

# R Packages: Document R Code

## ?mean

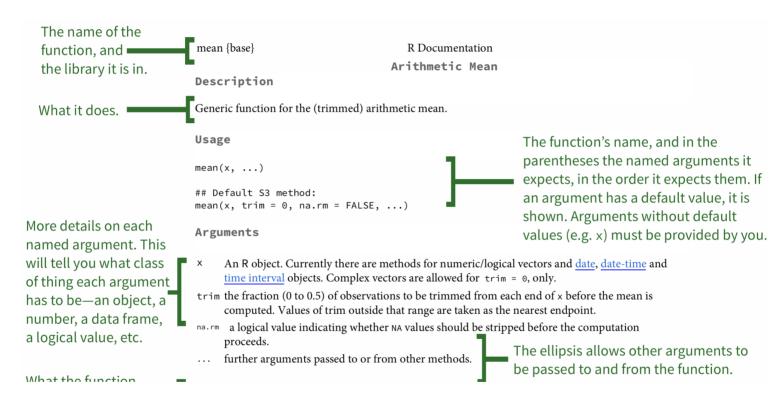


Image by Kieran Healy



roxygen2: In-Line Documentation for R



roxygen2: In-Line Documentation for R

Write documentation with your functions





roxygen2: In-Line Documentation for R

Write documentation with your functions

render with document()





# Insert roxygen skeleton

Code > Insert Roxygen
Skeleton



# Insert roxygen skeleton

Code > Insert Roxygen
Skeleton

Ctrl/Cmd + Shift + Alt/Opt + R



```
add_one <- function(x) {
    x <- x + 1
    x
}</pre>
```



#### Insert Roxygen Skeleton

```
#' Title
# 1
   @param x
# 1
#' @return
#' @export
# 1
#' @examples
add_one <- function(x) {</pre>
  x < -x + 1
```



```
Roxygen comments
#' Title
# 1
# 1
#' @return
#' @export
# 1
#' @examples
add_one <- function(x) {</pre>
  x < -x + 1
```



```
#' Title
              Roxygen tags
   @param x
#' @return
#' @export
# 1
#' @examples
add_one <- function(x) {</pre>
  x < - x + 1
```



```
#' Title
# 1
                Description and
   @param x
                     Details
# 1
#' @return
#' @export
# 1
add_one <- function(x) {</pre>
  x < -x + 1
```



```
Describe
               argument
#' Title
               parameter
#' @param x
#' @return
#' @export
# 1
#' @examples
add_one <- function(x) {</pre>
  x < -x + 1
```



```
#' Title
                  Describe
                  -what is
#' @return
                  returned
#' @export
# 1
add_one <- function(x)</pre>
  x < -x + 1
```

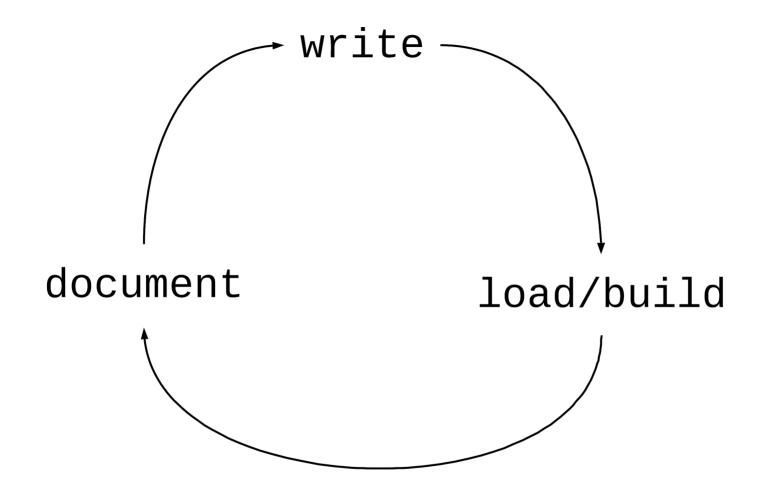


```
#' Title
                    Export the
                    -function
#' @return
                    pkg::add_one()
#' @export
# 1
add_one <- function(x) {</pre>
    \uparrow<- \times + 1
```



```
#' Title
#' @param x
# 1
#' @return
#' @export
# 1
                   -write examples
#' @examples-
add_one <- function(x) {</pre>
  x < -x + 1
```







# Write roxygen, run



## Write roxygen, run

```
theme_mako <- function(base_size = 14) {
   ggplot2::theme_dark(base_size = base_size) +
      ggplot2::theme(
      panel.background = ggplot2::element_rect(fill = "#0D98BA")
   )
}</pre>
```



# Write roxygen, run

```
#' A dark theme with a mako-like background
# "
  Oparam base size base font size
# "
  @return a ggplot2 theme
  Cexport
# "
  @examples
# "
  ggplot2::quickplot(iris$Sepal.Length) + theme mako()
# "
theme mako <- function(base size = 14) {
  ggplot2::theme dark(base size = base size) +
    ggplot2::theme(
     panel.background = ggplot2::element rect(fill = "#0D98BA")
```



```
— .Rbuildignore
  .gitignore
  DESCRIPTION
  NAMESPACE
   — themes.R
  man
  theme_mako.Rd
  shinRa.Rproj
```



### man/theme mako.Rd

```
## % Generated by roxygen2: do not edit by hand
## % Please edit documentation in R/themes.R
## \name{theme mako}
## \alias{theme mako}
## \title{A dark theme with a mako-like background}
## \usage{
## theme mako(base size = 14)
## }
## \arguments{
## \item{base size}{base font size}
## }
## \value{
## a ggplot2 theme
## }
## \description{
## A dark theme with a mako-like background
## }
## \examples{
##
## ggplot2::quickplot(iris$Sepal.Length) + theme mako()
##
## }
```



### ?theme\_mako

```
theme_mako {shinRa}
```

#### A dark theme with a mako-like background

#### **Description**

A dark theme with a mako-like background

#### Usage

 $theme_mako(base_size = 14)$ 

#### **Arguments**

base\_size base font size

#### Value

a ggplot2 theme

#### **Examples**

ggplot2::quickplot(iris\$Sepal.Length) + theme\_mako()



# Syntax

LaTeX like. See more at <a href="https://r-pkgs.org/man.html">https://r-pkgs.org/man.html</a>

use\_roxygen\_md() lets you write
in Markdown. See more at
https://roxygen2.rlib.org/articles/rd-formatting.html



Open the NAMESPACE file. What do you see?

Let's add documentation. Run use roxygen md()

Open r/themes.R. Insert a roxygen skeleton for theme avalanche().

**Change the title to "AVALANCHE ggplot2 themes"** 

Hit Enter/Return twice after the title. Make sure the new lines start with # '. Add this text: "Minimalistic ggplot themes for use on AVALANCHE reports."

Run document() or press Ctrl/Cmd + Shift + D. Read the help page for your function with ?theme\_avalanche.

Finally, look at the NAMESPACE file again. What changed?



exportPattern("^[^\\.]")



```
#' AVALANCHE ggplot2 themes
  Minimalistic ggplot themes for use on AVALANCHE reports.
# "
  Oparam base size
  @param ...
  @return
  @export
# "
#' @examples
theme avalanche <- function(base size = 14, ...) {
 ggplot2::theme minimal(base size = base size, ...) +
    ggplot2::theme(panel.grid.minor = ggplot2::element blank())
```



```
# Generated by roxygen2: do not edit by hand

export("%>%")
export(db_con)
export(get_resident_data)
export(theme_avalanche)
export(theme_avalanche_h)
export(theme_avalanche_v)
import(data.table)
importFrom(magrittr,"%>%")
```



# **Argument descriptions**

```
#' [other roxygen code]
#' @param x The name of a database to retrieve
get_data <- function(x) {
    # code to get data
}</pre>
```



# Argument descriptions: @inheritParams

```
#' [other roxygen code]
#' @param x The name of a database to retrieve
get_data <- function(x) {
    # code to get data
}

#' [other roxygen code]
filter_table <- function(x) {
    tbl <- get_data(x)
    # code to filter data
}</pre>
```



# Argument descriptions: @inheritParams

```
#' [other roxygen code]
#' @param x The name of a database to retrieve
get_data <- function(x) {
    # code to get data
}

#' [other roxygen code]
#' @inheritParams get_data
filter_table <- function(x) {
    tbl <- get_data(x)
    # code to filter data
}</pre>
```



# Examples

# Examples can be any kind of R code

```
#' [other roxygen code]

#' @examples

#'

#' library(dplyr)

#' get_data("daily_actice_users") %>%

#' filter(date == lubridate::today())

get_data <- function(x) {
    # code to get data
}</pre>
```



# Examples

# Examples can be any kind of R code

```
#' [other roxygen code]
#' @examples
#'

#' library(dplyr)

#' get_data("daily_actice_users") %>%

#' filter(date == lubridate::today())

get_data <- function(x) {
    # code to get data
}</pre>
```



# Examples

If you don't want to run examples, wrap them in \dontrun{} or

\donttest{}

```
#' [other roxygen code]
#' @examples
#'

#' \dontrun{
#' get_data("daily_active_users")
#' }

get_data <- function(x) {
    # code to get data
}</pre>
```

### Let's keep working on the documentation for

```
theme_avalanche():
```

Remove @param base\_size and replace it with: @inheritParams ggplot2::theme\_minimal

For @param ..., add: Additional arguments passed to [ggplot2::theme\_minimal()]

For @return, add: a ggplot theme.

For @examples, add two line breaks (make sure the new lines have roxygen comments!). Add this code: ggplot2::qplot(iris\$Sepal.Length) + theme\_avalanche()

Rebuild the documentation and check the help page.



```
#' AVALANCHE ggplot2 themes
# "
  Minimalistic ggplot themes for use on AVALANCHE reports.
# "
   @inheritParams ggplot2::theme minimal
   Oparam ... Additional arguments passed to [ggplot2::theme minimal
  @return a ggplot theme.
  @export
# "
  @examples
  ggplot2::gplot(iris$Sepal.Length) + theme avalanche()
# "
theme avalanche <- function (base size = 14, ...) {
  ggplot2::theme minimal(base size = base size, ...) +
    ggplot2::theme(panel.grid.minor = ggplot2::element blank())
```



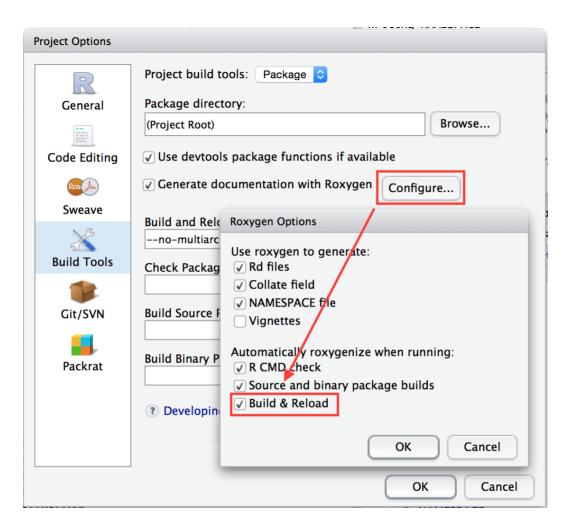


Image from <u>R Packages, ed. 2</u>





### Quoth Jenny Bryan:

- 1. Use functions.
- 2. A few little functions >> a monster function
- 3. Small well-named helper >> commented code

#### Helper functions

```
plot_daus <- function(daily_users) {
  daily_users <- daily_users %>%
    dplyr::mutate(date = as.Date(time)) %>%
    dplyr::group_by(date)
    dplyr::select(user_id) %>%
    dplyr::distinct() %>%
    dplyr::summarize(n = dplyr::n())

ggplot2::ggplot(ggplot2::aes(daily_users, x, n)) +
    ggplot2::geom_col()
}
```



#### Helper functions

```
plot daus <- function(daily users) {</pre>
  daily users <- count daus(daily users)</pre>
  ggplot2::ggplot(ggplot2::aes(daily users, x, n)) +
    ggplot2::geom col()
count daus <- function(daily users) {</pre>
  daily users %>%
    dplyr::mutate(date = as.Date(time)) %>%
    dplyr::group by(date)
    dplyr::select(user id) %>%
    dplyr::distinct() %>%
    dplyr::summarize(n = dplyr::n())
```



#### **Show of Hands**

## Which of these functions will be added to NAMESPACE?

```
#' Plot daily active users
#'
#' @param ...
#' @export
plot_daus <- function(...) {
    # ... code to plot daily active users
}

#' Count daily active users
#'
#' @param ...
count_daus <- function(...) {
    # ... code to count daily active users
}</pre>
```

#### **Show of Hands**

# Which of these functions will be added to NAMESPACE?

```
#' Plot daily active users
#'
#' @param ...
#' @export
plot_daus <- function(...) {
    # ... code to plot daily active users
}

#' Count daily active users
#'
#' @param ...
count_daus <- function(...) {
    # ... code to count daily active users
}</pre>
```

## Exported functions vs internal functions

```
@export =
shinRa::plot_daus()

library(shinRa)

plot_daus() ✓
```



## Exported functions vs internal functions



# Strategies for documenting helper functions:

- 1. Don't document them 🐄
- 2. @keywords internal
- 3. @noMd

```
#' [other roxygen code]
#' @param x The name of a database to retrieve
get_data <- function(x) {
    # code to get data
}

#' [other roxygen code]
#' @param x @inheritParam get_data
filter_table <- function(x) {
    tbl <- get_data(x)
    # code to filter data
}</pre>
```



```
#' [other roxygen code]
#' @param x The name of a database to retrieve
get_data <- function(x) {
    # code to get data
}

#' @rdname get_data
#' @export
filter_table <- function(x) {
    tbl <- get_data(x)
    # code to filter data
}</pre>
```



```
#' [other roxygen code]
#' @param x The name of a database to retrieve
#' @name data helpers
NULL
#' @rdname data helpers
#' @export
get data <- function(x) {</pre>
  # code to get data
#' @rdname data helpers
#' @export
filter table <- function(x) {
  tbl <- get data(x)
  # code to filter data
```



```
#' [other roxygen code]
#' @param x The name of a database to retrieve
#' @name data helpers
NULL
#' @rdname data helpers
#' @export
get data <- function(x) {</pre>
  # code to get data
#' @rdname data helpers
#' @export
filter table <- function(x) {
 tbl <- get data(x)
  # code to filter data
```



#### Your Turn 3

In R/themes.R, join the documentation of theme\_avalanche\_h() and theme\_avalanche\_v() to theme\_avalanche() by replacing the roxygen code for the first two functions with "#' @rdname theme\_avalanche".

Make sure both functions still have an export tag, as well!

Re-render the documentation and read the help page for ? theme\_avalanche\_h()



#### Your Turn 3

```
#' @rdname theme avalanche
#' @export
theme avalanche h <- function (base size = 14, ...) {
 ggplot2::theme minimal(base size = base size, ...) +
   gaplot2::theme(
     panel.grid.minor = ggplot2::element blank(),
     panel.grid.major.x = ggplot2::element blank()
#' @rdname theme avalanche
#' @export
theme_avalanche_v <- function(base size = 14, ...) {
 ggplot2::theme minimal(base size = base size, ...) +
   gaplot2::theme(
     panel.grid.minor = ggplot2::element blank(),
     panel.grid.major.y = ggplot2::element blank()
```

#### Package documentation

```
use_package_doc()
__
```

usethis also likes this for

```
use_import_from() and other metadata
```



#### help("tidyr")

tidyr-package {tidyr}

#### tidyr: Tidy Messy Data

#### Description

Tools to help to create tidy data, where each column is a variable, each row is an observation, and each cell contains a single value. 'tidyr' contains tools for changing the shape (pivoting) and hierarchy (nesting and 'unnesting') of a dataset, turning deeply nested lists into rectangular data frames ('rectangling'), and extracting values out of string columns. It also includes tools for working with missing values (both implicit and explicit).

Author(s)

Maintainer: Hadley Wickham hadley@rstudio.com

Authors:

• Lionel Henry lionel@rstudio.com

Other contributors:

· RStudio [copyright holder]

See Also

Useful links:

- · https://tidyr.tidyverse.org
- https://github.com/tidyverse/tidyr
- Report bugs at <a href="https://github.com/tidyverse/tidyr/issues">https://github.com/tidyverse/tidyr/issues</a>

[Package tidyr version 1.0.0 Index]





### pkgdown





#### pkgdown



```
use_pkgdown(),
use_pkgdown_github_pages()
```

Renders documentation, README, vignettes, and more as a website.



## Here's a good example



https://roxygen2.r-lib.org/

There are a lot more documentation tricks.

Read the vignettes!

