testit

build passing

This package provides two simple functions (30 lines of code in total):

- `assert(fact, ...)` : think of it as `message(fact) + stopifnot(...)`
- `test_pkg(package)` : runs tests with all objects (exported or non-exported) in the package namespace directly available, so no need to use `package:::name` for non-exported objects

Why?

The reason is laziness. It is tedious to type these commands repeatedly in tests:

```
message('checking if these numbers are equal...')
stopifnot(all.equal(1, 1+1e-10), 10*.1 == 1)

message('checking if a non-exported function works...')
stopifnot(is.character(package:::utility_foo(x = 'abcd', y = 1:100)))
```

With the two simple functions above, we type six letters (`assert`) instead of sixteen (`message + stopifnot`), and `assert` is also a more intuitive function name for testing purposes (you *assert* a fact followed by evidence):

```
assert(
  'these numbers are equal',
  all.equal(1, 1+1e-10), 10*.1 == 1
)

assert(
  'a non-exported function works',
  is.character(utility_foo(x = 'abcd', y = 1:100))
)
```

R CMD check

Put the tests under the directory `pkg_name/tests/testit/` (where `pkg_name` is the root directory of your package), and write a `test-all.R` under `pkg_name/tests/` :

```
library(testit)
```

```
test_pkg('pkg_name')
```

That is all for `R CMD check`. For package development, it is recommended to use **devtools**. In particular, `Ctrl + Shift + L` in RStudio makes all objects in a package visible to you, and you can play with the tests freely.

Installation

Stable version on CRAN:

```
install.packages('testit')
```

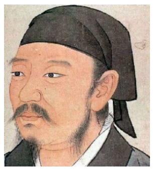
Development version:

```
devtools::install_github('yihui/testit')
```

More

How about **testthat**? Well, this package is far less sophisticated than **testthat**. There is nothing fancy in this package. Please do consider **testthat** if your tests require more granularity. I myself do not use **testthat** because I find it unnecessary to invent a new vocabulary (`testthat::expect_xxx`), and the error message of **testthat** is somehow obscure in my eyes. For **testit**, I do not need to think if I should use `expect_equal`, `expect_equivalent`, or `expect_identical`; I just write test conditions that return TRUE or FALSE. That is the only single rule to remember.

There is no plan to add new features or reinvent anything in this package. It is an intentionally tiny package.



Although he did not really mean it, Xunzi said something that happens to apply well to unit testing:

> □□□□□□□□□□□□□□□□□□□□□□

This package is free and open source software, licensed under GPL.