# script

### Turn on keyboard shortcut broadcasting

```
library(devtools)
create_package("~/Desktop/libminer")
```

```
# gert::git_config()
# gert::git_config_global()

use_git_config(
  user.name = "Andy Teucher",
  user.email = "andy.teucher@gmail.com"
)

use_git()
```

```
devtools::use_devtools()
```

```
use_r("lib-summary")
```

```
lib_summary <- function() {
  pkgs <- utils::installed.packages()
  pkg_tbl <- table(pkgs[, "LibPath"])
  pkg_df <- as.data.frame(pkg_tbl, stringsAsFactors = FALSE)
  names(pkg_df) <- c("Library", "n_packages")
  pkg_df
}
```

**Commit**

```
load_all()
```

```
check()
```

Open DESCRIPTION file

```
use_mit_license()
```

```
check()
```

**Commit**

DESCRIPTION file:

```
Package: libminer
Title: Explore Your R Libraries
Version: 0.0.0.9000
Authors@R:
    person("Andy", "Teucher", , "andy.teucher@gmail.com", role = c("aut", "cre"),
           comment = c(ORCID = "0000-0002-7840-692X"))
Description: Provides functions for learning about your R libraries, and the
    packages you have installed.
```

```
check()
```

**Commit**

```
git_sitrep()
```

```
use_github()
```

```
edit_r_profile()
```

*put this in the etherpad*

```
# Set usethis options:
options(
  usethis.description = list(
    "Authors@R" = utils::person(
      "Andy", "Teucher",
      email = "andy.teucher@gmail.com",
```

```
      role = c("aut", "cre"),
      comment = c(ORCID = "0000-1111-2222-3333")
    )
  )
)

options(
  warnPartialMatchArgs = TRUE,
  warnPartialMatchDollar = TRUE,
  warnPartialMatchAttr = TRUE
)
```

Ctrl + .

Ctrl+Alt+Shift+R

**show command palette here**

```
#' R Library Summary
#'
#' Provides a brief summary of the package libraries on your machine
#'
#' @return A data.frame containing the count of packages in each of the user's
#'   libraries
#' @export
#'
#' @examples
#' lib_summary()
```

```
document()

check()
```

**commit**

```
use_package_doc()

document()

check()
```

**commit**

```
use_testthat()
```

**commit**

```
use_test()
```

```
test_that("lib_summary returns expected results", {
  res <- lib_summary()
  expect_s3_class(res, "data.frame")
  expect_equal(ncol(res), 2)
  expect_equal(names(res), c("Library", "n_packages"))
  expect_type(res$Library, "character")
  expect_type(res$n_packages, "integer")
})

test_that("lib_summary fails appropriately", {
  expect_error(lib_summary("foo"), "unused argument")
})
```

```
test()

check()
```

**commit**

```
use_package("fs")
```

**commit**

```
lib_summary <- function(sizes = FALSE) {
  pkgs <- utils::installed.packages()
  pkg_tbl <- table(pkgs[, "LibPath"])
  pkg_df <- as.data.frame(pkg_tbl, stringsAsFactors = FALSE)
  names(pkg_df) <- c("Library", "n_packages")

  if (sizes) {
    pkg_df$lib_size <- vapply(
      pkg_df$Library,
      function(x) {
        sum(fs::file_size(fs::dir_ls(x, recurse = TRUE)))
      },
```

```
      FUN.VALUE = numeric(1)
    )
  }
  pkg_df
}
```

```
test() # failure for unused argument
```

```
test_that("lib_summary fails appropriately", {
  expect_error(lib_summary(sizes = "foo"), "not interpretable as logical")
})

test_that("sizes argument works", {
  res <- lib_summary(sizes = TRUE)
  expect_equal(names(res), c("Library", "n_packages", "lib_size"))
  expect_type(res$lib_size, "double")
})
```

**commit**

```
check() # will warn about undocumented parameter
```

Ctrl+Alt+Shift+R will insert the spot for the sizes param

```
#' Provides a brief summary of the package libraries on your machine
#'
#' @param sizes Should the sizes of the libraries be calculated?
#'   Logical; default `FALSE`.
#'
#' @return A data.frame containing the count of packages in each of the user's
#'   libraries. A `size` column is included if `sizes = TRUE`.
#' @export
#'
#' @examples
#' lib_summary()
#' lib_summary(sizes = TRUE)
```

```
document()

check()
```

**commit**

```
use_import_from("purrr", "map_dbl")
```

```
if (sizes) {
  pkg_df$lib_size <- map_dbl(
    pkg_df$Library,
    ~ sum(fs::file_size(fs::dir_ls(.x, recurse = TRUE))),
  )
}
```

```
test()

check()
```

**commit**

```
use_readme_rmd()
```

```
---
output: github_document
---

<!-- README.md is generated from README.Rmd. Please edit that file -->



# libminer

<!-- badges: start -->
<!-- badges: end -->

The goal of libminer is to provide an overview of your R library setup. It is a toy
package created as a part of a workshop and not meant for serious use.

## Installation

You can install the development version of libminer from [GitHub](https://github.com/) wit

```r
# install.packages("devtools")
```
```

```
devtools::install_github("ateucher/libminer")
```

## Example usage

To get a count of installed packages in each of your library locations,
optionally with the total sizes, use the `lib_summary()` function:

```{r example}
library(libminer)
lib_summary()
# specify `sizes = TRUE` to calculate the total size on disk of your packages
lib_summary(sizes = TRUE)
```
```

```
build_readme()
```

```
check()
```

```
install()
```

**commit**

```
use_github_action()
```

```
use_pkgdown_github_pages()
```

```
use_vignette("lib-sitrep", "Package Library Situation Report")
```

```
use_release_issue()
```

```
use_news_md()
```

```
use_cran_comments()
```
```