% names(x)) {
    x[[name]] <- value
    x
  } else {
    y <- setNames(data.frame(value), name)
    insert_into(x, y, where)
  }
}
```

You write specially formatted comments in .R

In R/add_col.R

Roxygen comment

```
'#' Add a column to a data frame
'#'
'#' @description Add a column to a data frame in a specified position.
'#'
'#'
'#' @param name Name of the new column.
'#' @param value Vector of new value(s).
'#' @param where Position in existing data frame to insert new values.
'#'
'#' @return Data frame
'#' @export
'#'
'#' @examples
'#' df <- data.frame(a = 1:3)
'#' add_col(df, name = "x", value = c("a", "b", "c"))
'#' add_col(df, name = "x", value = c("a", "b", "c"), where = 2)
add_col <- function(x, name, value, where = ncol(x) + 1) {
  if (name %in% names(x)) {
    x[[name]] <- value
    x
  } else {
    y <- setNames(data.frame(value), name)
    insert_into(x, y, where)
  }
}
```

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, where = ncol(x) + 1)
}

\arguments{
\item{x}{A data frame.}

\item{name}{Name of the new column.}

\item{value}{Vector of new value(s).}

\item{where}{Position in existing data frame to insert new values.}
}

\value{
Data frame
}

\description{
Add a column to a data frame in a specified position.
}

\examples{
df <- data.frame(a = 1:3)
add_col(df, name = "x", value = c("a", "b", "c"))
add_col(df, name = "x", value = c("a", "b", "c"), where = 2)
}
```

*In almost all cases
you can ignore
these files*

Add a column to a data frame

Description

Add a column to a data frame in a specified position.

R translates to
.html for viewing

Usage

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

Arguments

x A data frame.

name Name of the new column.

value Vector of new value(s).

where Position in existing data frame to insert new values.

Value

Data frame

Examples

```
df <- data.frame(a = 1:3)
add_col(df, name = "x", value = c("a", "b", "c"))
add_col(df, name = "x", value = c("a", "b", "c"), where = 2)
```

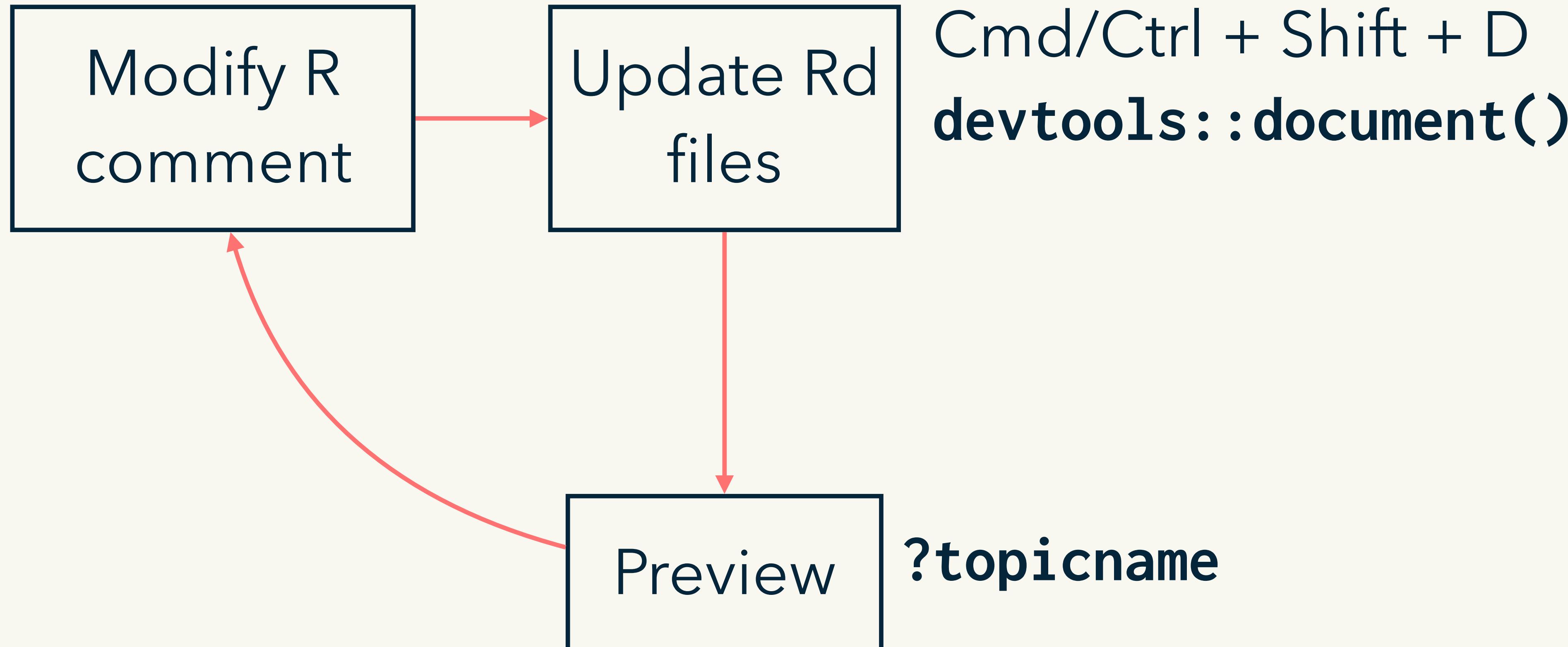
Join at
slido.com
#80875



Passcode: **tidytools**



Documentation workflow

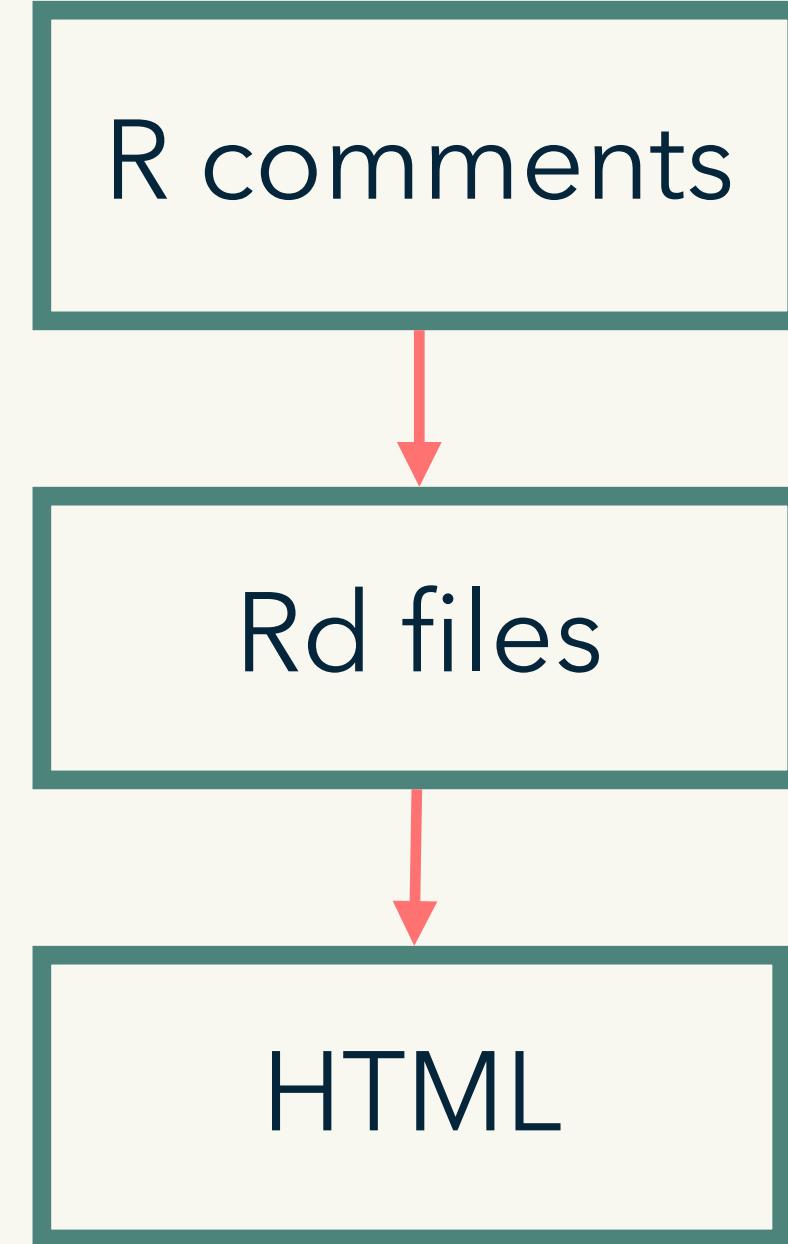


You must have loaded the package with `load_all()` at least once.

Check for message "Rendering development documentation..."

Can't preview links.

Easier



Harder

Writing text that
other humans can
understand.

Change project to:

[fordogs]

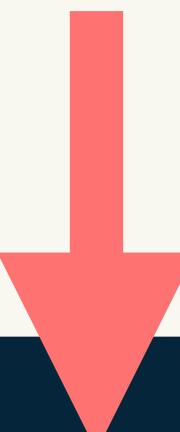


```
create_from_github("skaltman/fordogs", fork = FALSE, protocol = "https")
```

Your turn: Breakout rooms

Follow the steps on the **following slides** to:

1. Open the `fordogs` package.
2. Fix the typos in the `fbind()` documentation.
3. Practice the documentation workflow.
4. Let me know when you're done.



Your turn: 1. Open fordogs



Get the package with:

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

OR: <https://github.com/skaltman/fordogs>

Clone or Download -> Download .zip.

Unzip, open fordogs.Rproj

Your turn: 2. Fix the typos in the fbind() docs

Fix any typos you find, and fill in any missing information.

Your turn: 3. Practice the workflow

```
# Cmd/Ctrl + Shift + D  
devtools::document()
```

```
# Preview documentation  
?fbind
```

Your turn: 4. Ready to move on?

```
# Check in with your breakout room.  
# Is everyone ready to move on?  
# If not, help them out!  
# If yes, edit your status in the Google Doc:  
# bit.ly/built-tt-breakout  
  
# While you wait:  
# Go beyond fixing typos and continue editing the  
fbind() documentation for clarity and thoroughness
```

R: Sum of Vector Elements

R Documentation

First sentence is the **title**

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 is returned, otherwise the sum of the non-missing elements.

Everything else is the **details**

The description block

First sentence is the **title**

```
#' 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.
```

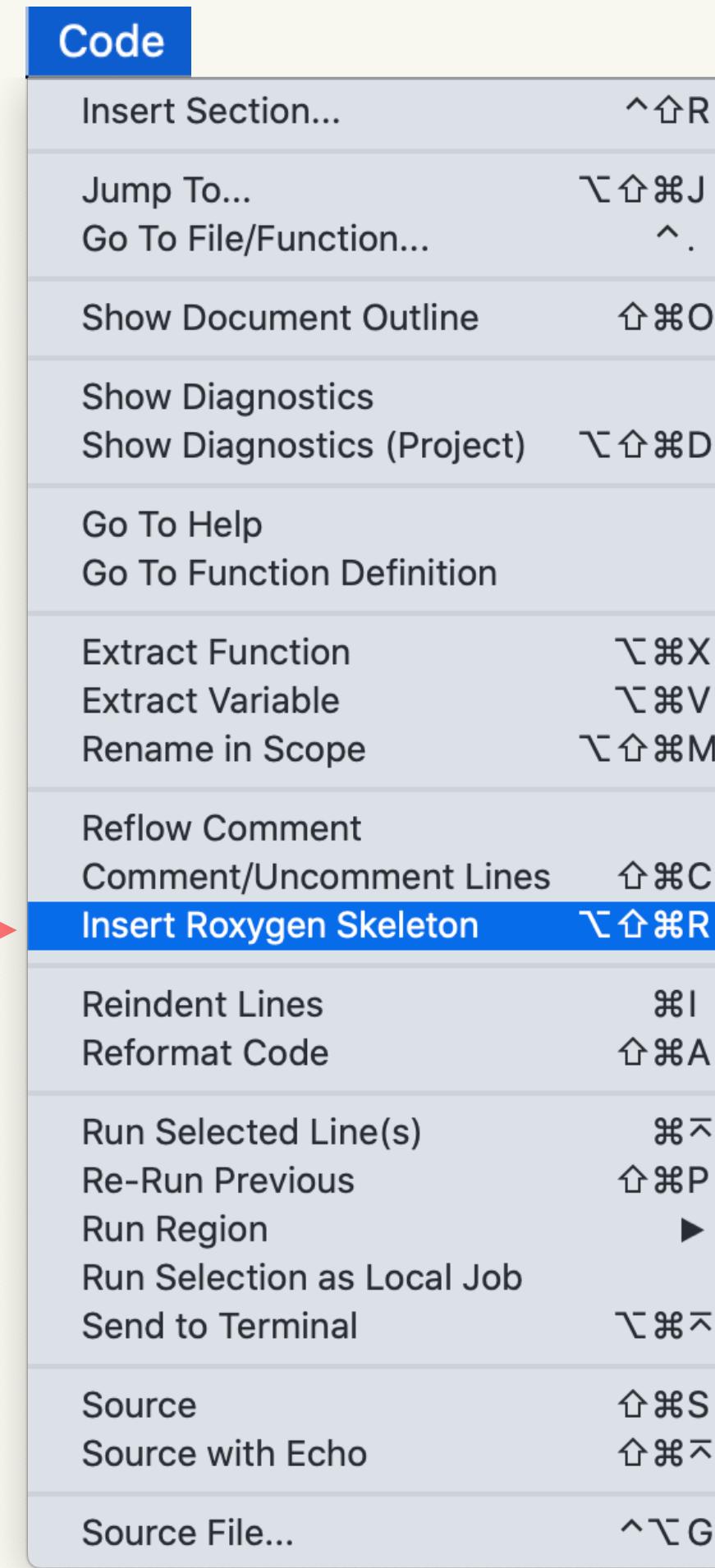
Next paragraph is
the **description**

Everything else is the **details**

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

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

Add Roxygen outline



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

Cursor must be inside the function.

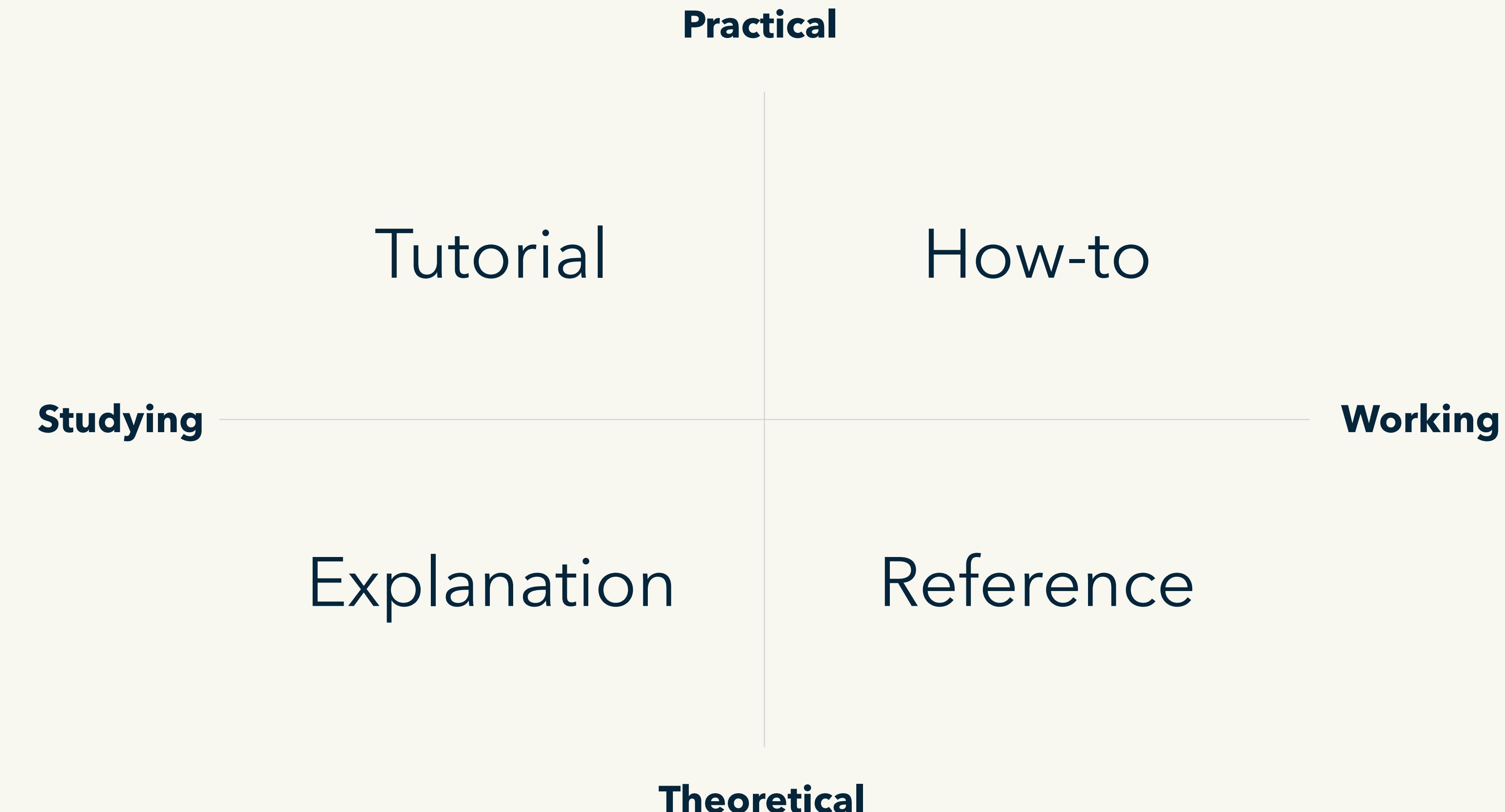
Can also make a [snippet!](#)

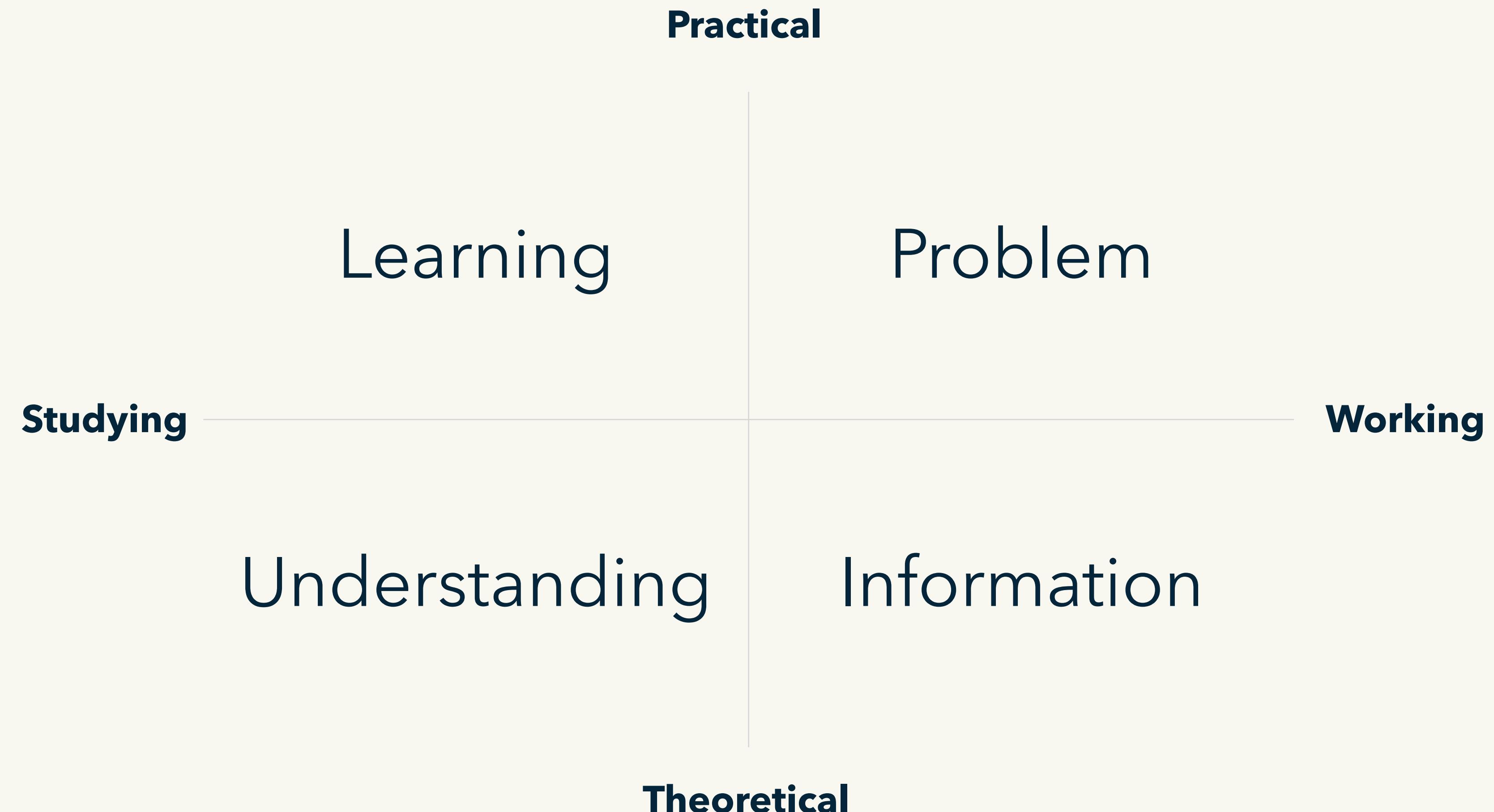
Join at
slido.com
#80875

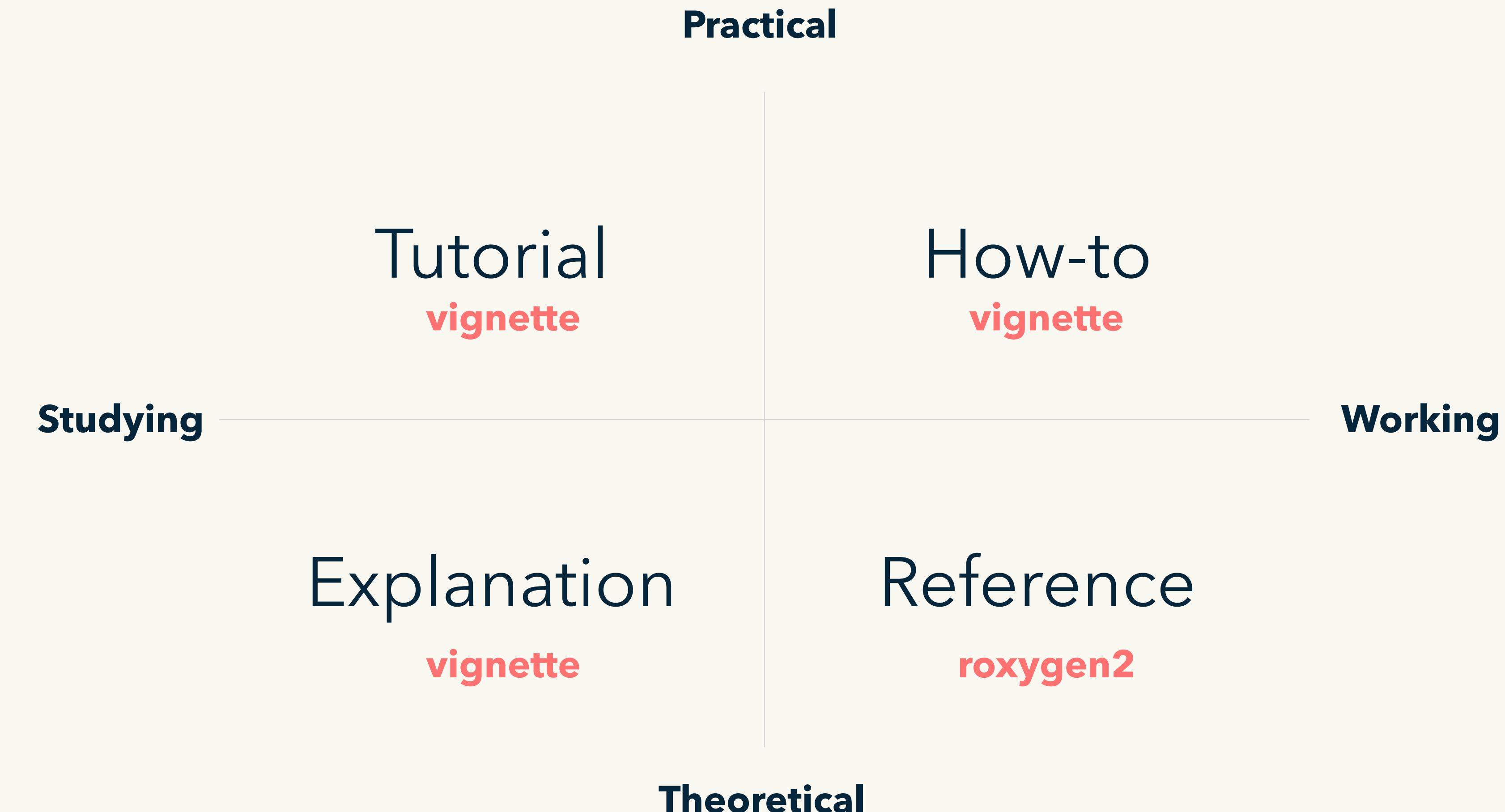


Passcode: **tidytools**









Join at
slido.com
#80875



Passcode: **tidytools**



Your turn: Breakout rooms

Follow the steps on the **following slides** to:

1. Complete the `fbind()` documentation.
2. Document `fdist()`.
3. Let me know when you're done.

Your turn: 1. Complete fbind() documentation

Complete the documentation for fbind().

A good minimum is to describe the types of the input and output.

Your turn: 2. Document fdist()

The function `fdist()` is written for you.

Document `fdist()`.

```
# Example calls to fdist():
fdist(factor(rpois(50, 5)))
fdist(factor(rpois(500, 5)))
```

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

Your turn: 4. Ready to move on?

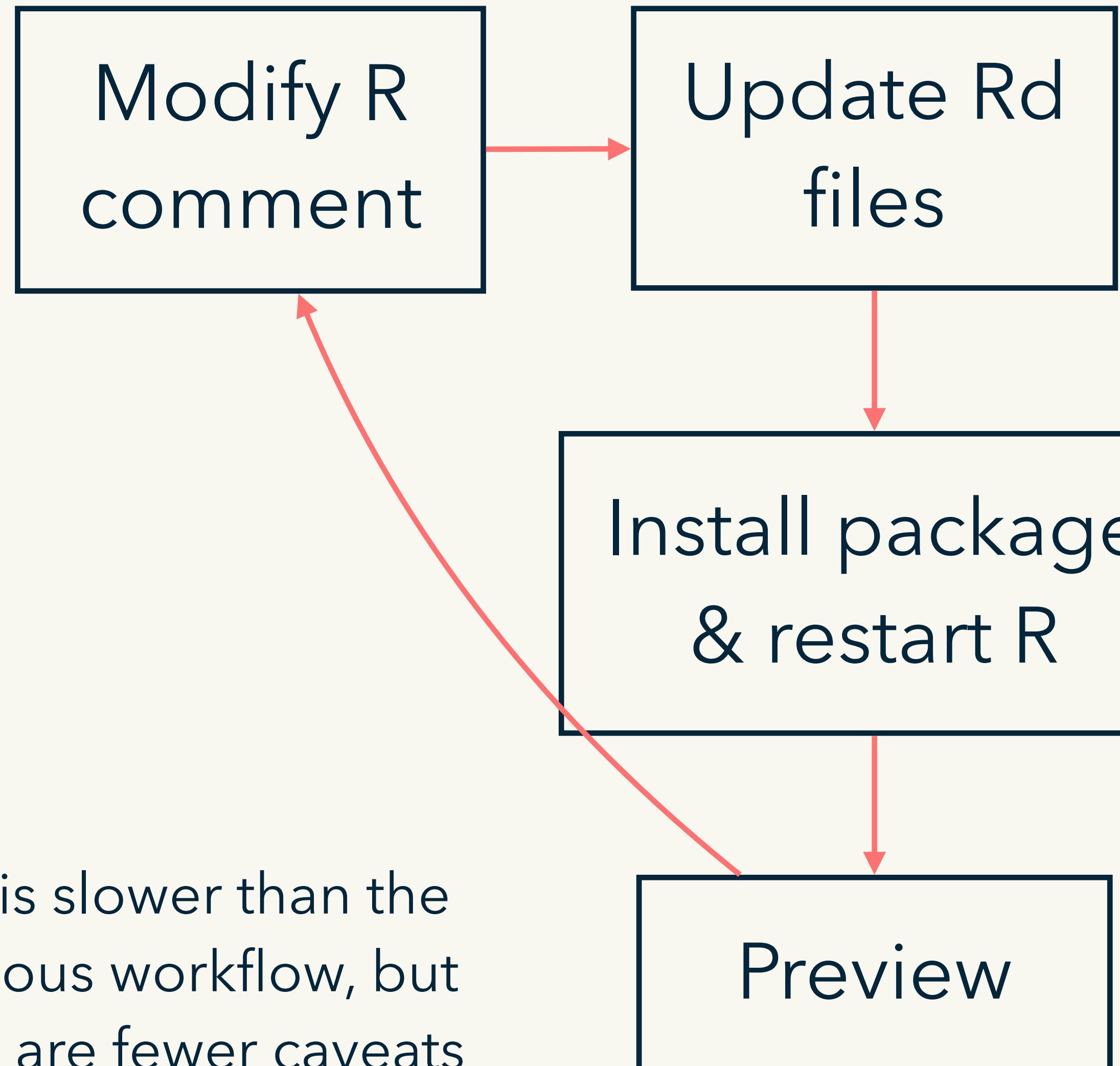
```
# Check in with your breakout room.  
# Is everyone ready to move on?  
# If not, help them out!  
# If yes, edit your status in the Google Doc:  
# bit.ly/built-tt-breakout  
  
# While you wait:  
# Does fordogs include any tests? Try them out and  
add your own.
```

Use markdown for formatting

```
# usethis::use_roxygen_md()

**bold**, _italic_, `code`  
  
* [func()]  
* [pkg::func()]  
* [link text][func()]  
* [link text][pkg::func()]
```

Documentation workflow 2



Cmd/Ctrl + Shift + D
devtools::document()

Cmd/Ctrl + Shift + B
 Install and Restart

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

?topicname

Package
documentation with
rmarkdown

Use vignettes for broader topics

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

Adds to DESCRIPTION

Suggests: knitr

VignetteBuilder: knitr

Creates vignettes/

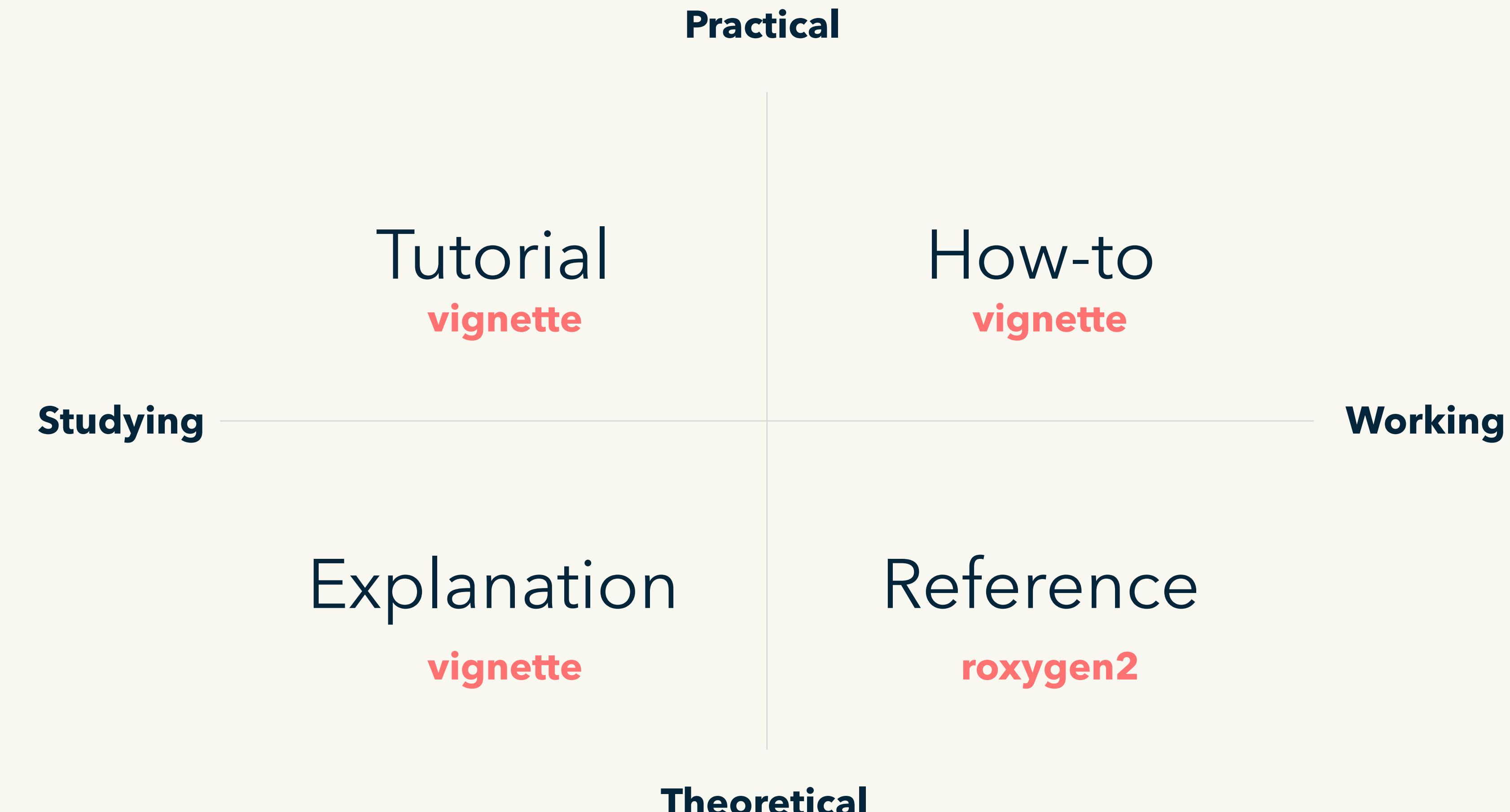
Drafts vignettes/name.Rmd

Use vignettes for broader topics

```
--  
title: "Vignette Title"  
output: rmarkdown::html_vignette  
vignette: >  
  \%VignetteIndexEntry{Vignette Title}  
  \%VignetteEngine{knitr::rmarkdown}  
  \%VignetteEncoding{UTF-8}  
--
```

Special output format

Special metadata
needed by R



Big picture in README

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

```
# * 2-3 paragraph overview  
# * Installation instructions  
# * Usage example, with pointer to vignettes
```

Your turn: Breakout rooms

Follow the steps on the **following slides** to:

1. Activate markdown formatting.
2. Add a vignette and README.
3. Let me know when you're done.

Your turn: 1. Activate markdown formatting

```
# Activate markdown formatting  
usethis::use_roxygen_md()
```

```
# Use markdown formatting in fbind()/fdist()  
documentation
```

Your turn: 2. Add a vignette and README

```
# Add a vignette  
usethis::use_vignette("name of your vignette")  
# Edit your vignette (doesn't need to be long)  
  
# Add an Rmd README  
usethis::use_readme_rmd()  
# Follow guidelines on slide 38 to add info  
  
# Preview both
```

Your turn: 3. Ready to move on?

- # Check in with your breakout room.
- # Is everyone ready to move on?
- # If not, help them out!
- # If yes, edit your status in the Google Doc:
- # bit.ly/built-tt-breakout

- # While you wait:
- # Add more information to your vignette/README

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()`

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.



To avoid frustration run
early and run often

“If it hurts,
do it more often”

— Martin Fowler

Types of problem

ERROR

Must fix!

WARNING

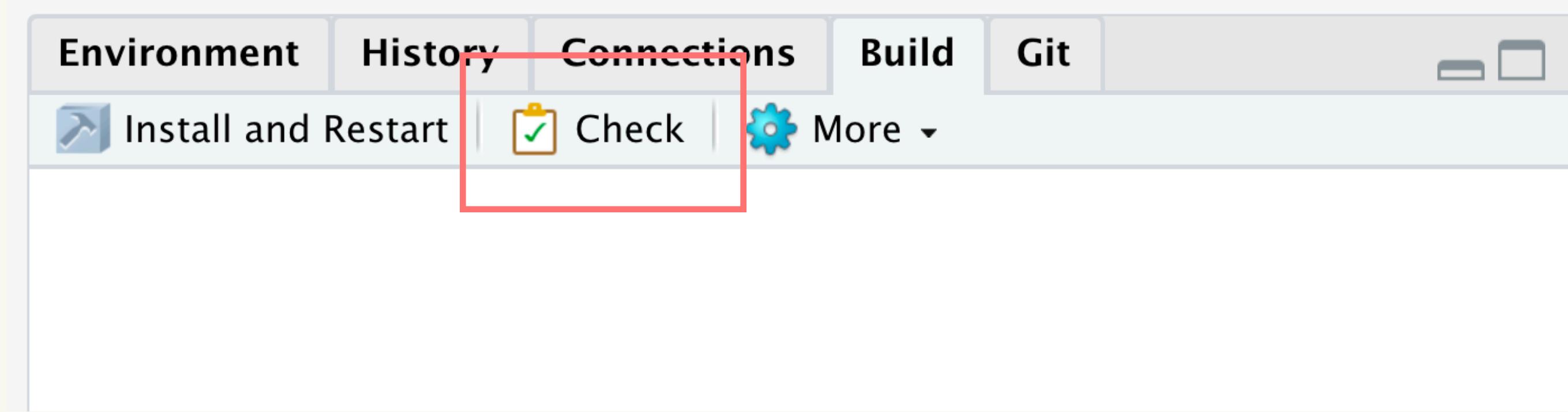
Fix if submitting to CRAN

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!
```

Git and GitHub

```
# Sets up local Git repo for your project  
usethis::use_git()  
  
# Creates associated GitHub repo  
usethis::use_github()
```

More info:

<https://usethis.r-lib.org/articles/articles/usethis-setup.html>

<https://dcl-workflow.stanford.edu/project-setup.html#setup-github>

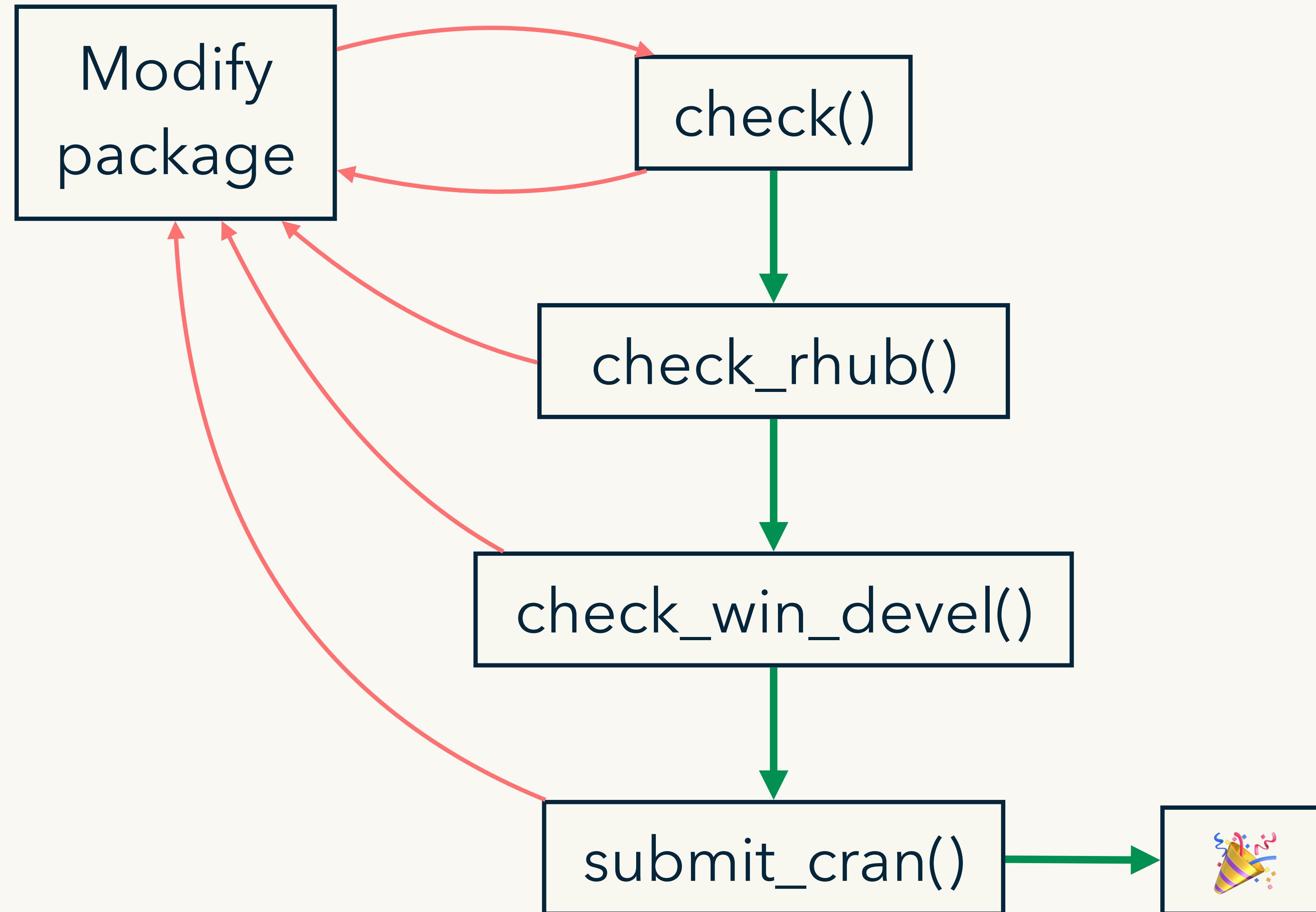
Continuous integration (CI)

```
# use_git()  
# use_github()  
  
use_coverage()  
use_github_actions()
```

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
```

...

Summary

Setup

Create R file
`use_r()`

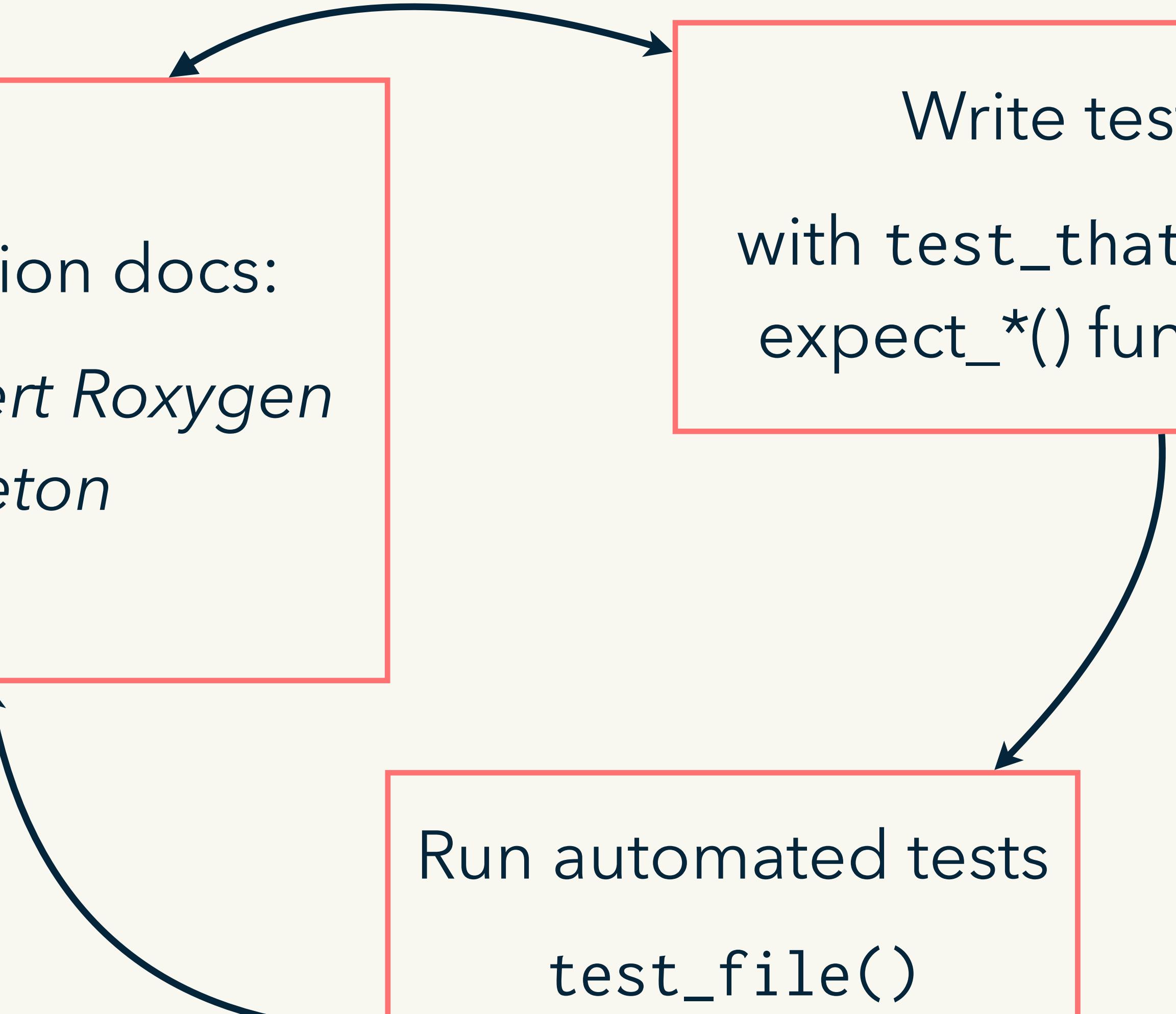
Create test file
`use_test()`

Add function docs:
Code > Insert Roxygen skeleton

Write tests

with `test_that()` and
`expect_*`() functions

Run automated tests
`test_file()`
Cmd/Ctrl + T



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/)