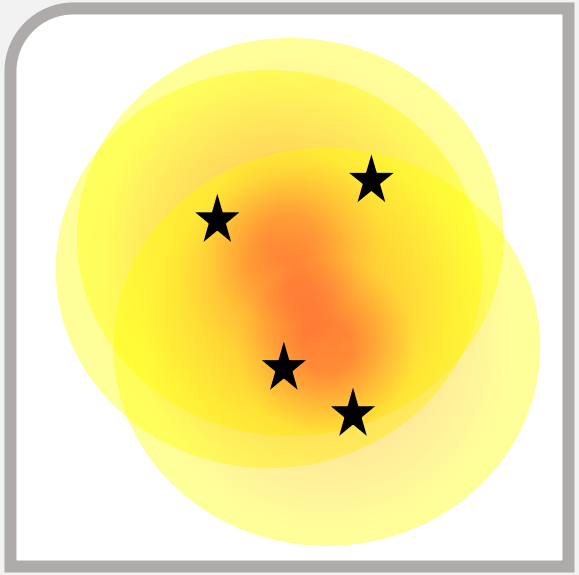
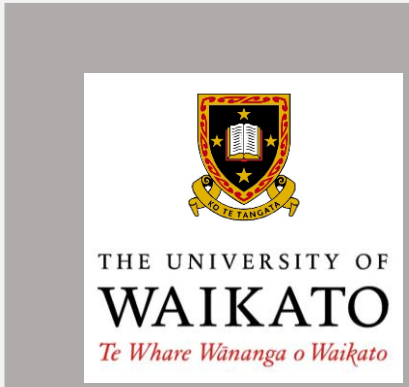
Writing R packages

How to write R packages that make life easier for you yourself and for others





NSCR Workshop 9 June 2022

Sophie Curtis-Ham

OVERVIEW

1. Why to write a package
2. How to write a package
   - Resources
   - Workflow overview
   - Demonstration

× Github integration

× CRAN submission

× Adding package data

× Testing in depth

× ReadMe file

WHY?

- ✓ Reduce duplication for yourself
- ✓ Future you will thank past you for tidy, well documented code
- ✓ Share code with others
- ✓ Congratulations, you're now a software developer

… and it's easier than you'd think!

WHY?

https://github.com/Sophie-c-h/gpsmartr

Person Identifier

Suspect Rank

1 — 2,107

1 212 423 634 845 1,056 1,267 1,478 1,689 1,900 2,107

Node Type

☐ home ☐ family_immediate ☐ family_ip ☐ family_other ☐ school
☐ work ☐ offence ☐ victim_witness ☐ incident ☐ police_other

Frequency

☐ freq_weekly ☐ freq_monthly ☐ freq_yearly

Recency

☐ rec_1to2d ☐ rec_3to30d ☐ rec_1to12m ☐ rec_1to5y
☐ rec_over5y

Duraction

☐ dur_1to2d ☐ dur_3to30d ☐ dur_1to12m ☐ dur_1to5y
☐ dur_over5y

Same crime

☐ 0 ☐ 1

Same location type

☐ 0 ☐ 1

Same daypart

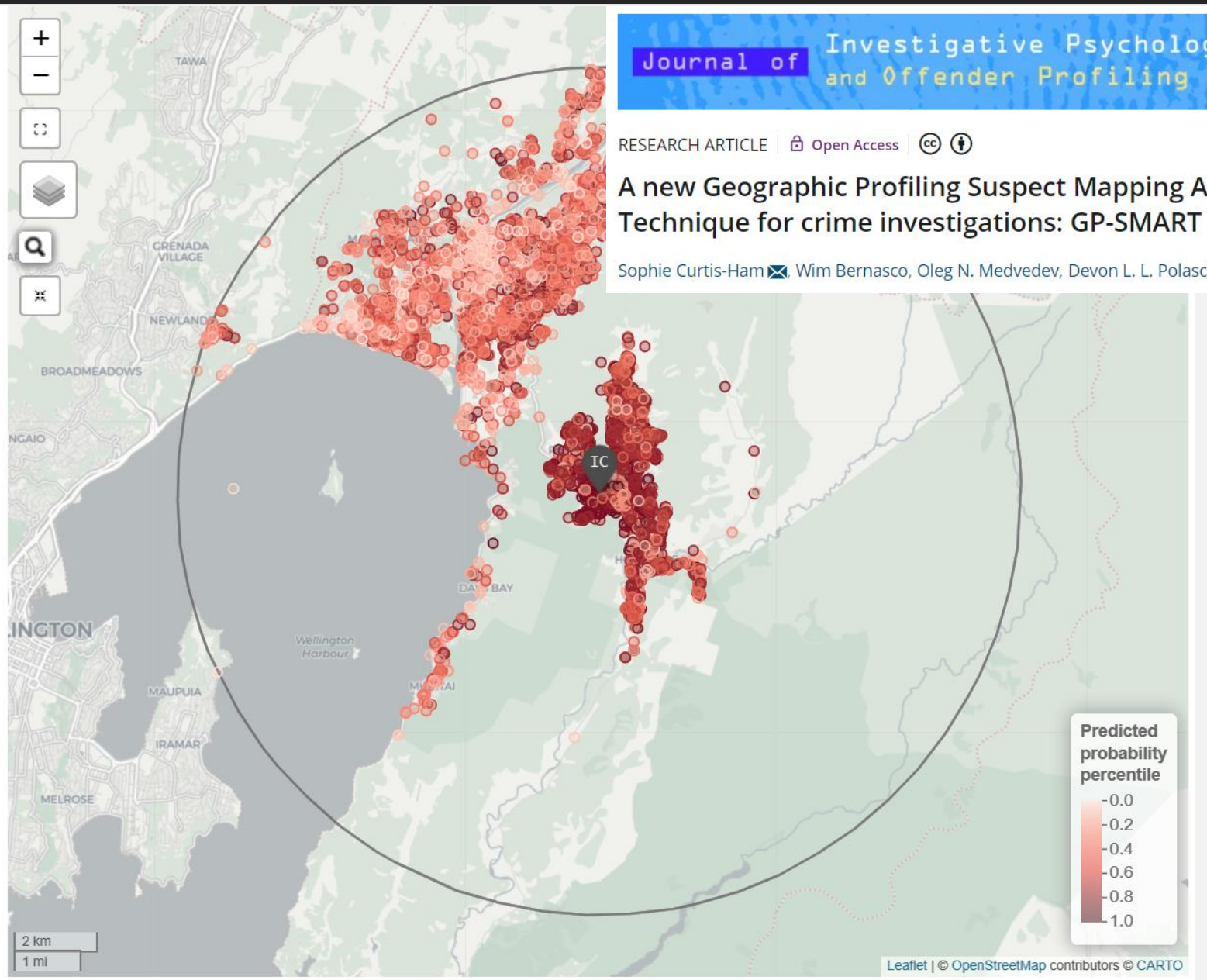
☐ 0 ☐ 1

Same_weekpart

☐ 0 ☐ 1

Same season

☐ 0 ☐ 1



Journal of Investigative Psychology and Offender Profiling

RESEARCH ARTICLE | Open Access

A new Geographic Profiling Suspect Mapping And Ranking Technique for crime investigations: GP-SMART

Sophie Curtis-Ham, Wim Bernasco, Oleg N. Medvedev, Devon L. L. Polaschek

HOW - RESOURCES

Videos

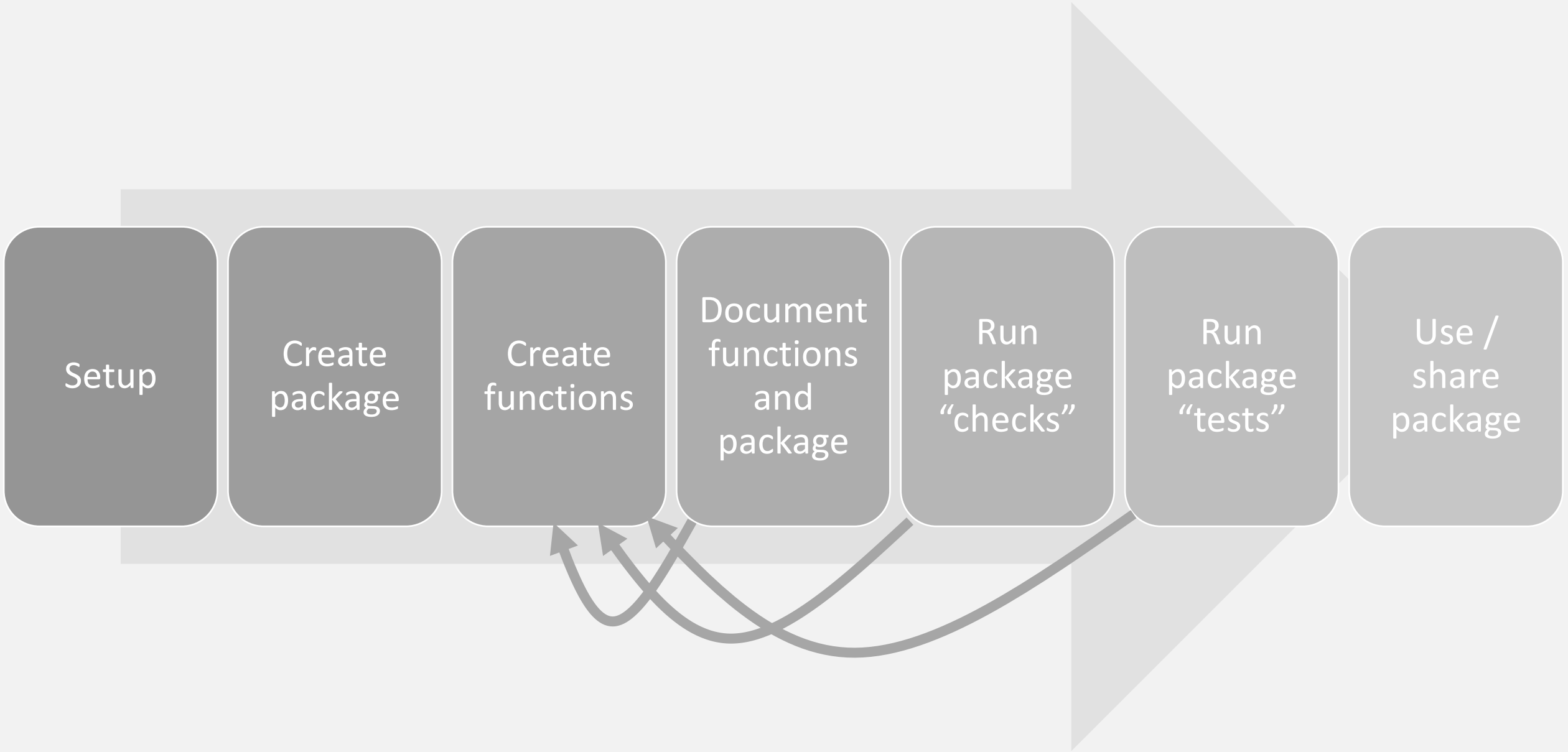
- https://datawarrior.wordpress.com/2016/09/08/developing-r-packages/ quick video tutorial
- https://www.youtube.com/watch?v=47PN2VG3RmI&ab_channel=StatistikinDD create a package in 2 minutes!
- https://www.rstudio.com/resources/rstudioconf-2018/you-can-**make-a-package-in-20-minutes**-jim-hester/
- https://www.youtube.com/watch?v=xcXzaEmZ-m4 and accompanying blog https://www.pipinghotdata.com/posts/2020-10-25-your-first-**r-package-in-1-hour/**
- https://www.youtube.com/watch?v=OIirKRgIsdc another hour tutorial
- https://www.youtube.com/watch?v=1ZrjWKcG1C4&ab_channel=RStudio testthat tutorial

HOW - RESOURCES

Web books/blogs

- https://hilaryparker.com/2014/04/29/writing-an-r-package-from-scratch/ blog with a nice simple example
- https://r-pkgs.org/index.html Hadley Wickham and Jenny Bryan's detailed guide to developing packages
- https://support.rstudio.com/hc/en-us/articles/200486488-Developing-Packages-with-the-RStudio-IDE RStudio guide
- https://cran.r-project.org/doc/manuals/R-exts.html#Data-in-packages R Core Team very detailed guide
- https://bookdown.org/rdpeng/RProgDA/building-r-packages.html Roger Peng chapters on packages and mapping functions
- https://style.tidyverse.org/documentation.html tidyverse style guide - covers both code and documentation
- https://cran.r-project.org/web/packages/roxygen2/vignettes/rd-formatting.html roxygen documentation guide
- https://www.r-bloggers.com/2019/03/unit-tests-in-r/ more on constructing tests

HOW - WORKFLOW



HOW - DEMONSTRATION

Setup

Make sure you have these packages installed:

- devtools
- usethis
- tidyverse
- rlang
- broom
- here

For the demo, we will loosely follow these instructions (but create different functions):

https://www.pipinghotdata.com/posts/2020-10-25-your-first-r-package-in-1-hour/

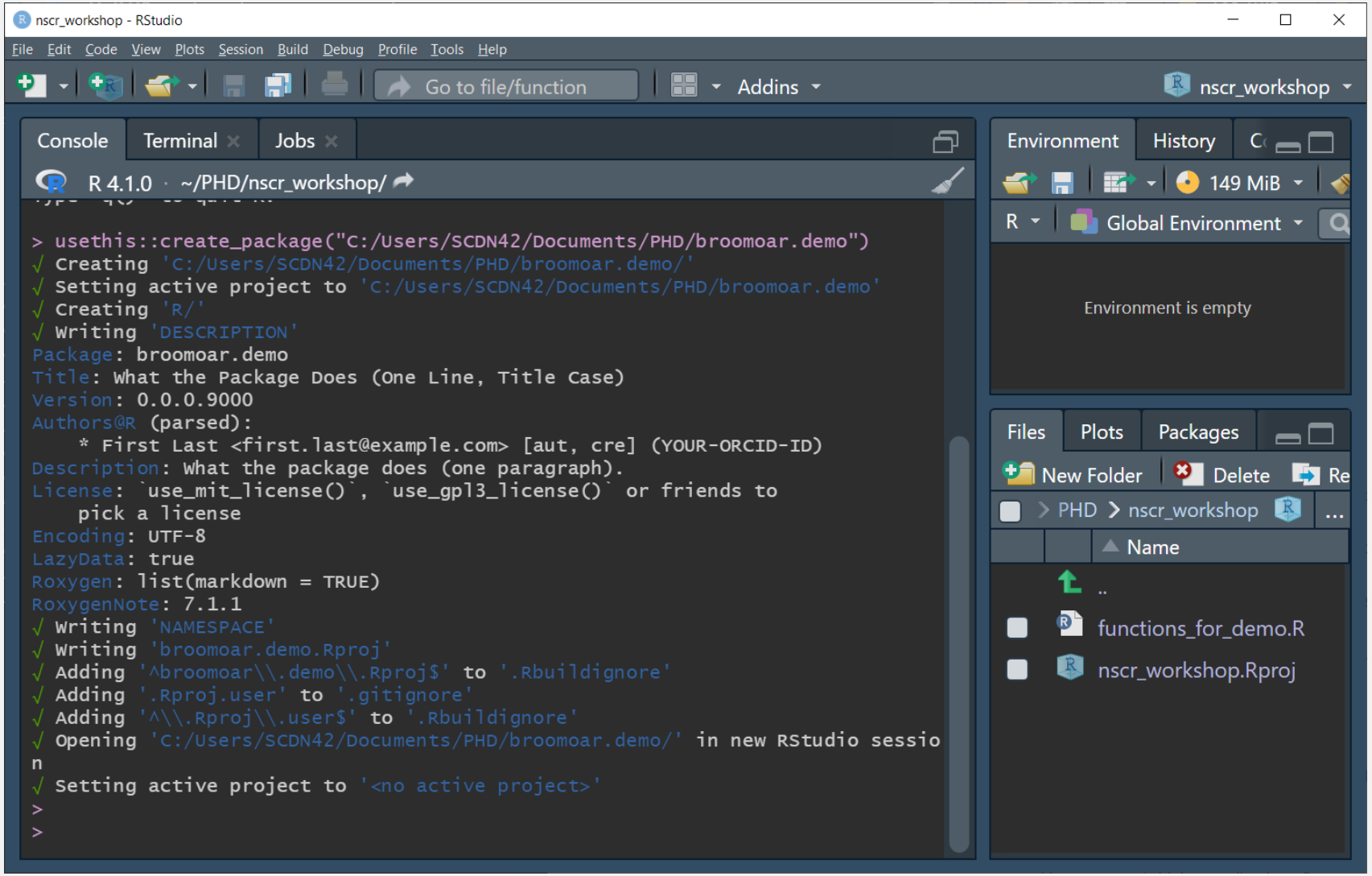
HOW - DEMONSTRATION

Grey background = enter in console

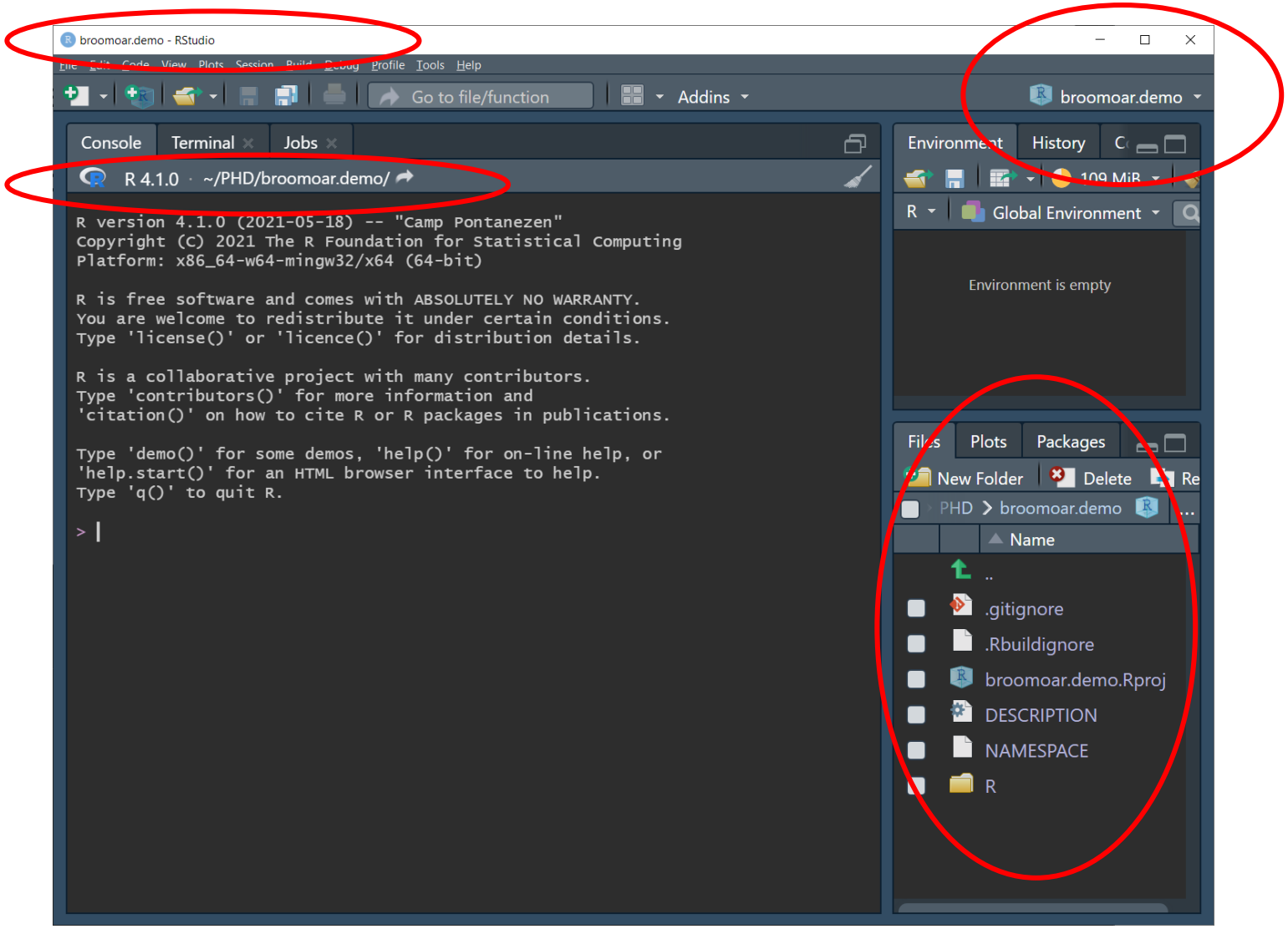
```
here::here()
usethis::create_package("path/to/package/broomoar.demo")
```

Create

package



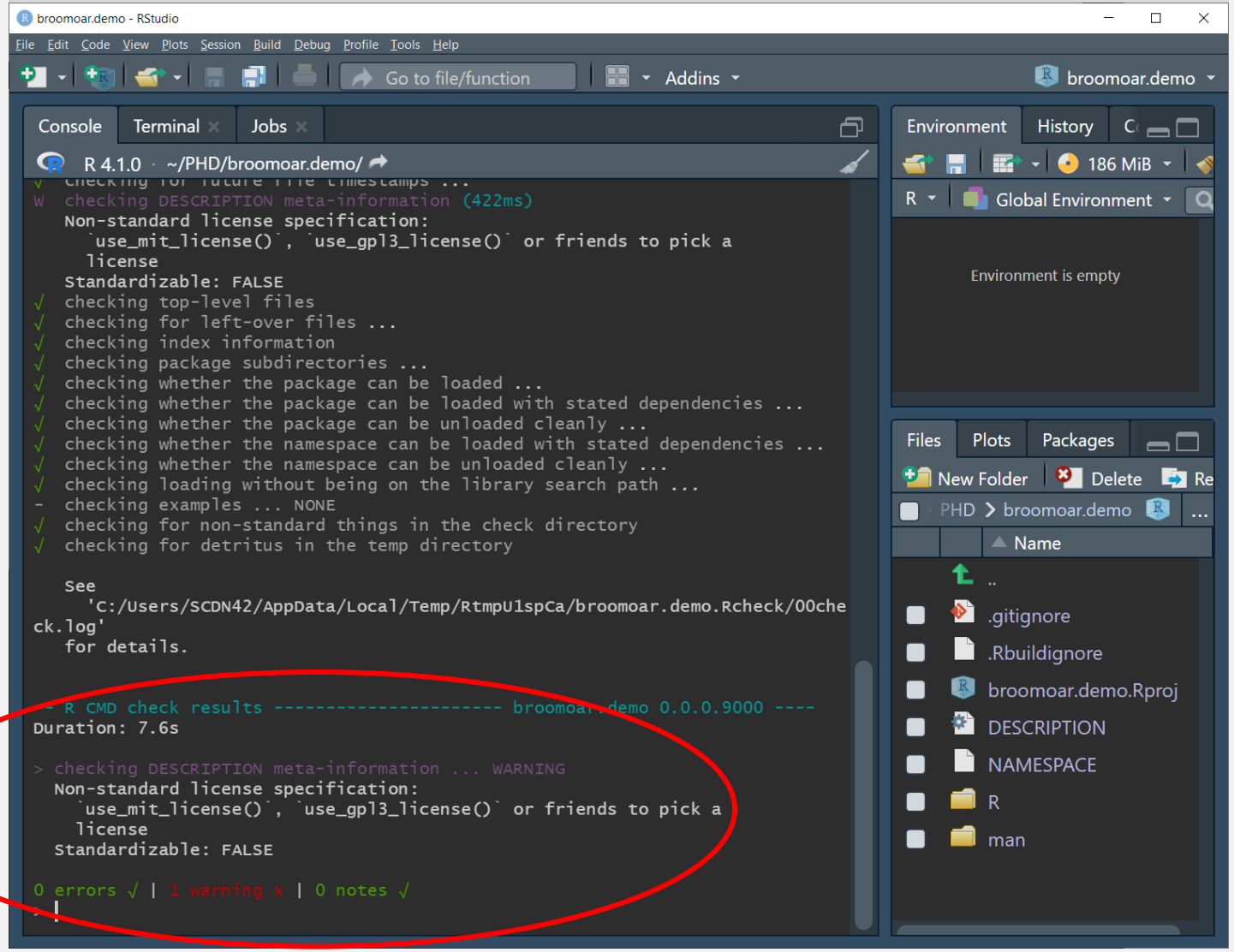
HOW - DEMONSTRATION



HOW - DEMONSTRATION

Run check

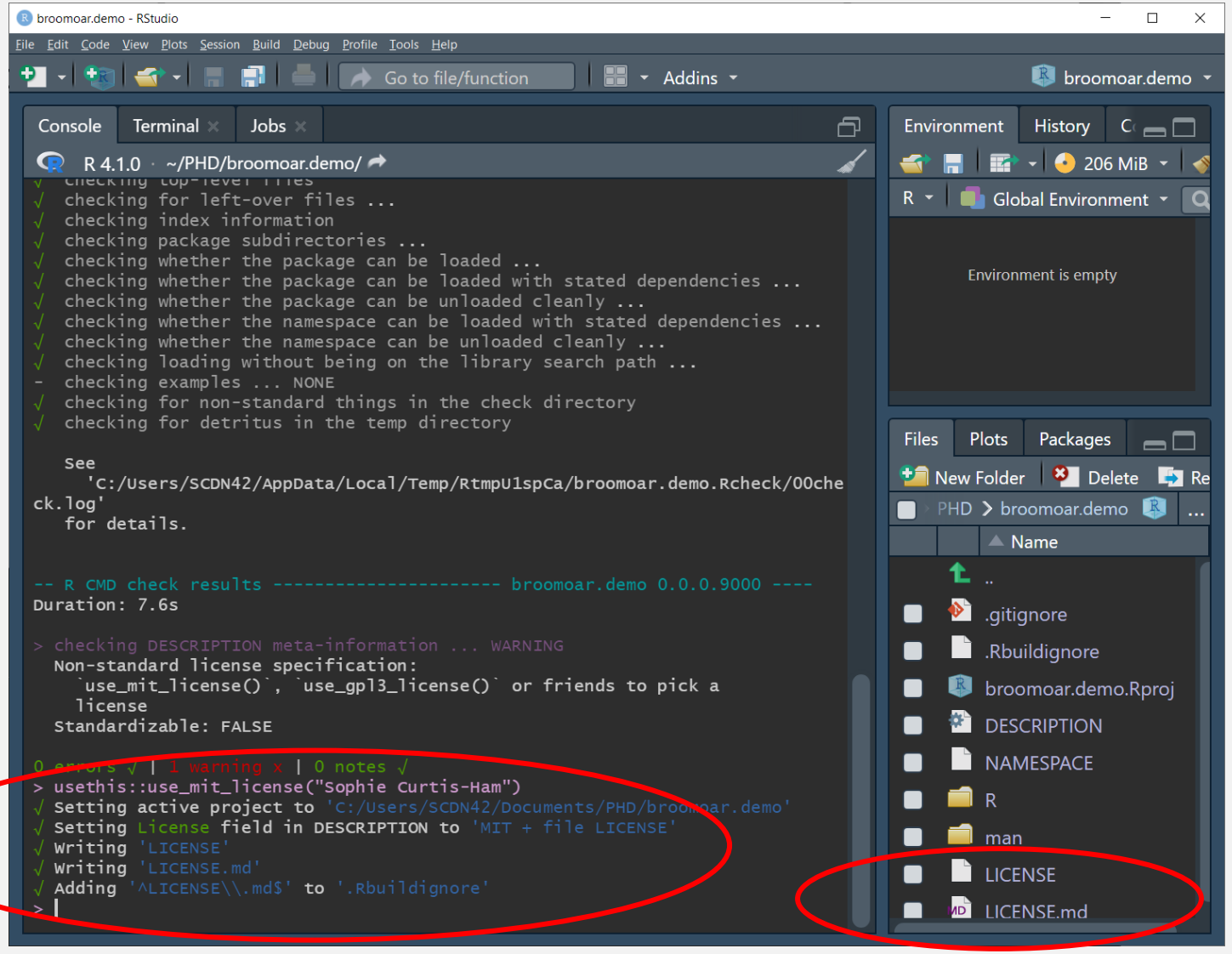
```
devtools::check()
```



HOW - DEMONSTRATION

Add licence

```
usethis::use_mit_license("Your Name")
```



HOW - DEMONSTRATION

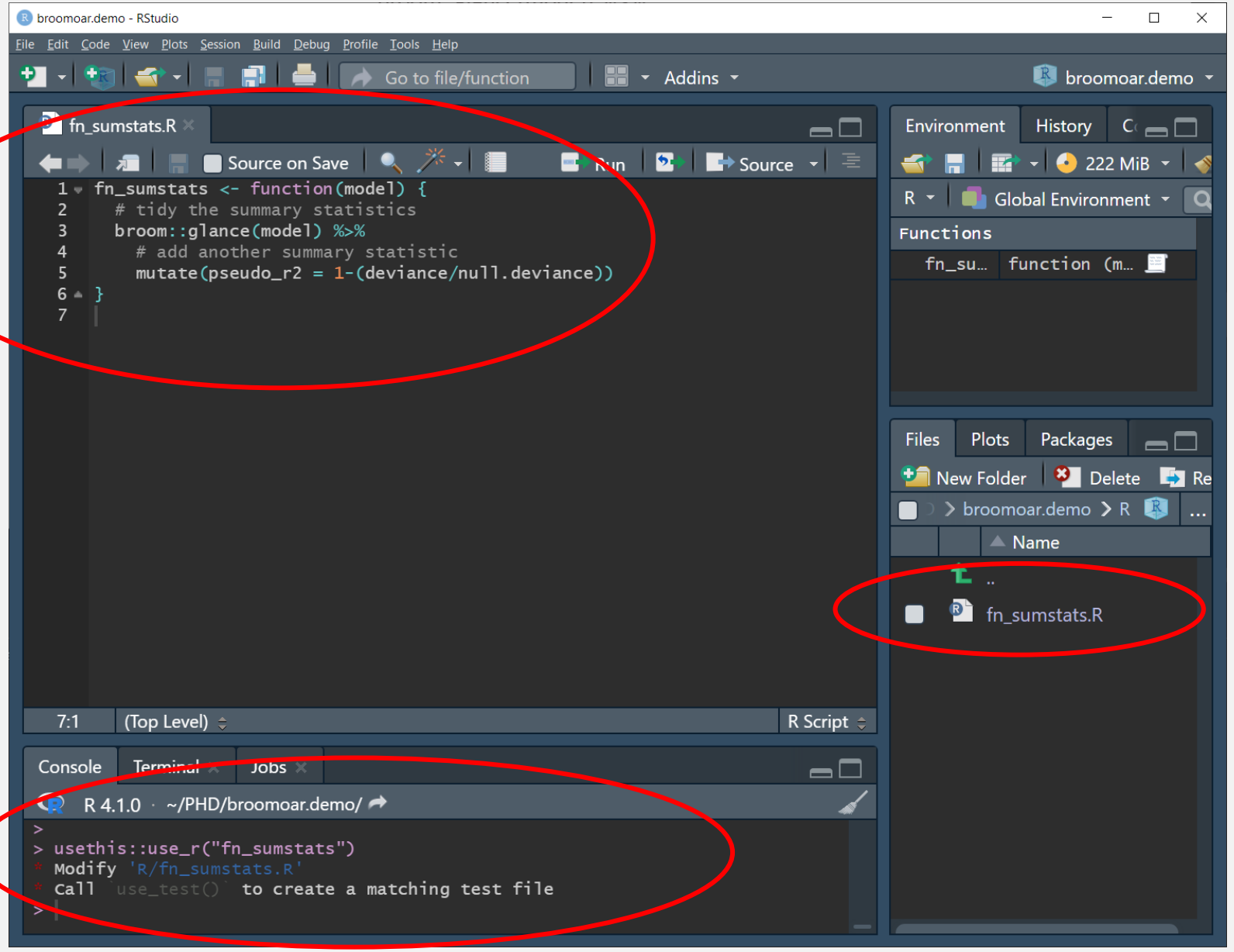
Create

function

```
fn_sumstats <- function(model) {
       # tidy the summary statistics
       broom::glance(model) %>%
       # add another summary statistic
       mutate(pseudo_r2 = 1-(deviance/null.deviance))
       }
usethis::use_r("fn_sumstats")
```

THEN COPY AND PASTE THE FUNCTION TO THE NEW BLANK R SCRIPT
THEN SAVE THE SCRIPT

HOW - DEMONSTRATION

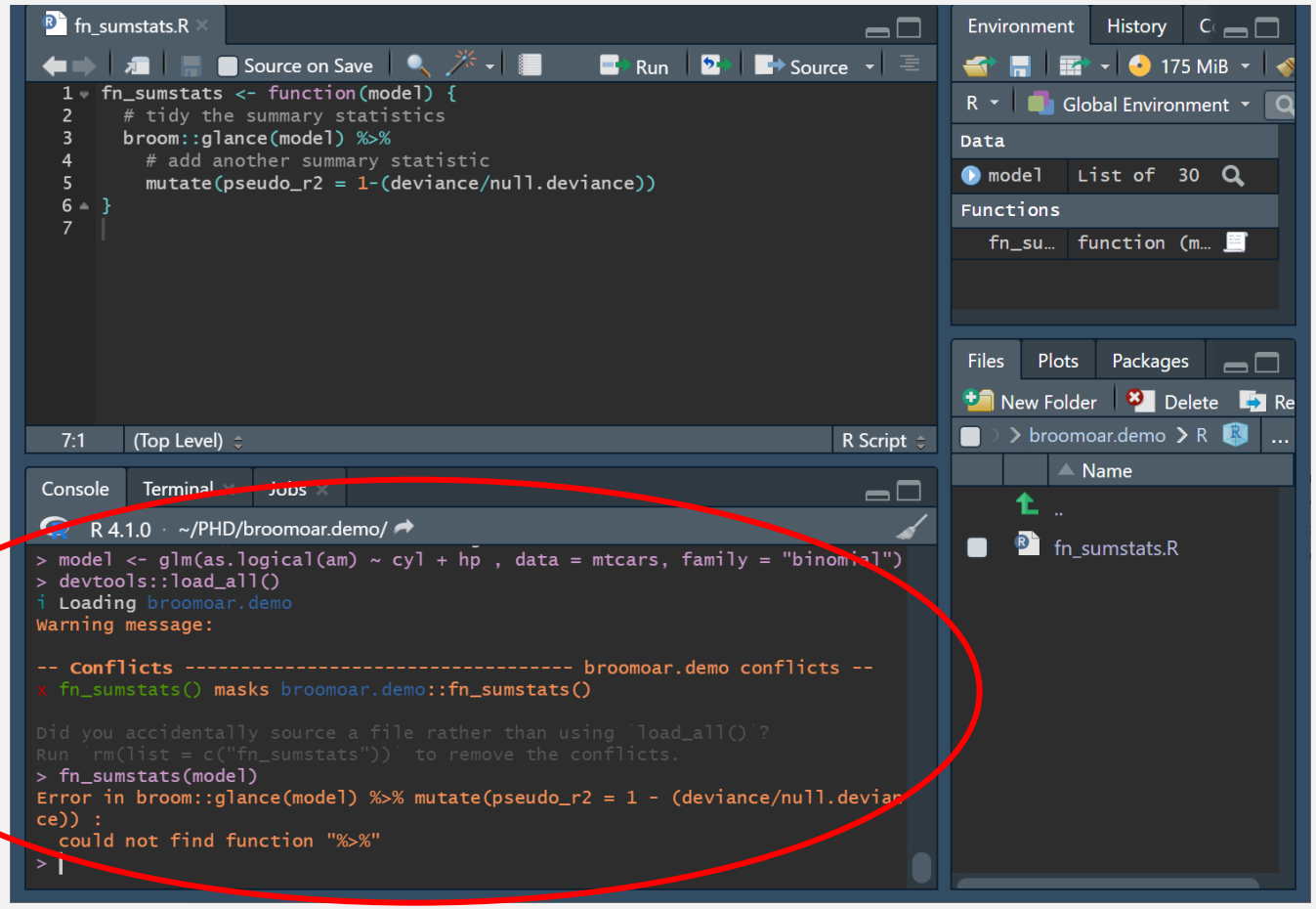


HOW - DEMONSTRATION

Execute

function

```
model <- glm(as.logical(am) ~ cyl + hp,
             data = mtcars, family = "binomial")
devtools::load_all()
fn_sumstats(model)
rm(fn_sumstats) # if you get the 'conflicts' error
```



HOW - DEMONSTRATION

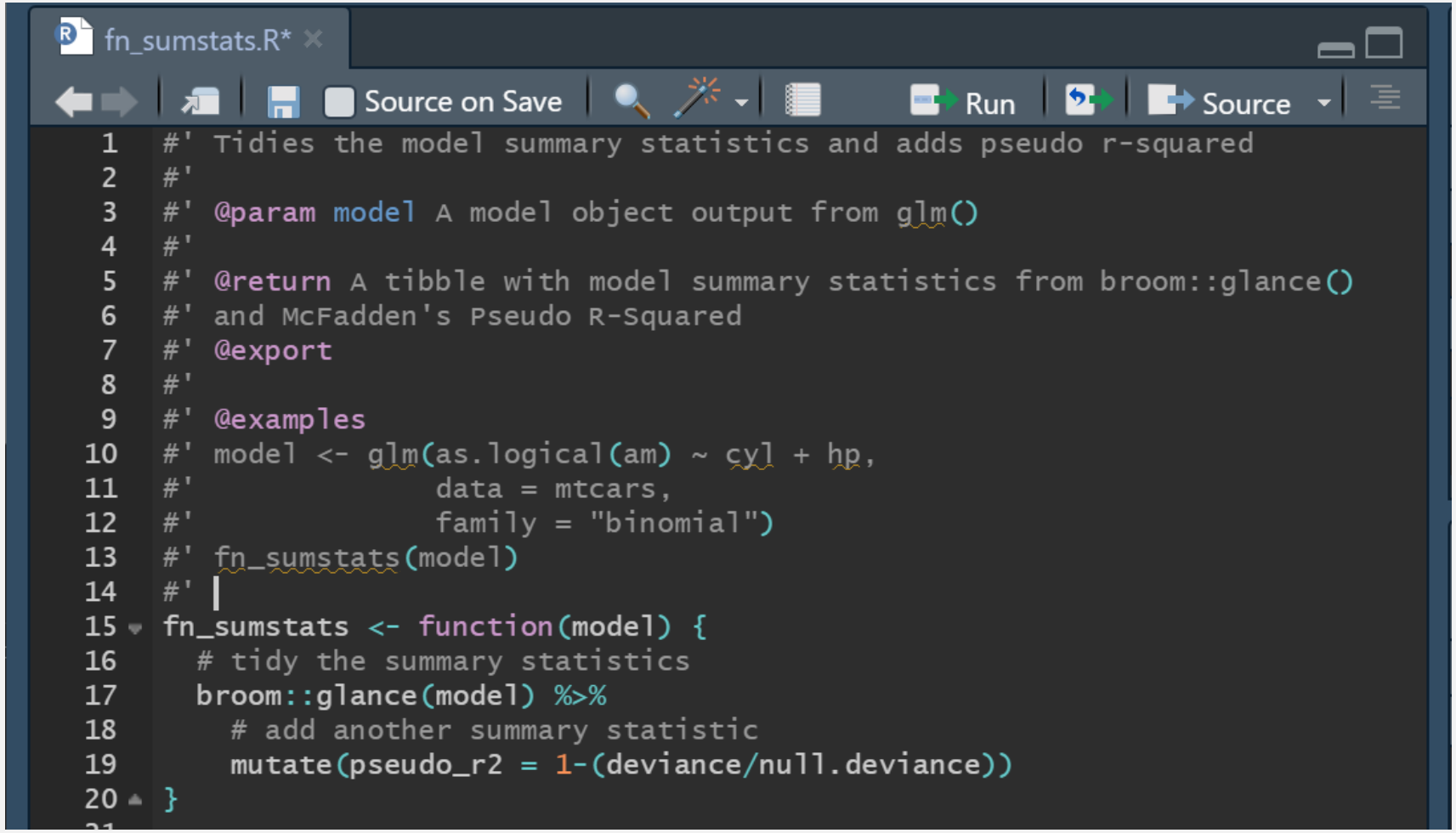
Document

function

PUT CURSOR INSIDE THE FUNCTION IN THE SCRIPT

Code > Insert Roxygen Skeleton

FILL IN DOCUMENTATION DETAILS

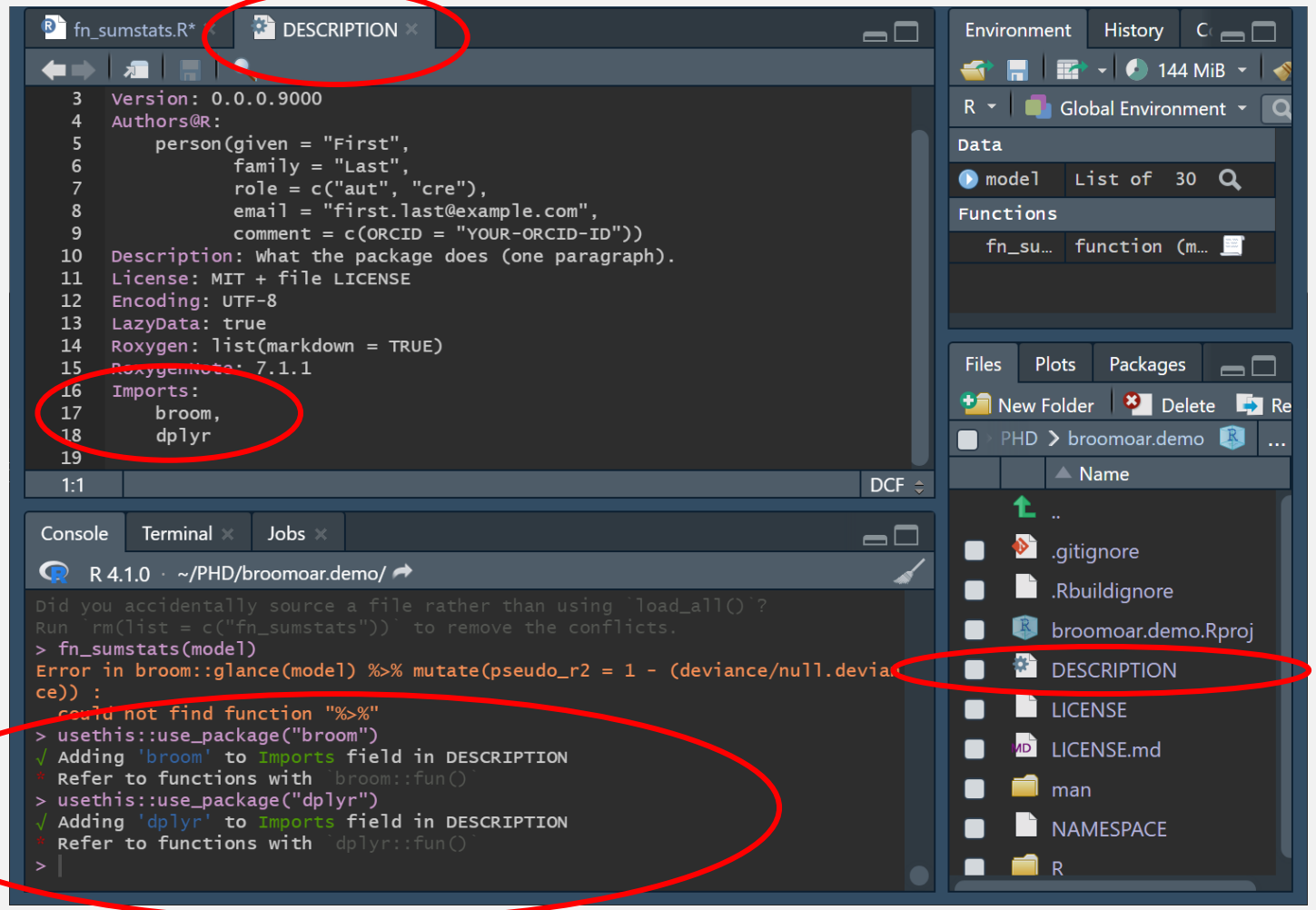


HOW - DEMONSTRATION

```
usethis::use_package("broom")
usethis::use_package("dplyr")
```

Document

function

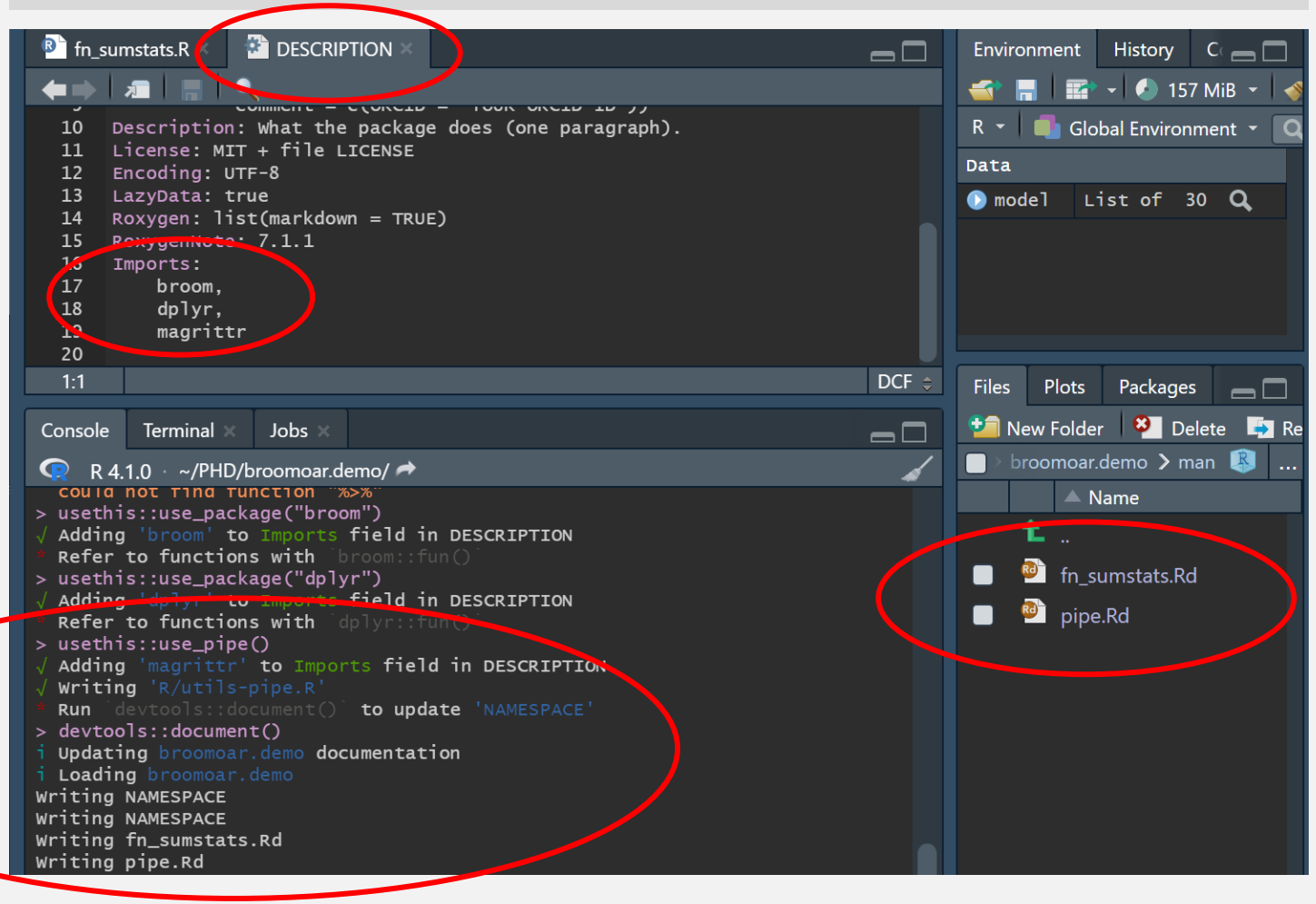


HOW - DEMONSTRATION

Document

function

```
usethis::use_pipe()
devtools::document()
```

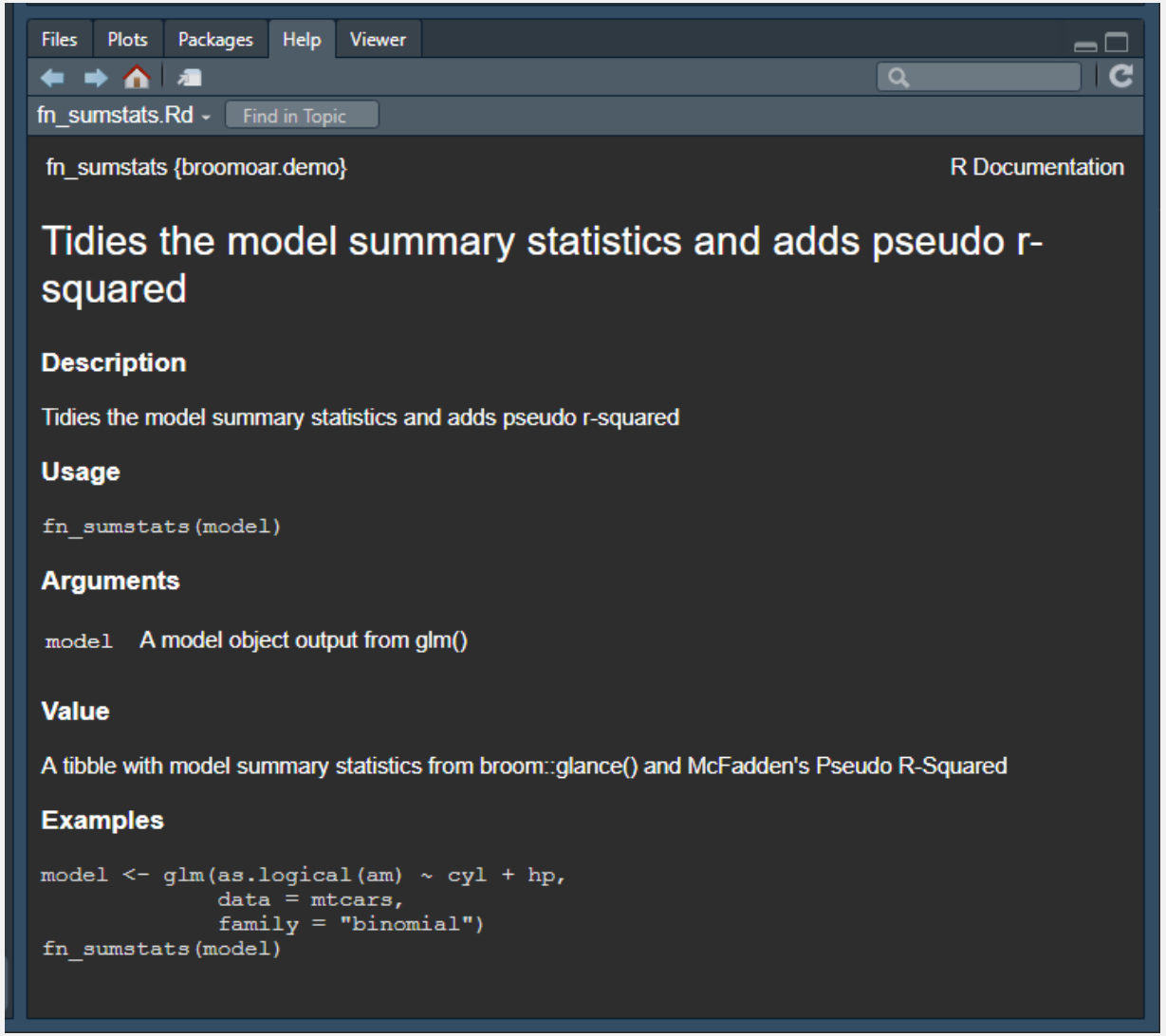


HOW - DEMONSTRATION

Document

function

```
?fn_sumstats
```



HOW - DEMONSTRATION

```r
#' Run GP-SMART for an input crime and input suspects
#'
#' @description
#' Calculates suspect variables relative to the input crime,
#' calculates probability of crime per suspect activity location (node) within the user's input search radius,
#' ranks suspects according to their highest probability activity node,
#' and returns ranked suspects and (optionally) their activity nodes for mapping.
#'
#' @param input_crime A data frame with 1 row and 16 columns, created by [fn_prepare_input_crime()].
#'
#' @param input_suspects A data frame with 1 row per suspect activity node and 21 columns, created by [fn_prepare_suspect_data()].
#'
#' @param search_radius The distance (in kilometres) around the input crime to search for suspect nodes. Suspects with nodes within this dist
#'
#' @param weights A list of 8 data frames, one per weighting variable (node attribute),
#'   specifying the weight to use for each value of each attribute for each input crime type.
#'   If `NULL`, the default, built-in weights will be used (see Details).
#'
#' @param return_node_predictions Logical. Whether to return per-node predictions for the shortlisted suspects, required as input for [fn_map
#'
#' @author Sophie Curtis-Ham
#'
#' @importFrom rlang .data
#' @importFrom utils data
#'
#' @rawNamespace import(data.table, except = c(month, hour, quarter, week, year, wday, second, minute, mday, yday, isoweek))
#'
#' @return A list with four elements:
#' \describe{
#'  \item{`suspects_ranked`}{A data frame with n rows = n suspects with nodes within the `search_radius` of the input crime,
#'    ranked by their maximum predicted crime probability.}
#'  \item{`node_predictions`}{If `return_node_predictions` = `TRUE`, a data frame with n rows = n suspect nodes within search radius,
#'    including all node attributes and predicted crime probability per node;
#'    if `return_node_predictions` = `FALSE`, a character string "no node predictions".}
#'  \item{`input_crime`}{The data frame entered as input for [fn_gpsmart()].
#'    It is included in the output for use in the [fn_map_gpsmart_output()] function.}
#'  \item{`search_radius`}{The search radius (in kilometres), entered as input for [fn_gpsmart()].
#'    It is included in the output for use in the [fn_map_gpsmart_output()] function.}
#' }
#'
#' @export
#'
#' @details
#' \itemize{
#'  \item{In calculating the probability of crime per suspect activity node,
#'    the function weights the node (adjusts the probability) based on its values across 7 attributes (weighting variables).
#'    The weighting variables are: frequency, recency, duration, behaviour similarity, location type similarity, daypart similarity, weekpart
#'    The default (built in) weights were estimated using a New Zealand sample of offenders and activity locations as described in Curtis-Ham
#'    To create your own weights, use the built in list of data frames as a template, replacing only the weight values.
#'    (use \code{gpsmartr.private:::node_weights_lookups} to access the built in weights).}
#'  \item{The function will return an error if there are no suspect nodes within the search radius.}
#'  \item{Season variables are based on Southern Hemisphere.}
#' }
#'
#' @seealso
#' * [fn_prepare_input_crime()] checks that the minimum necessary variables are present and creates the `input_crime` data frame for use in t
#' * [fn_prepare_suspect_data()] checks that the minimum necessary variables are present and creates the `input_suspects` data frame for use
#' * [fn_map_gpsmart_output()] creates an interactive map visualising the output of this function.
#'
#' @references
#' Curtis-Ham S., Bernasco, W., Medvedev, O. N., & Polaschek, D. L. L (2022).
#'   'A new geographic profiling method for mapping and ranking suspects in crime investigations: GP-SMART'.
#'   Journal of Investigative Psychology and Offender Profiling. https://doi.org/10.1002/jip.1585
#'
#' @examples
#' \dontrun{
#' data(example_input_crime_raw)
#' data(example_input_suspects_raw)
#' fn_gpsmart(
#'   input_crime = fn_prepare_input_crime(example_input_crime_raw),
#'   input_suspects = fn_prepare_suspect_data(example_input_suspects_raw),
#'   search_radius = 10,
#'   return_node_predictions = FALSE
#' )
#' }
#'
fn_gpsmart <- function(input_crime, input_suspects, search_radius = 10, weights = NULL, return_node_predictions = FALSE) {
```

fn_gpsmart.Rd

fn_gpsmart {gpsmartr} R Documentation

Run GP-SMART for an input crime and input suspects

Description

Calculates suspect variables relative to the input crime, calculates probability of crime per suspect activity location (node) within the user's input search radius, ranks suspects according to their highest probability activity node, and returns ranked suspects and (optionally) their activity nodes for mapping.

Usage

```
fn_gpsmart(
  input_crime,
  input_suspects,
  search_radius = 10,
  weights = NULL,
  return_node_predictions = FALSE
)
```

Arguments

input_crime	A data frame with 1 row and 16 columns, created by fn_prepare_input_crime().
input_suspects	A data frame with 1 row per suspect activity node and 21 columns, created by fn_prepare_suspect_data().
search_radius	The distance (in kilometres) around the input crime to search for suspect nodes. Suspects with nodes within this distance are shortlisted. The default is 10km.
weights	A list of 8 data frames, one per weighting variable (node attribute), specifying the weight to use for each value of each attribute for each input crime type. If NULL, the default, built-in weights will be used (see Details).
return_node_predictions	Logical. Whether to return per-node predictions for the shortlisted suspects, required as input for fn_map_gpsmart_output().

Details

- In calculating the probability of crime per suspect activity node, the function weights the node (adjusts the probability) based on its values across 7 attributes (weighting variables). The weighting variables are: frequency, recency, duration, behaviour similarity, location type similarity, daypart similarity, weekpart similarity and season similarity. The default (built in) weights were estimated using a New Zealand sample of offenders and activity locations as described in Curtis-Ham et al (2022). To create your own weights, use the built in list of data frames as a template, replacing only the weight values. (use gpsmartr.private:::node_weights_lookups to access the built in weights).
- The function will return an error if there are no suspect nodes within the search radius.
- Season variables are based on Southern Hemisphere.

Value

A list with four elements:

suspects_ranked

A data frame with n rows = n suspects with nodes within the search_radius of the input crime, ranked by their maximum predicted crime probability.

node_predictions

If return_node_predictions = TRUE, a data frame with n rows = n suspect nodes within search radius, including all node attributes and predicted crime probability per node; if return_node_predictions = FALSE, a character string "no node predictions".

input_crime

The data frame entered as input for fn_gpsmart(). It is included in the output for use in the fn_map_gpsmart_output() function.

search_radius

The search radius (in kilometres), entered as input for fn_gpsmart(). It is included in the output for use in the fn_map_gpsmart_output() function.

Author(s)

Sophie Curtis-Ham

References

Curtis-Ham S., Bernasco, W., Medvedev, O. N., & Polaschek, D. L. L (2022). 'A new geographic profiling method for mapping and ranking suspects in crime investigations: GP-SMART'. Journal of Investigative Psychology and Offender Profiling. https://doi.org/10.1002/jip.1585

See Also

- fn_prepare_input_crime() checks that the minimum necessary variables are present and creates the input_crime data frame for use in this function.
- fn_prepare_suspect_data() checks that the minimum necessary variables are present and creates the input_suspects data frame for use in this function.
- fn_map_gpsmart_output() creates an interactive map visualising the output of this function.

Examples

```
## Not run:
data(example_input_crime_raw)
data(example_input_suspects_raw)
fn_gpsmart(
  input_crime = fn_prepare_input_crime(example_input_crime_raw),
  input_suspects = fn_prepare_suspect_data(example_input_suspects_raw),
  search_radius = 10,
  return_node_predictions = FALSE
)
```

HOW - DEMONSTRATION

FILL IN DESCRIPTION DETAILS

Document

```
devtools::document()
```

package

fn_sumstats.R | DESCRIPTION

```
1  Package: broomoar.demo
2  Title: Adds custom statistics to broom tidied model outputs
3  Version: 0.0.0.9000
4  Authors@R:
5      person(given = "Sophie",
6             family = "Curtis-Ham",
7             role = c("aut", "cre"),
8             email = "sc398@students.waikato.ac.nz",
9             comment = c(ORCID = "YOUR-ORCID-ID"))
10 Description: Adds pseudo-r2 to glance(); adds ORs and CIs and custom units to tidy().
11 License: MIT + file LICENSE
12 Encoding: UTF-8
13 LazyData: true
14 Roxygen: list(markdown = TRUE)
15 RoxygenNote: 7.1.1
16 Imports:
17     broom,
18     dplyr (>= 1.0.7),
19     magrittr (>= 2.0.1)
```

HOW - DEMONSTRATION

Run

check

```
devtools::check(cran = FALSE)
```

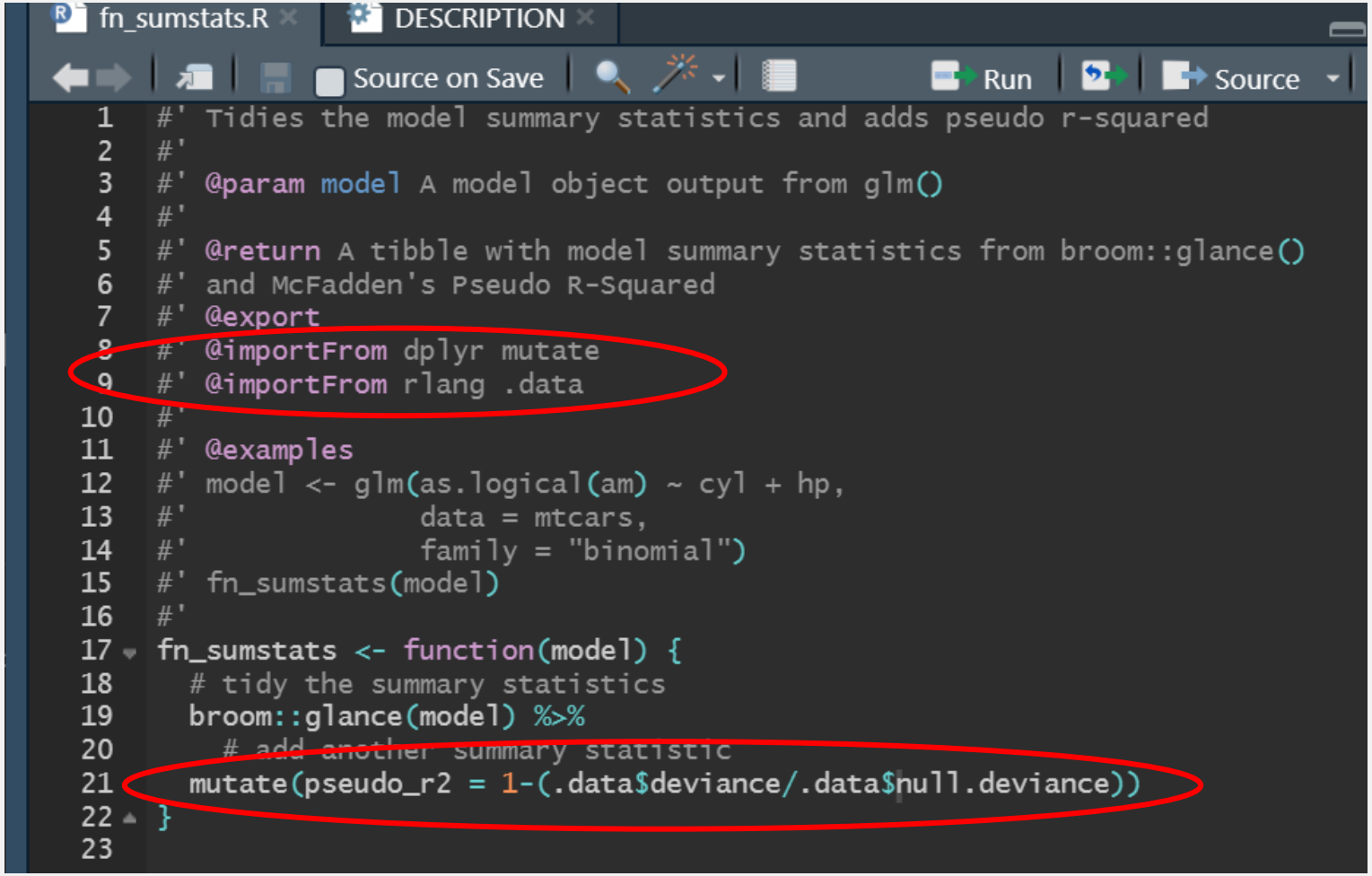
```
> checking examples ... ERROR
  Running examples in 'broomoar.demo-Ex.R' failed
  The error most likely occurred in:

  > base::assign(".ptime", proc.time(), pos = "CheckExEnv")
  > ### Name: fn_sumstats
  > ### Title: Tidies the model summary statistics and adds pseudo r-squared
  > ### Aliases: fn_sumstats
  >
  > ### ** Examples
  >
  > model <- glm(as.logical(am) ~ cyl + hp,
  +              data = mtcars,
  +              family = "binomial")
  > fn_sumstats(model)
  Error in mutate(., pseudo_r2 = 1 - (deviance/null.deviance)) :
    could not find function "mutate"
  Calls: fn_sumstats -> %>%
  Execution halted

> checking dependencies in R code ... NOTE
  Namespace in Imports field not imported from: 'dplyr'
    All declared Imports should be used.

> checking R code for possible problems ... NOTE
  fn_sumstats: no visible global function definition for 'mutate'
  fn_sumstats: no visible binding for global variable 'deviance'
  fn_sumstats: no visible binding for global variable 'null.deviance'
  Undefined global functions or variables:
    deviance mutate null.deviance
  Consider adding
    importFrom("stats", "deviance")
  to your NAMESPACE file.

1 error x | 0 warnings v | 2 notes x
```

HOW - DEMONSTRATION

Fix

issues

ADJUST ROXYGEN HEADER
ADD rlang TO DESCRIPTION IMPORTS [NOT SHOWN]
ADD .data$ TO FUNCTION



HOW - DEMONSTRATION

```
devtools::document()
devtools::check(cran = FALSE)
```

Console | Terminal | Jobs

R 4.1.0 · ~/PHD/broomoar.demo/

```
v  checking top-level files
v  checking for left-over files
v  checking index information
v  checking package subdirectories ...
v  checking R files for non-ASCII characters ...
v  checking R files for syntax errors ...
v  checking whether the package can be loaded (711ms)
v  checking whether the package can be loaded with stated dependencies (608ms)
v  checking whether the package can be unloaded cleanly (712ms)
v  checking whether the namespace can be loaded with stated dependencies (611ms)
v  checking whether the namespace can be unloaded cleanly (818ms)
v  checking loading without being on the library search path (699ms)
v  checking dependencies in R code (711ms)
v  checking S3 generic/method consistency (1s)
v  checking replacement functions (610ms)
v  checking foreign function calls (609ms)
v  checking R code for possible problems (2s)
v  checking Rd files ...
v  checking Rd metadata ...
v  checking Rd cross-references ...
v  checking for missing documentation entries (612ms)
v  checking for code/documentation mismatches (1.9s)
v  checking Rd \usage sections (1.2s)
v  checking Rd contents ...
v  checking for unstated dependencies in examples ...
v  checking examples (1.1s)


-- R CMD check results ------------------------ broomoar.demo 0.0.0.9000 ----
Duration: 22.6s

0 errors v | 0 warnings v | 0 notes v
>
```

HOW - DEMONSTRATION

Create

another

function

THEN COPY AND PASTE THE FUNCTION TO THE NEW BLANK R SCRIPT, THEN SAVE THE SCRIPT

```
fn_coefs <- function(model, vars_to_adjust = NULL, multiply_by = 1) {
 broom::tidy(model) %>%
  dplyr::rename(variable = .data$term) %>% # rename the variable column
  # adjust units, create OR and CIs, adjust decimals, relabel adjusted variable
  mutate(estimate = ifelse(.data$variable %in% vars_to_adjust,
                           .data$estimate*multiply_by,
                           .data$estimate),
         std.error = ifelse(.data$variable  %in% vars_to_adjust,
                            .data$std.error*multiply_by,
                            .data$std.error),
         conf.low = ifelse(.data$variable  %in% vars_to_adjust,
                           confint(model)[,1]*multiply_by,
                           confint(model)[,1]),
         conf.high  = ifelse(.data$variable  %in% vars_to_adjust,
                             confint(model)[,2]*multiply_by,
                             confint(model)[,2]),
         OR = exp(data$estimate),
         OR_CI_low = exp(.data$conf.low),
         OR_CI_high = exp(.data$conf.high),
         p.value = round(.data$p.value, 5), # round p-value
        variable = ifelse(.data$variable %in% vars_to_adjust,
                          paste0(.data$variable, " (per ", multiply_by, ")"),
                          .data$variable))
  }
usethis::use_r("fn_coefs")
```

HOW - DEMONSTRATION

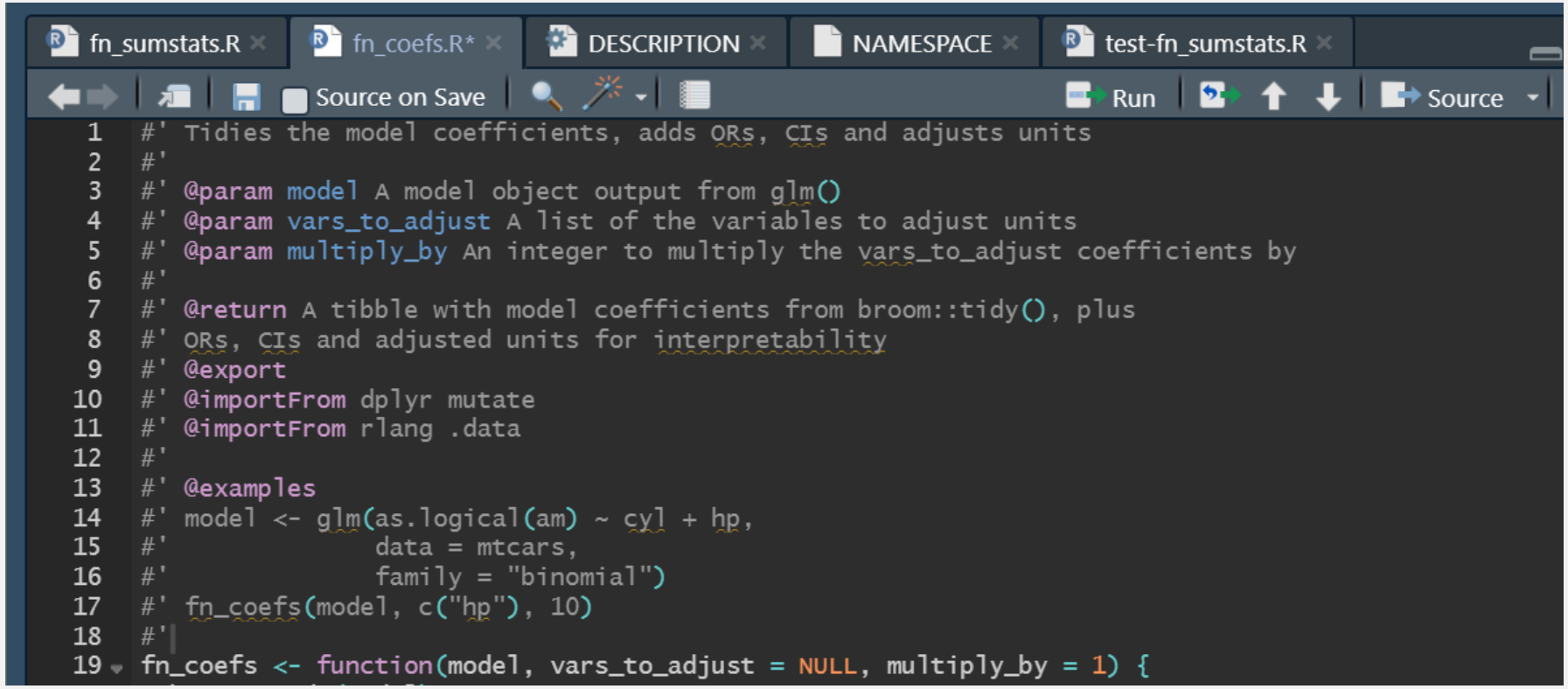
Document

function

PUT CURSOR INSIDE THE FUNCTION IN THE SCRIPT

Code > Insert Roxygen Skeleton

FILL IN DOCUMENTATION DETAILS

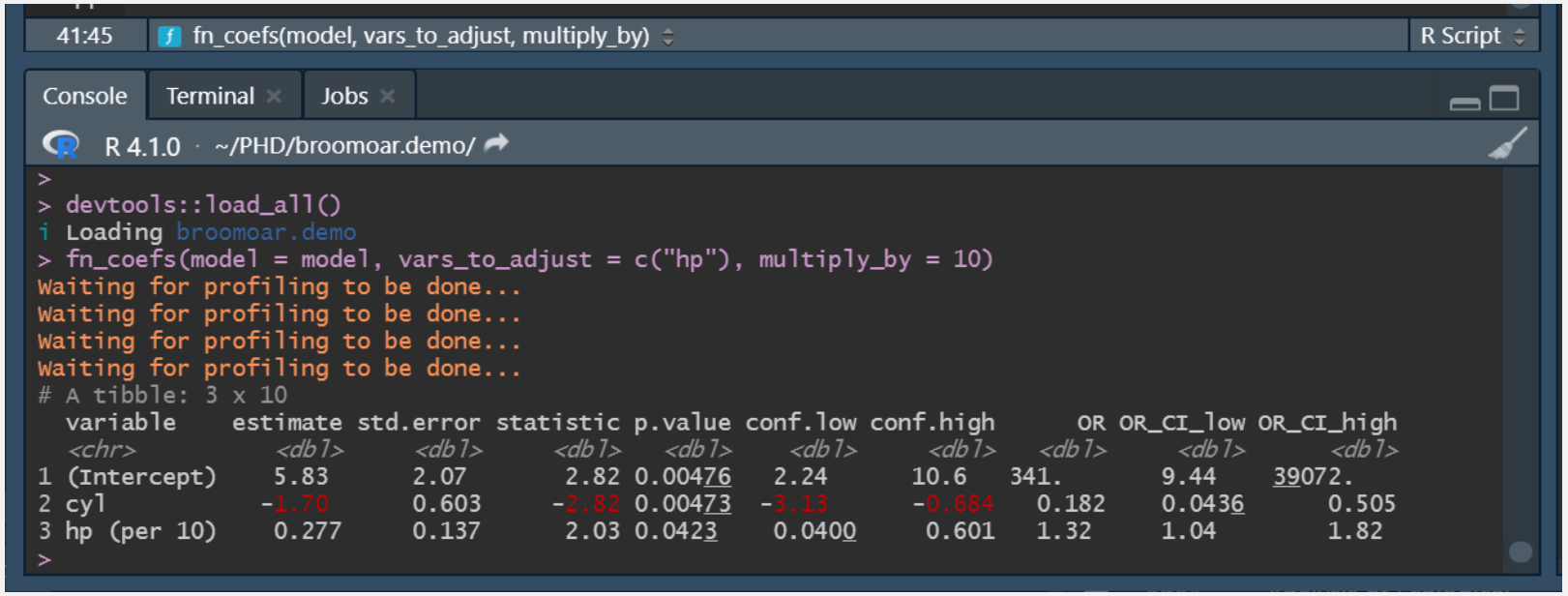


HOW - DEMONSTRATION

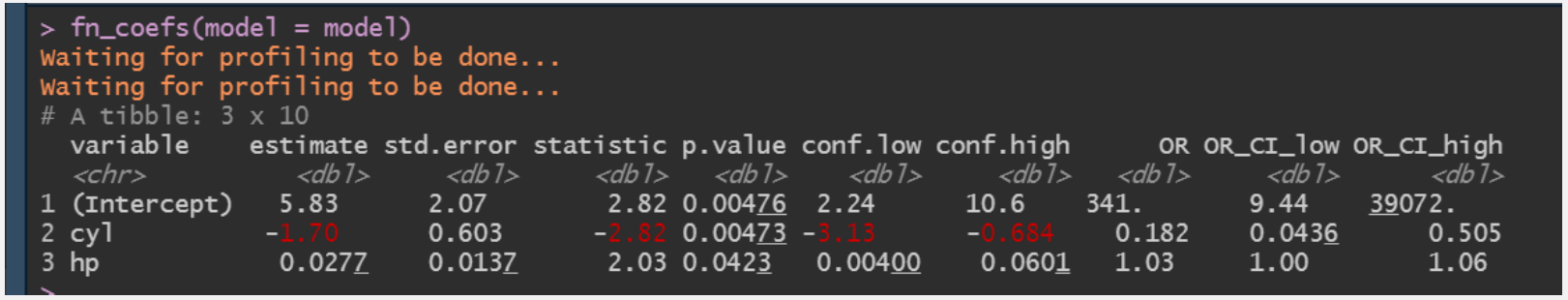
Execute

function

```
model <- glm(as.logical(am) ~ cyl + hp,
             data = mtcars, family = "binomial")
rm(fn_coefs) # prevents the 'conflicts' error
devtools::load_all()
fn_coefs(model = model, vars_to_adjust = c("hp"), multiply_by = 10)
```




```
fn_coefs(model = model)
```

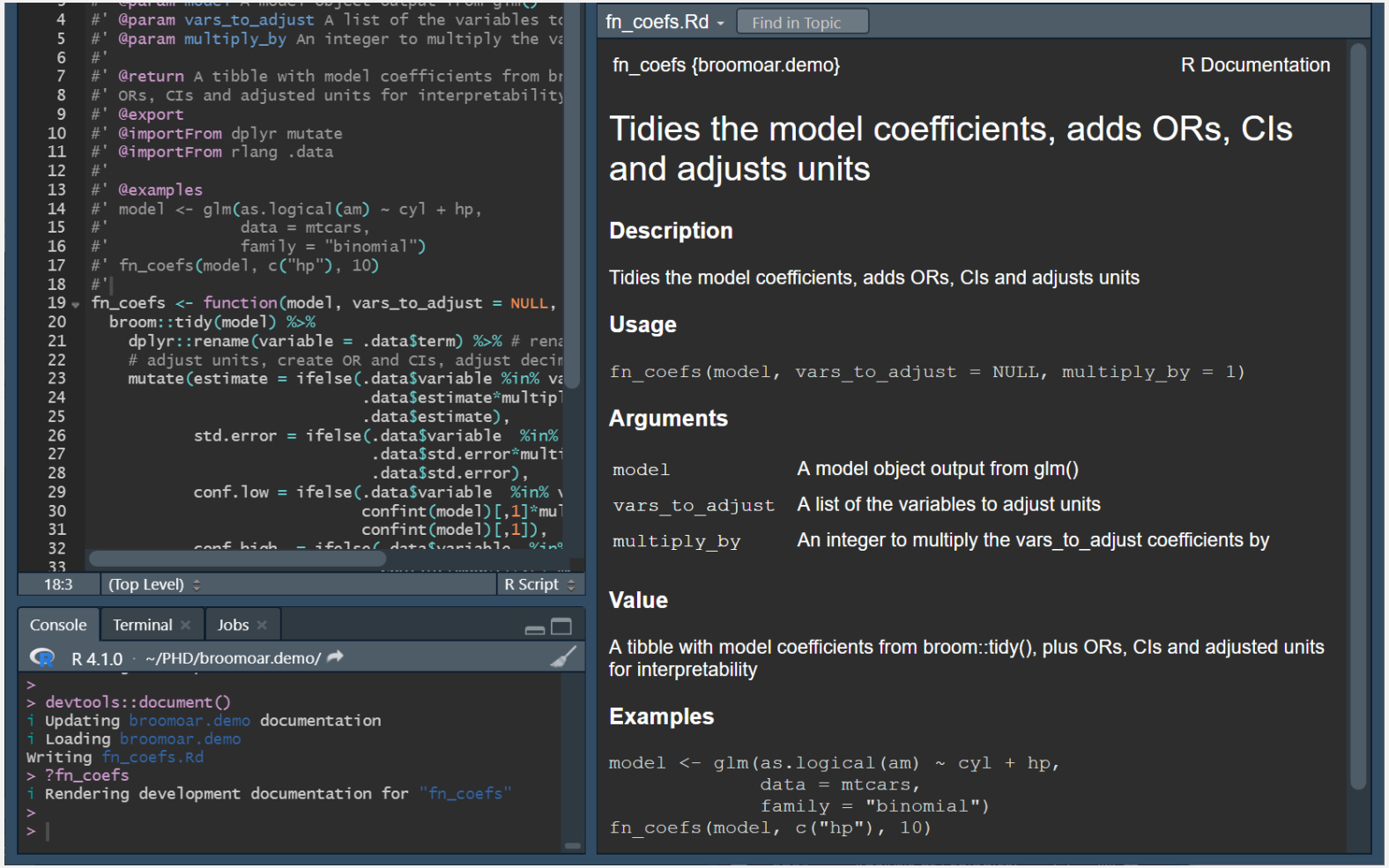


HOW - DEMONSTRATION

Document

function

```
devtools::document()
?fn_coefs
```



HOW - DEMONSTRATION

Run

check

```
devtools::check(cran = FALSE)
```

```
v  Running 'testthat.R' (908ms)

   See
     'C:/Users/SCDN42/AppData/Local/Temp/Rtmpu4DsqT/broomoar.demo.Rcheck/00check.log'
   for details.


-- R CMD check results ------------------------------------ broomoar.demo 0.0.0.9000 ----
Duration: 26.7s

> checking R code for possible problems ... NOTE
  fn_coefs: no visible global function definition for 'confint'
  Undefined global functions or variables:
    confint
  Consider adding
    importFrom("stats", "confint")
  to your NAMESPACE file.

0 errors v | 0 warnings v | 1 note x
>
```

ADD @importFrom stats confint TO THE ROXYGEN HEADER

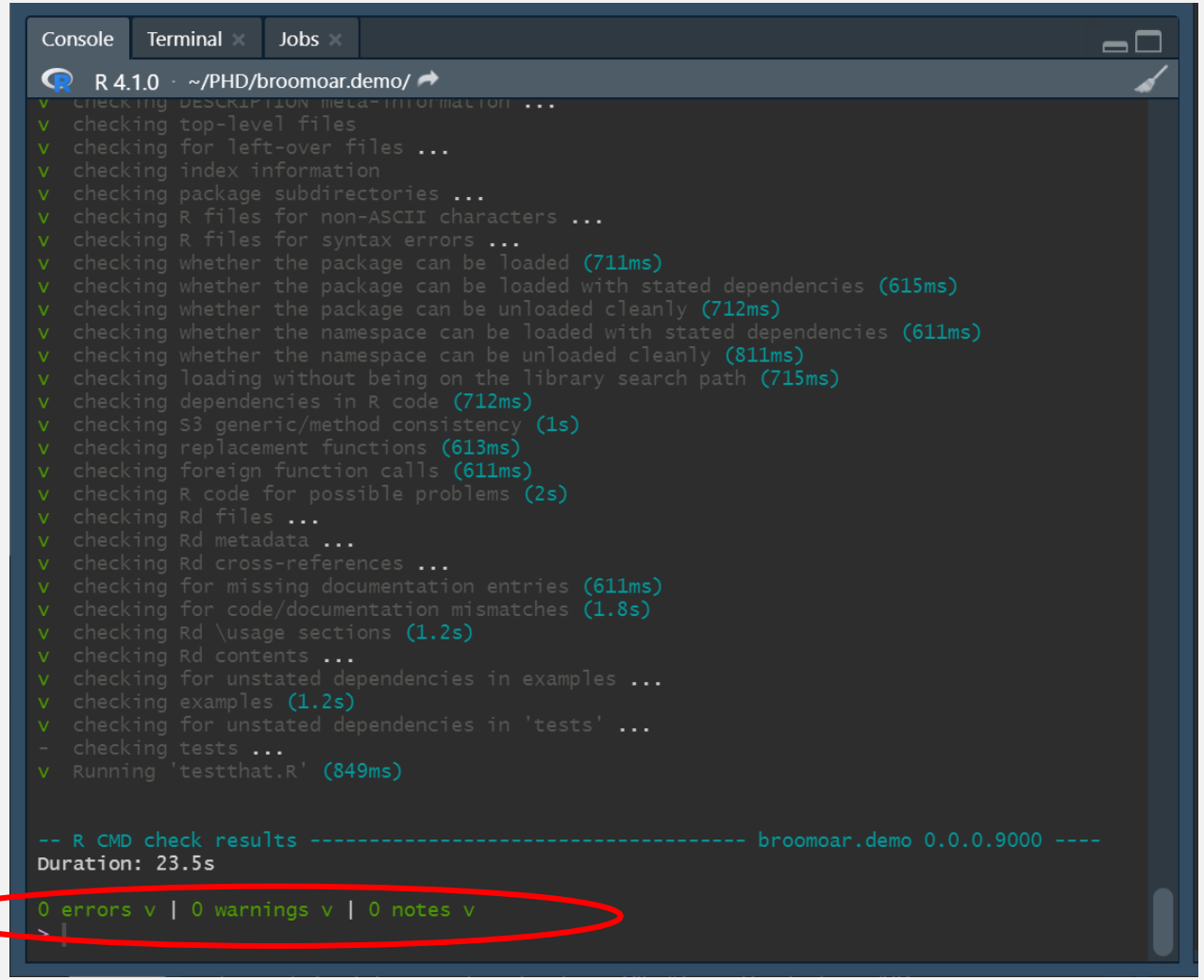
```
devtools::document()
```

OPEN THE NAMESPACE FILE AND CHECK IT'S UPDATED

**** NEVER EDIT THE NAMESPACE FILE ITSELF ****

HOW - DEMONSTRATION

```
devtools::check(cran = FALSE)
```



HOW - DEMONSTRATION

Design

and run

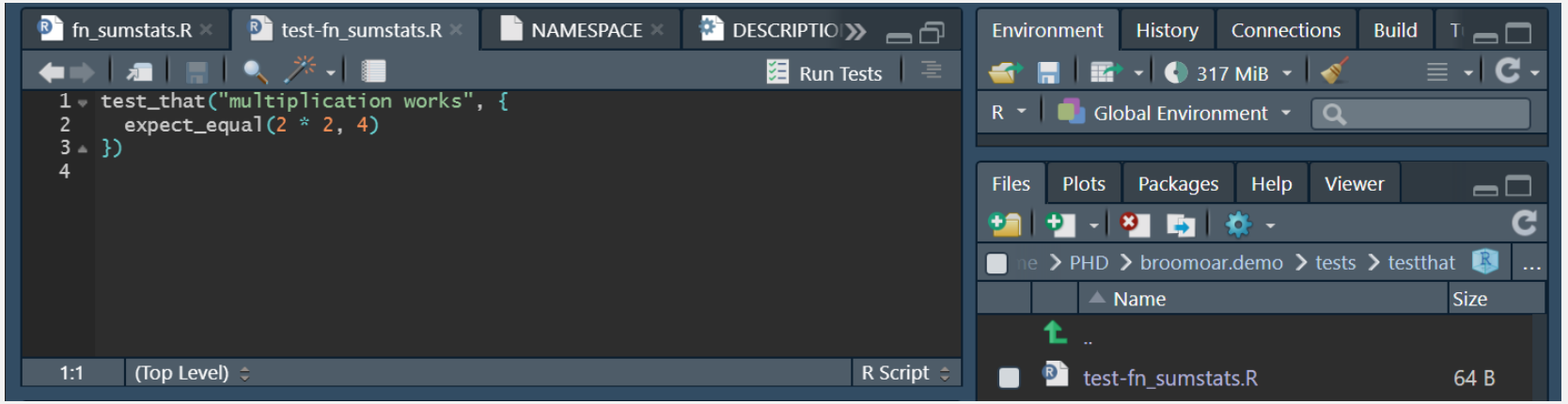
tests

```
usethis::use_testthat()
```

→ updates the DESCRIPTION file to include testthat as a suggested package and creates a test directory that contains:

- a script named testthat.R that contains all the code required to run the tests
- a directory testthat where you will save all of your test scripts.

```
usethis::use_test()
```



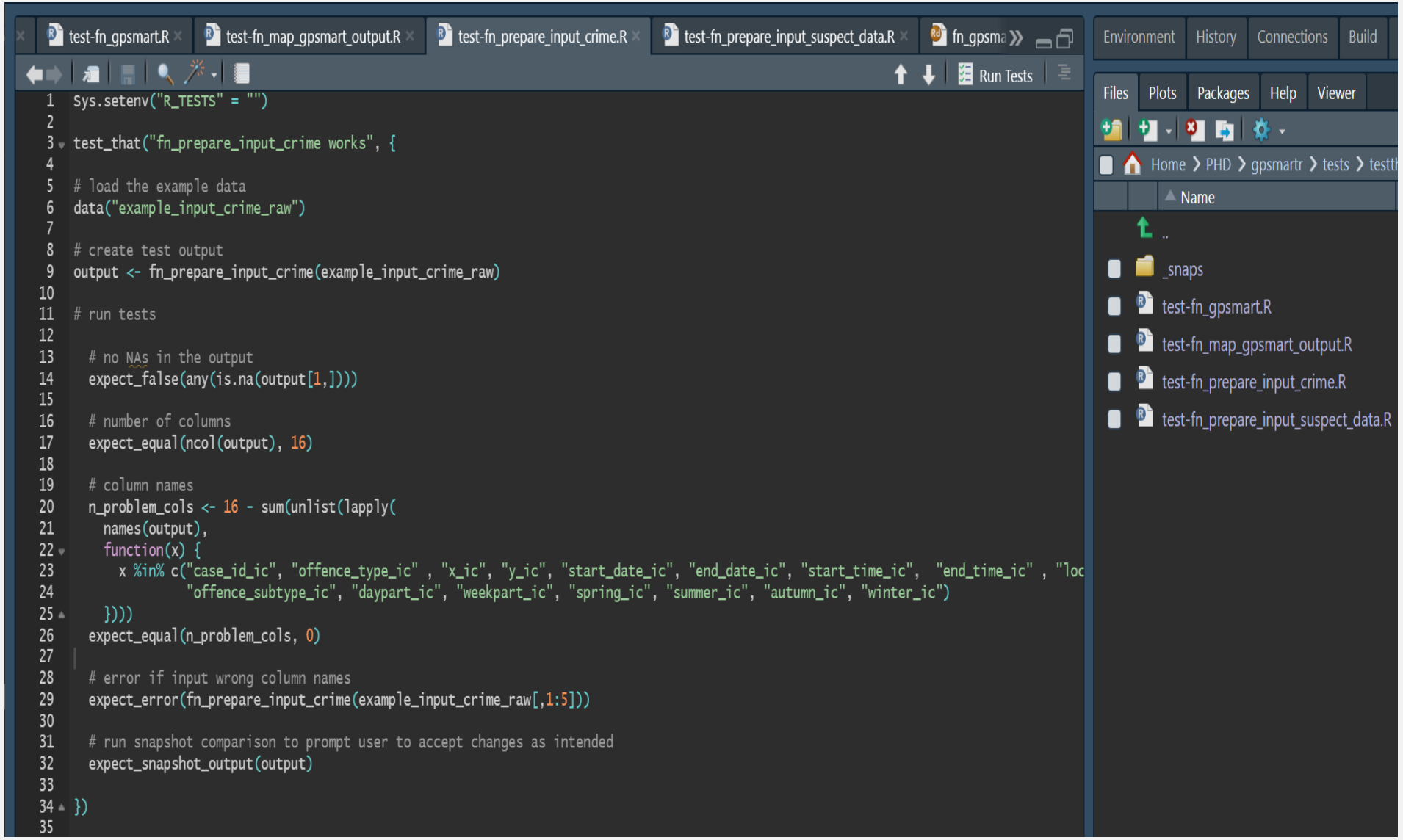

```
devtools::test()
```

```
i Loading broomoar.demo
i Testing broomoar.demo
v | F W S  OK | Context
v |         1 | fn_sumstats

== Results =====================================
[ FAIL 0 | WARN 0 | SKIP 0 | PASS 1 ]
```

HOW - DEMONSTRATION

Design

tests



HOW - DEMONSTRATION

Design

tests

psmart.R | DESCRIPTION | test-fn_gpsmart.R | test-fn_map_gpsmart_output.R | fn_gpsmart.Rd | fn_map_gpsmart_output.R | RE

Run Tests

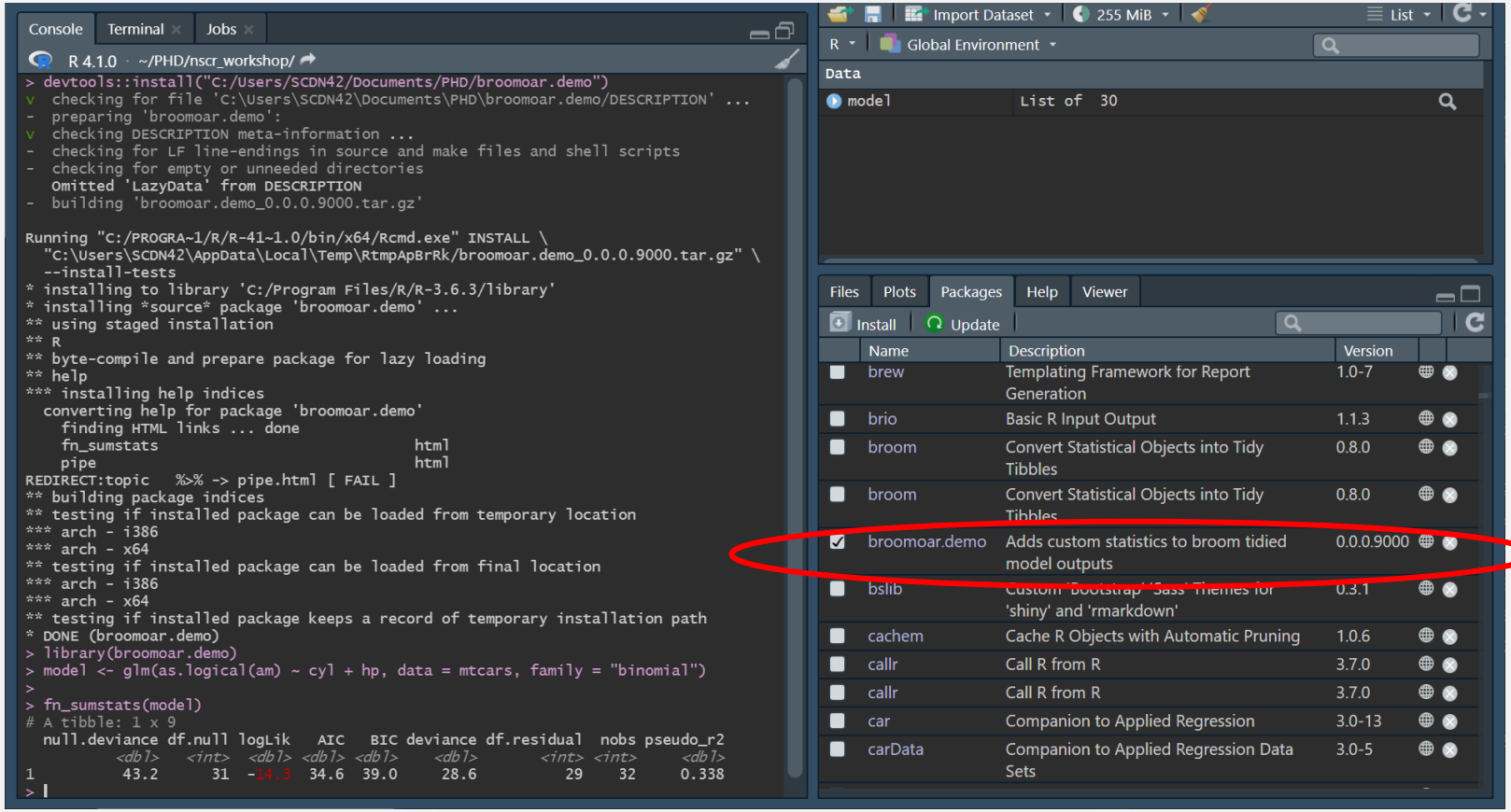
```r
Sys.setenv("R_TESTS" = "")

# load the example data
data("example_input_crime_raw")
data("example_input_suspects_raw")

# select a random sample to test ensuring representation of:
# - example input crime offender
# - records of each node_type for other suspects
# - but this time we don't need records of each level of each reliability and relevance factor, to check that the function copes with
set.seed(99)
input_suspects <-  example_input_suspects_raw %>%
  dplyr::group_by(node_type) %>%
  dplyr::slice_sample(n=100)  %>%
  dplyr::ungroup() %>%
  dplyr::bind_rows(
    example_input_suspects_raw %>%
      dplyr::filter(stringr::str_detect(person_id, stringr::str_sub(example_input_crime_raw$case_id[[1]], start = 1L, end = 7L)))
  ) %>% fn_prepare_suspect_data()

# create test outputs
output_default_weights_node_preds <- fn_gpsmart(input_crime = fn_prepare_input_crime(example_input_crime_raw),
                                                input_suspects = input_suspects,
                                                search_radius = 10,
                                                return_node_predictions = TRUE)

output_defaults <- fn_gpsmart(input_crime = fn_prepare_input_crime(example_input_crime_raw),
                              input_suspects = input_suspects)

# run tests
test_that("output no error", {

  expect_error(fn_map_gpsmart_output(output_default_weights_node_preds), NA)
})

test_that("output no warning", {

  expect_warning(fn_map_gpsmart_output(output_default_weights_node_preds), NA)

})

test_that("error if no node predictions in input", {

  expect_error(fn_map_gpsmart_output(output_defaults),
               "No suspect node predictions to map",
               fixed = TRUE)

})

# run snapshot comparison to prompt user to accept changes as intended
test_that("output matches expected output for test input using defaults", {

  expect_snapshot_output(fn_map_gpsmart_output(output_default_weights_node_preds))

})
```

HOW - DEMONSTRATION

Use

package

```
devtools::install("path/to/package/broomoar.demo")
library(broomoar.demo)
model <- glm(as.logical(am) ~ cyl + hp,
             data = mtcars, family = "binomial")
fn_sumstats(model)
```

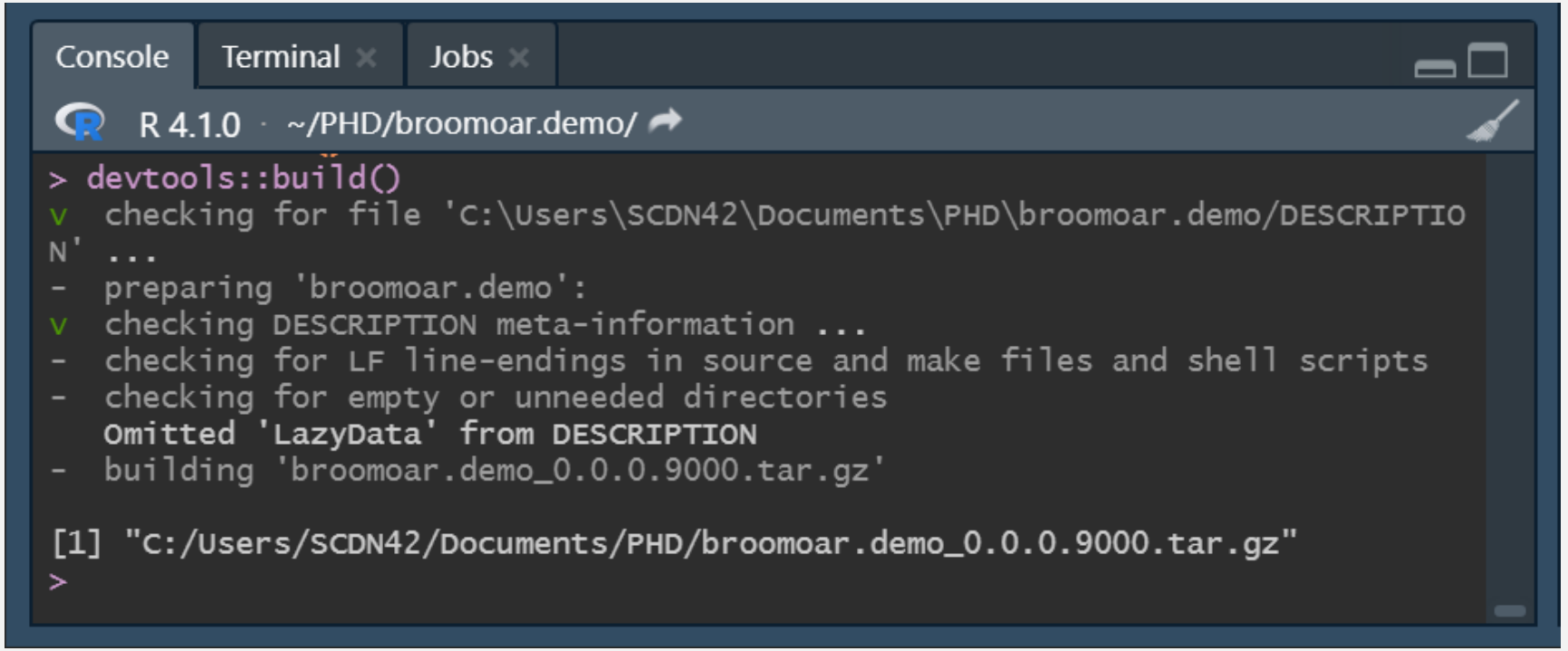


HOW - DEMONSTRATION

Share

package

```
devtools::build()
```




```
install.packages("path/to/file/broomoar.demo_0.0.0.9000.tar.gz")
```

OR PUSH PACKAGE TO GITHUB WITHIN RSTUDIO, THEN INSTALL FROM GITHUB:

```
devtools::install_github("author/packagename")
```