



Testing and Debugging

July 11, 2014

Devasena Inupakutika

The Software Sustainability Institute

University of Southampton

“I tried it, and it worked.”

But ...

Does it still work? Can you reproduce what you did? With what variety of inputs did you try?

“It’s not that we don’t test our code, it’s that we don’t store our tests so that they can be re-run automatically.”

- Hadley Wickham

Type of tests

- Check inputs
- Unit tests
- Integration tests
- Regression tests

Tests in R packages

- Examples in .Rd files
- Vignettes
- tests/ directory

R CMD Check

testthat package

- **Expectations**

`expect_equal(10, 10 + 1e-7)`

`expect_identical(10, 10)`

`expect_equivalent(c("one"=1), 1)`

`expect_warning(log(-1))`

`expect_error(1 + "a")`

- **Tests**

- **Contexts**

- **Store tests in inst/tests**

- **tests/run-all.R file containing**

```
library(testthat) library(mypkg) test_package("mypkg")
```

testthat test structure

Hierarchical structure made up of:

- Expectations
- Test (groups multiple expectations)
- Context (groups multiple tests)

Expectations

- Finest level of testing
- Makes binary assertion about whether or not a value is as you expect
- If expectation isn't true, `testthat` will raise an error
- Easy to read: Syntax close to English language

Built in expectations

Full

`expect_that(x, is_true())`
`expect_that(x, is_false())`
`expect_that(x, is_a(y))`
`expect_that(x, equals(y))`
`expect_that(x, is_equivalent_to(y))`
`expect_that(x, is_identical_to(y))`
`expect_that(x, matches(y))`
`expect_that(x, prints_text(y))`
`expect_that(x, shows_message(y))`
`expect_that(x, gives_warning(y))`
`expect_that(x, throws_error(y))`

Abbreviation

`expect_true(x)`
`expect_false(x)`
`expect_is(x, y)`
`expect_equal(x, y)`
`expect_equivalent(x, y)`
`expect_identical(x, y)`
`expect_match(x, y)`
`expect_output(x, y)`
`expect_message(x, y)`
`expect_warning(x, y)`
`expect_error(x, y)`

Example `testthat` test

```
context("String length")

test_that("str_length is number of characters", {
  expect_that(str_length("a"), equals(1))
  expect_that(str_length("ab"), equals(2))
  expect_that(str_length("abc"), equals(3))
})

test_that("str_length of missing is missing", {
  expect_that(str_length("NA"), equals(2))
})

test_that("str_length of factor is length of level", {
  expect_that(str_length(factor("a")), equals(1))
  expect_that(str_length(factor("ab")), equals(2))
  expect_that(str_length(factor("abc")), equals(3))
})
```

Running tests

Run using one of the following:

- `test_file("filename.R")`
- `test_dir("path to test folder")`
- `source("filename.R")`

Load and test package

```
library(testthat)
```

```
library(package_name)
```

```
test_package("package_name")
```

Any Questions?