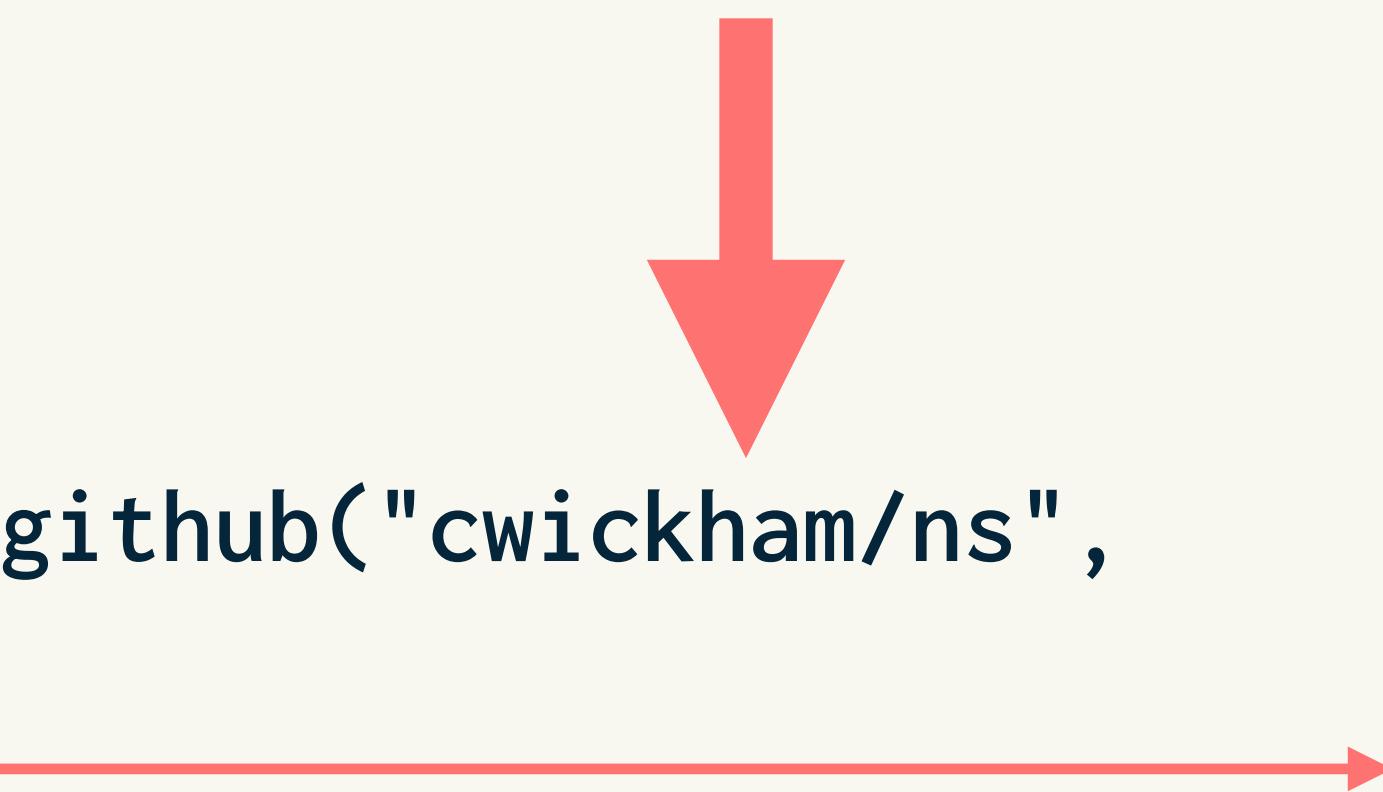


Welcome!

# Building Tidy Tools

Charlotte Wickham and Sara Altman

```
# We'll get started at 8:35am  
  
# Materials on GitHub: http://bit.ly/build-tt  
# 0. Skim the README  
# 1. Check you've completed  
#     the Setup section  
# 2. Get these slides from the  
#     Schedule section  
# 3. Run usethis::create_from_github("cwickham/ns",  
fork = FALSE)  
# Questions? Ask in Slido
```



**Materials:** <http://bit.ly/build-tt>

# Document and share

**September 2020**

Charlotte Wickham & Sara Altman

Adapted from *Tidy Tools* by Hadley Wickham



# Overview

1. Review
2. Function docs
3. Other docs
4. R CMD check
5. Sharing your package

# Review: workflow

## Setup

Create package  
`create_package()`

Create R file  
`use_r()`

Create test file  
`use_test()`

Modify code

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

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

# Function docs with roxygen2

# Function documentation

```
# View documentation for one function
```

```
?function
```

```
# All package docs, including all function docs
```

```
help(package = "mypackage")
```

## Create or edit R or test files

### Description

This pair of functions makes it easy to create paired R and test files, using the convention that the tests for `R/foofy.R` should live in `tests/testthat/test-foofy.R`. You can use them to create new files from scratch by supplying `name`, or if you use RStudio, you can call to create (or navigate to) the paired file based on the currently open script.

### Usage

```
use_r(name = NULL, open = rlang::is_interactive())  
  
use_test(name = NULL, open = rlang::is_interactive())
```

### Arguments

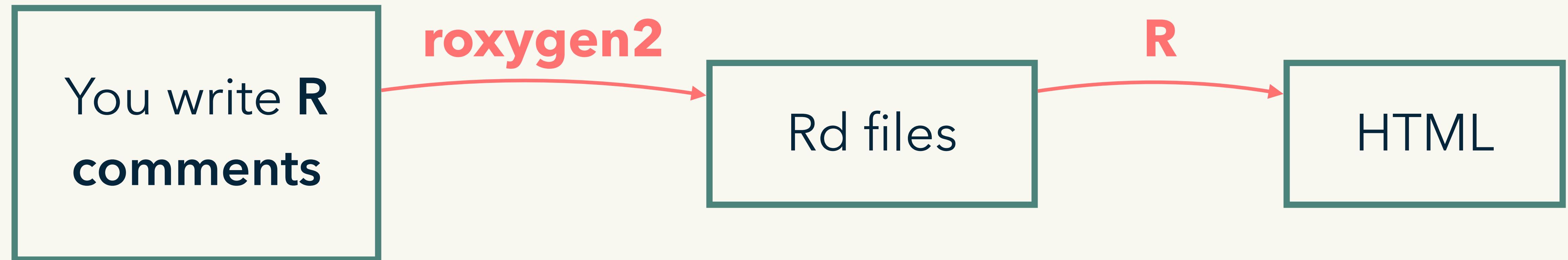
`name` Either a name without extension, or `NULL` to create the paired file based on currently open file in the script editor. If the R file is open, `use_test()` will create/open the corresponding test file; if the test file is open, `use_r()` will create/open the corresponding R file.

`open` Whether to open the file for interactive editing.

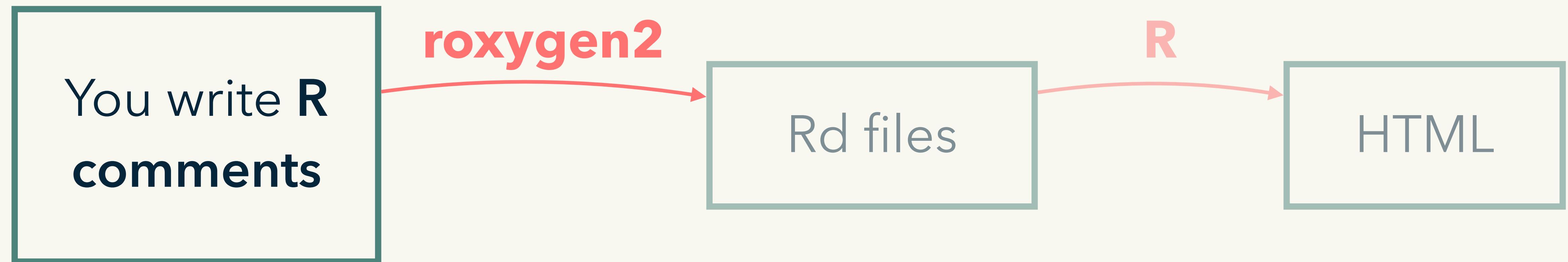
### See Also

The [testing](#) and [R code](#) chapters of [R Packages](#).

# 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)
  }
}
```

The comments go **above** the function definition

# 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.

## 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)
```

R translates to  
.html for viewing

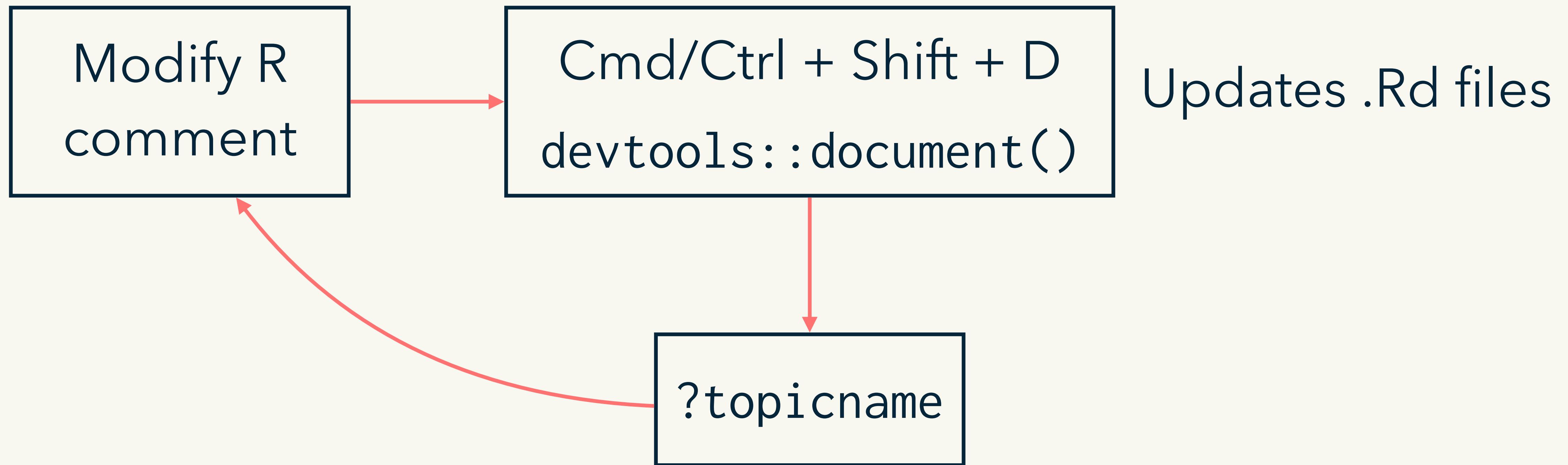
Join at  
**slido.com**  
**#80875**



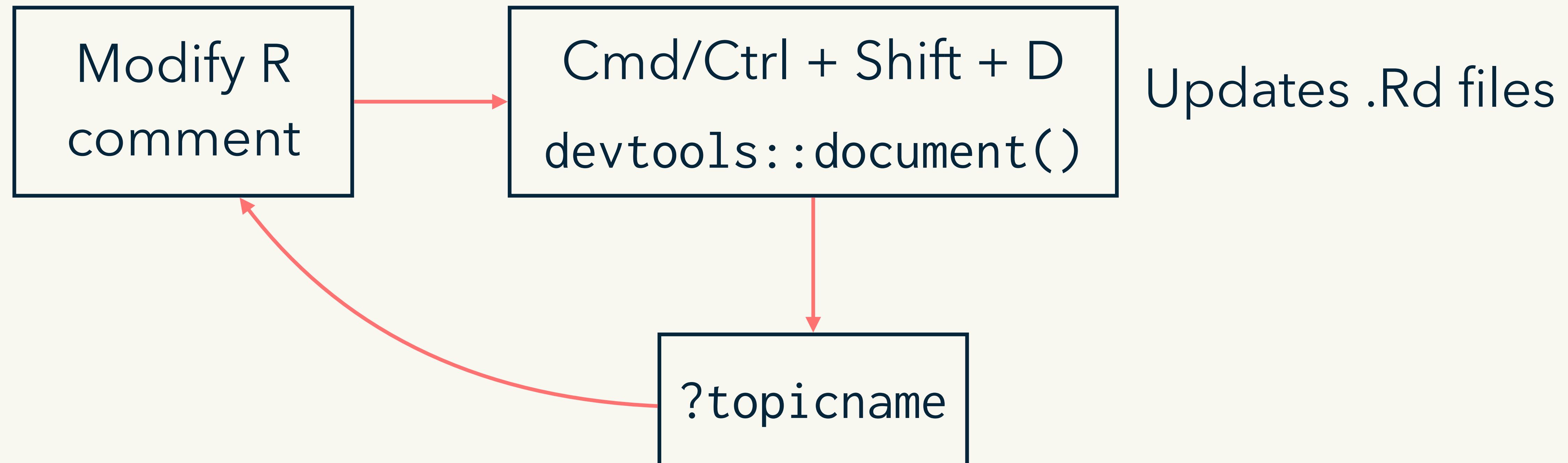
Passcode: **tidytools**



# Documentation workflow



# Documentation workflow



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

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/download `fordogs`.
2. Fix typos `fbind()`'s documentation.
3. Practice the workflow.
4. Let me know when you're done.

Goals:

1. Understand how to edit function documentation.
2. Understand the documentation workflow.

10 minutes



# Your turn: 1. Open fordogs



**If you already have fordogs:**

Open it up in RStudio.

**If you don't have fordogs:**

Get the package:

```
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/build-tt-breakout  
  
# While you wait:  
# Go beyond fixing typos and continue editing the  
# fbind() documentation for clarity and thoroughness.
```

# Questions?

 Editing function docs

 Updating function docs

Writing function docs

# 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!](#)

# The description block

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

# 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**

R: Sum of Vector Elements

su 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 is returned.

Next paragraph is the **description**

Everything else is the **details**

# Tags

@param name Description.

@param name Description.

@param name Description.

@param x Data frame.

@param value Vector of new values.

- \* Type
- \* Describe what the argument does.
- \* Phrase ending in “.”

# Tags

@return Description.

- \* Describe output from the function.
- \* Not always necessary.

@return A data frame.

# Tags

@examples

example(1)

example(2)

@examples

```
df <- data.frame(a = 1)
```

```
add_col(df, "b", 2)
```

- \* Simple examples of how to use your function.
- \* Should run without errors.

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

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

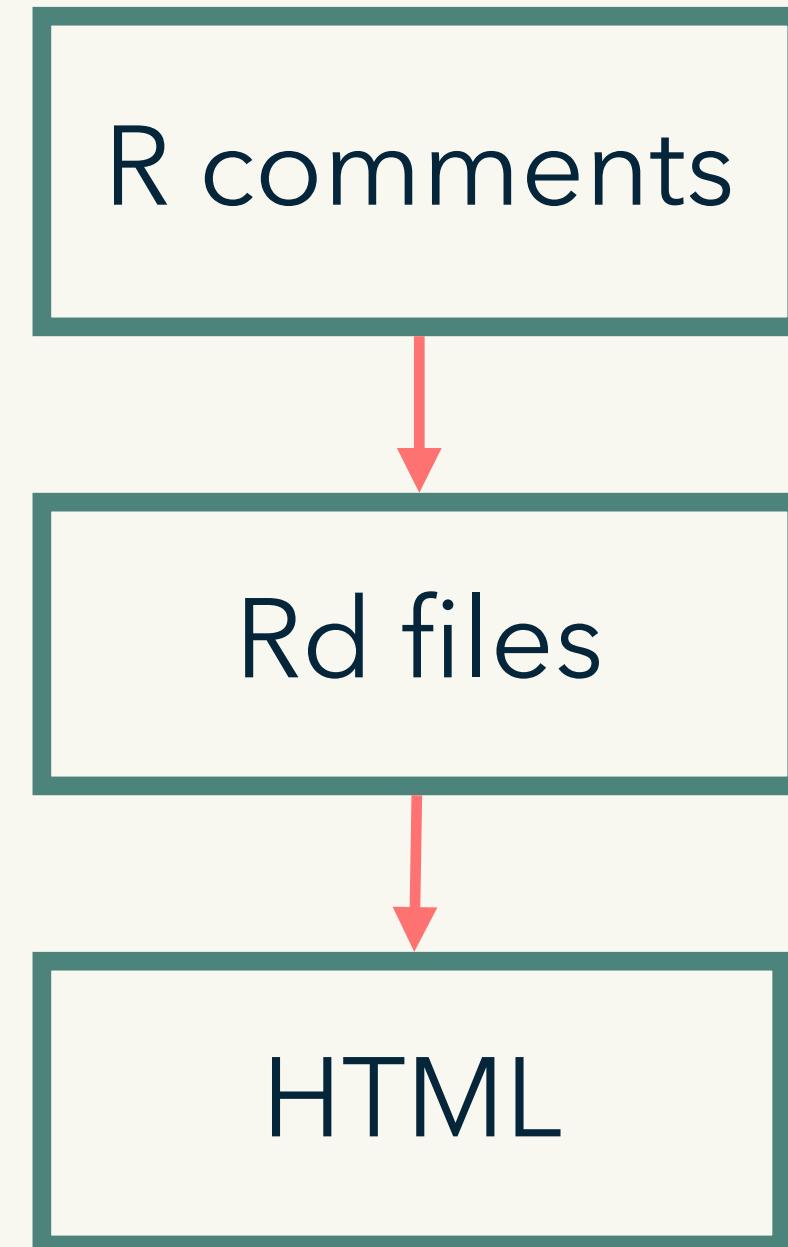
Join at  
**slido.com**  
**#80875**



Passcode: tidytools



# Easier



# Harder

Writing text that  
other humans can  
understand.

# 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.

12 minutes



Slides: <http://bit.ly/build-tt>

# 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?

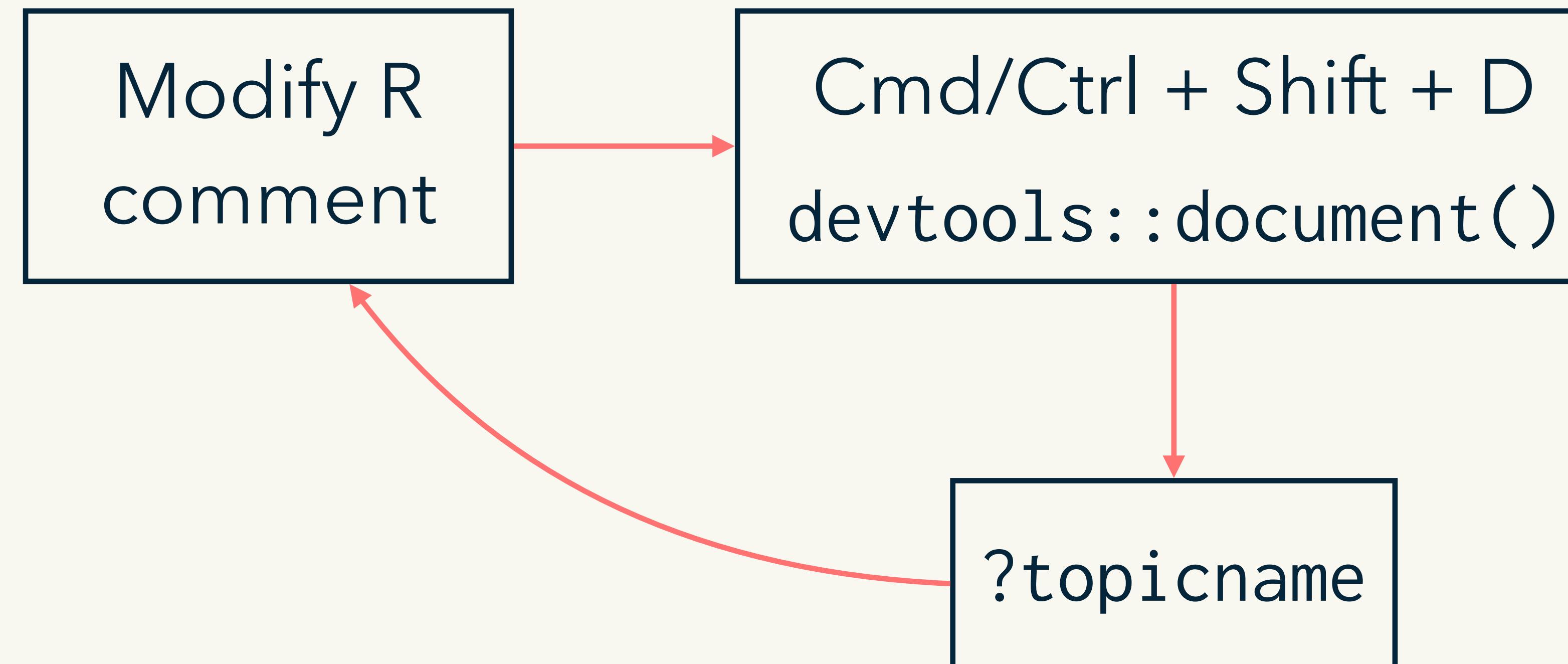
```
# Check in with your breakout room.  
# Is everyone ready to move on?  
# If not, help them out!  
# If yes, edit your status in the Google Doc:  
# bit.ly/built-td-breakout  
  
# While you wait:  
# Does fordogs include any tests? Try them out and  
add your own.
```

# Questions?

# Use markdown for formatting

```
# usethis::use_roxygen_md()

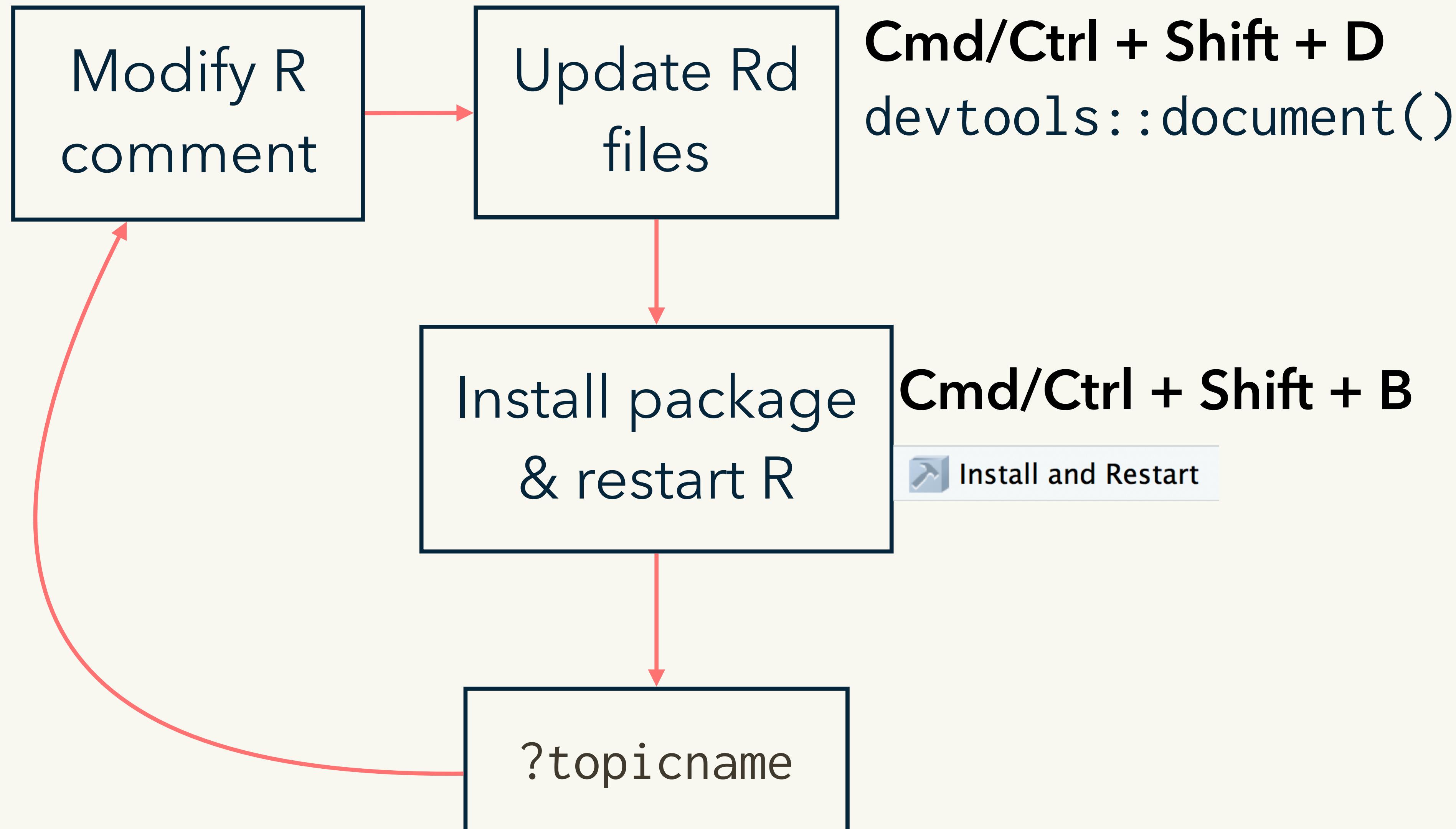
**bold**, _italic_, `code`  
  
* [func()] # Link to function in same package  
* [pkg::func()] # Link to function in another package  
* [link text][func()] # Different display text  
* [link text][pkg::func()]
```



Updates .Rd files



# Documentation workflow 2



Join at  
**slido.com**  
**#80875**



Passcode: **tidytools**



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{test}
```

```
  %\VignetteEngine{knitr::rmarkdown}
```

```
  %\VignetteEncoding{UTF-8}
```

```
---
```

```
```{r, include = FALSE}
```

```
knitr::opts_chunk$set(
```

```
  collapse = TRUE,
```

```
  comment = "#>"
```

```
)
```

```
```
```

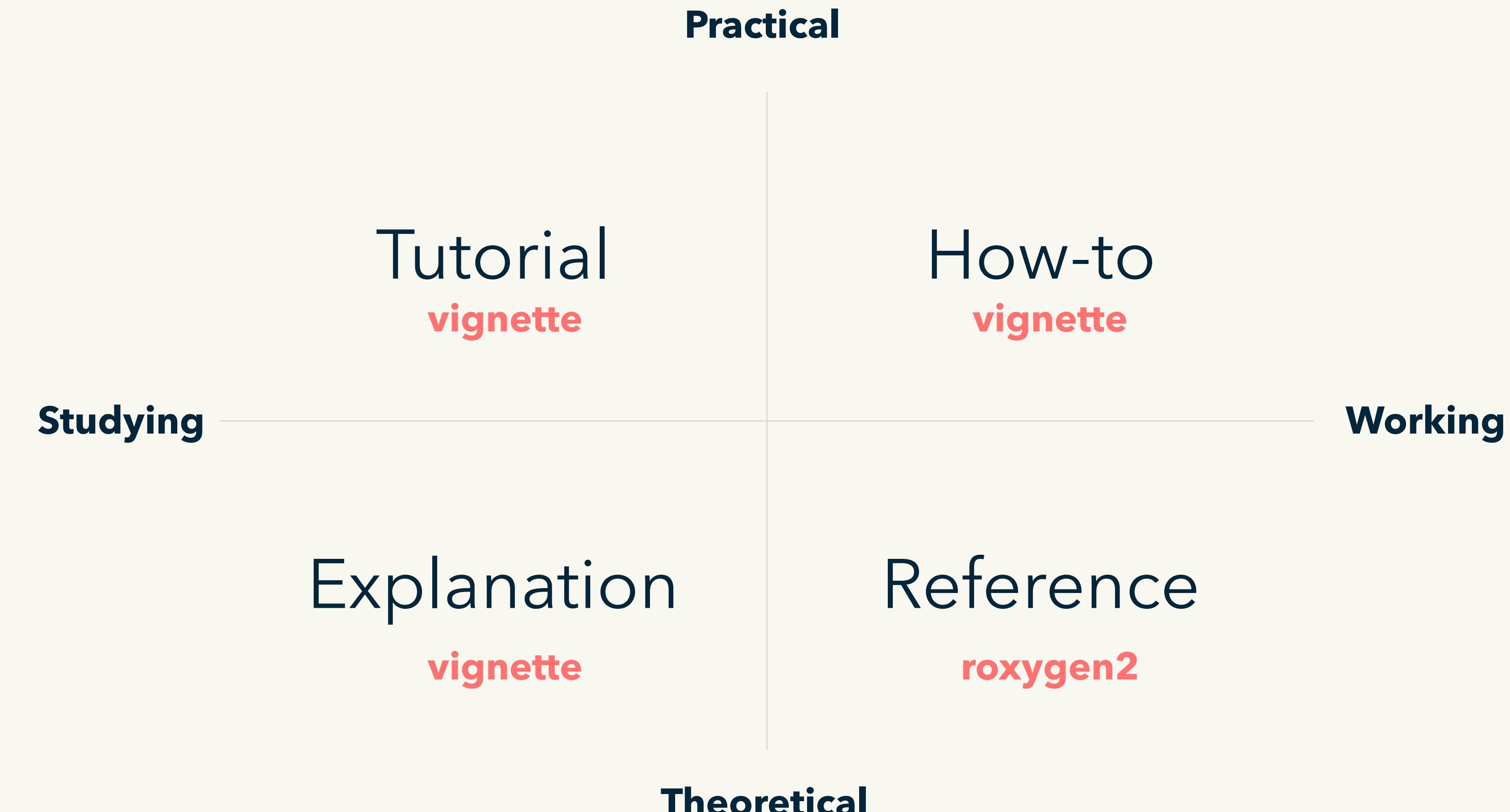
```
```{r setup}
```

```
library(fordogs)
```

```
```
```

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
```

# 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()`

# R CMD check

# R CMD check

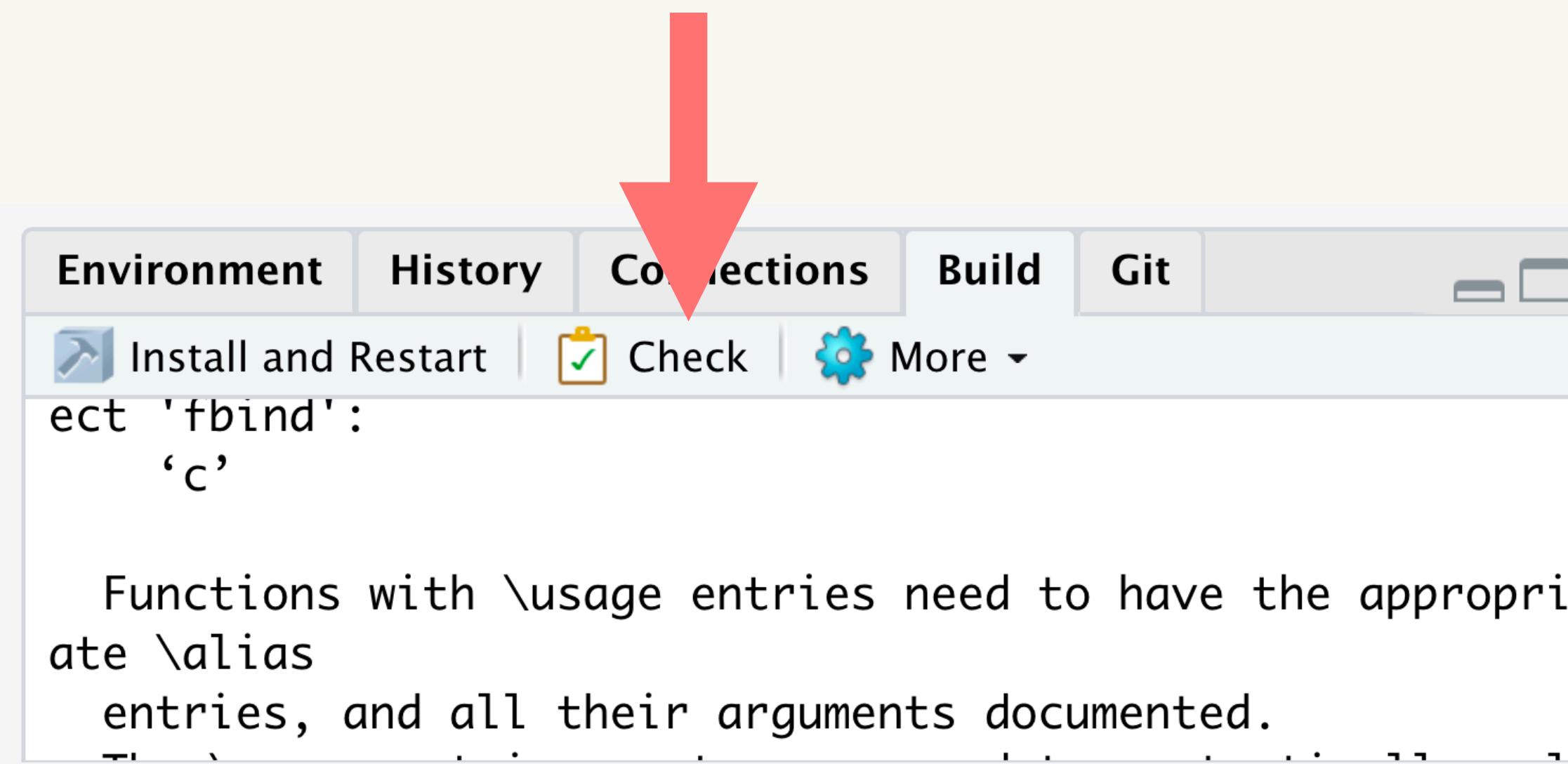
Runs automated checks for common problems in R packages.

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

# R CMD check: 3 methods

```
# If you don't understand an error,  
# google it!  
  
devtools::check()
```

**Shift + Cmd/Ctrl + E**



# R CMD check

Run early.



Run often.



“If it hurts,  
do it more often”

— Martin Fowler

# Types of problem

|         | Local | CRAN |
|---------|-------|------|
| ERROR   | FIX   | FIX  |
| WARNING |       | FIX  |
| NOTE    |       | FIX  |

# Your turn: Breakout rooms

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

1. Run R CMD check. Make sure it passes.
2. Add a vignette.
3. Activate markdown formatting and try it out.
4. Let me know when you're done.

10 min



# Your turn: 1. Run R CMD check

```
devtools::check()
```

```
# Or: Click “Check”
```

```
# Or: Cmd/Ctrl + Shift + E
```

```
# If it doesn't pass (besides the timestamp  
note), fix the problems.
```

```
# If you don't understand an error, try  
Googling it!
```

## Your turn: 2. Add a vignette

```
# Add a vignette  
usethis::use_vignette("name of your vignette")  
# Add one-two sentences.  
  
# Can preview by install and restart-ing, then  
# knitting
```

# Your turn: 3. Activate markdown formatting

```
usethis::use_roxygen_md()
```

```
# Try it out! See Slide 35 for details.
```

```
# Remember: To render links, you have to  
install + restart. Then ?function.
```

# 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/build-tt-breakout](http://bit.ly/build-tt-breakout)
  
- # While you wait:
- # Add more information to your vignette.

# Questions?

# Sharing: Git and GitHub

# Create a repo for your package

```
# 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>

# Install from GitHub

```
# Once your package is on GitHub, others can  
# install it with:  
remotes::install_github("username/package_name")  
  
# Or:  
devtools::install_github("username/package_name")
```

# Sharing: CRAN

# Submitting to CRAN

More stringent requirements than just hosting on GitHub.

Must cleanly pass R CMD check.

See <https://r-pkgs.org/release.html> for (much) more info.

# Summary

## Setup

Add function docs:

*Code > Insert  
Roxygen skeleton*

Run early, run often

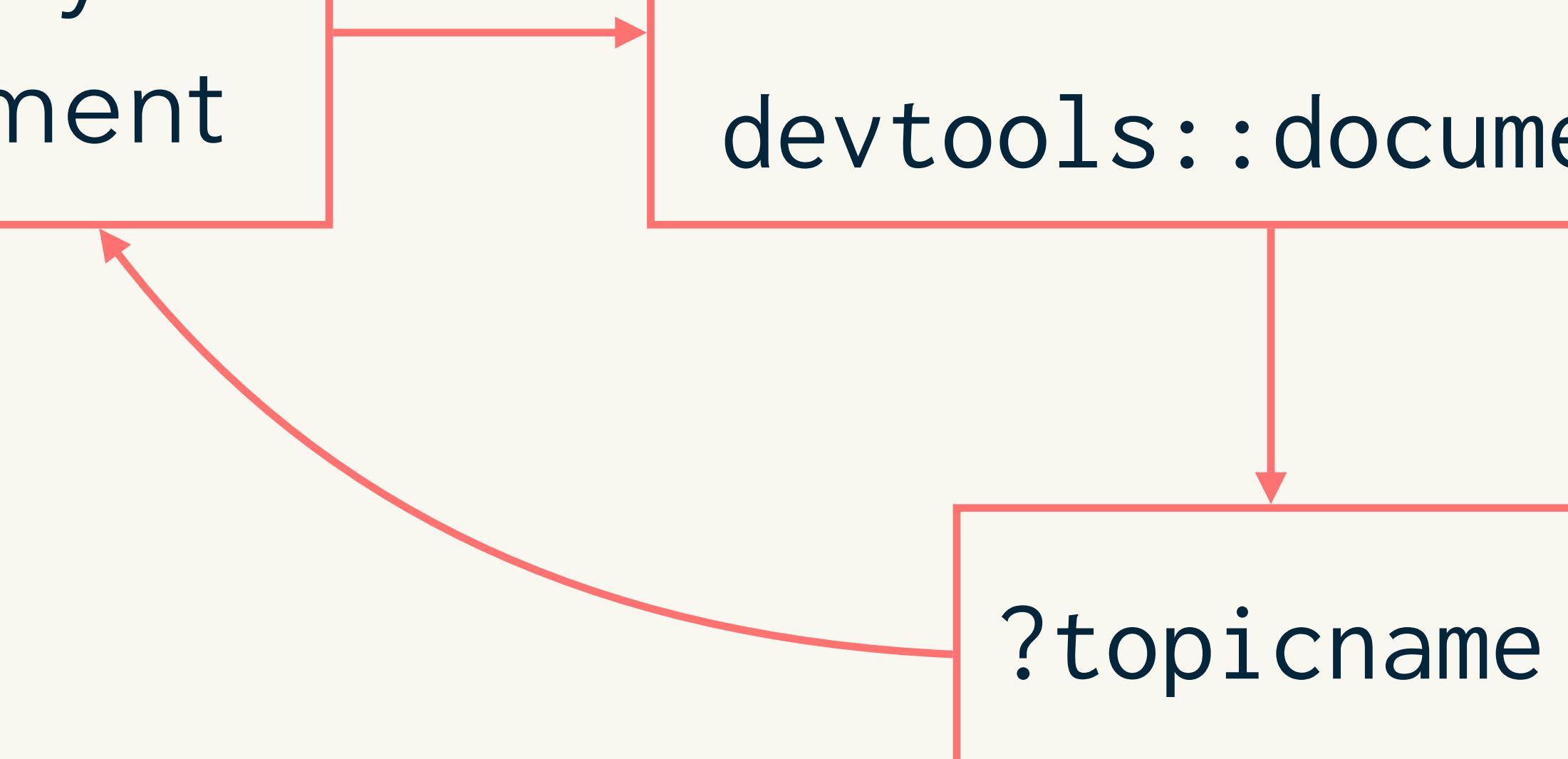
R CMD check  
`devtools::check()`

Modify R comment

Updates .Rd files

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

?topicname



# Building Tidy Tools

Charlotte Wickham and Sara Altman

# We'll get started at 8:35am

# Materials on GitHub: <http://bit.ly/build-tt>

# 0. Skim the README

# 1. Check you've completed

# the Setup section

# 2. Get these slides from the

# Schedule section

# Questions? Ask in Slido



*ON BREAK UNTIL 10:30am PDT*

Join at

**slido.com**

**#80875**

Passcode: **tidytools**

**Materials:** <http://bit.ly/build-tt>

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/)