# Intro to ‘testthat’ package

Niels Aka

27 September 2016

# Why (unit) test?

Wikipedia: “unit testing is a software testing method by which individual units of source code [. . . ] are tested to determine whether they are fit for use”
⇒ lowest level of testing

- ensures that single functions work as expected
- catch bugs early on
- save time later on

# Why test as a scientist?

- a plus/minus sign error can invalidate research results...
- more integrity, reliability, trust

# Why test in R?

- starts to make sense when you
  - write own functions
  - customise other functions
- R language is flexible (implicit coercion, recycling)
  $\Rightarrow$ make sure inputs + output are as expected

# Unit testing in R: the `testthat` package

- ideal for packages, but works also with scripts
- tests are organised in three levels
  1. context → general functionality
  2. test suites → a specific aspect
  3. expectations → a single unit of functionality

# Simple example

```
context("String length")

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

# Workflow

When using testthat, there are usually three steps

1. write code
2. write tests
3. source testthat.R which runs all tests in separate testing environments and reports summary of successes and failures

# Example file: transform.R

```r
# transform time-series in levels into growth rates
trans_gr <- function(x) {
  diff(log(x))
}
```

# Example file: test-transform.R

```r
context("Testing transformation routines")

test_that("transforming to growth rates works", {
  n    <- 5
  rate <- 0.02
  x <- (1 + rate) ^ (1:n)

  # test class
  expect_is(object = trans_gr(x),
            class = "numeric")
  # test numerical results
  ref <- rep(rate, n - 1)
  expect_equal(object = trans_gr(x), expected = ref,
               tol = 1e-3)
  expect_identical(object = trans_gr(x), expected = ref)
})
```

# Example file: testthat.R

```r
source("transform.R")
testthat::test_file("test-transform.R")
```

```
## Testing transformation routines: ..1
##
## Failed -----------------------------------------------
## 1. Failure: transforming to growth rates works (@test-tr
## trans_gr(x) not identical to `ref`.
## 4/4 mismatches (average diff: 0.000197)
## [1] 0.0198 - 0.02 == -0.000197
## [2] 0.0198 - 0.02 == -0.000197
## [3] 0.0198 - 0.02 == -0.000197
## [4] 0.0198 - 0.02 == -0.000197
##
##
## DONE ===================================================
```

# Resources

- R Journal entry
- CRAN website
- Hadley's website