Package: testmath (via r-universe)

November 8, 2026

Title Unit Tests for Mathematical Properties

Version 0.1.0

Description Inspired by 'testthat', 'testmath' aims to provide some functions for unit tests of mathematical properties in a readable manner. At the moment, these functions cover a few aspects of two distinct areas - unit tests for matrices and functions. Failure should provide detailed information for what condition led to the test failing.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 8.0.0

Config/testthat/edition 3

Imports rlang, testthat, waldo

Config/pak/sysreqs cmake make libuv1-dev

Repository https://stefanobarbieri4.r-universe.dev

Date/Publication 2026-10-14 18:01:03 UTC

RemoteUrl https://github.com/stefanobarbieri4/testmath

RemoteRef HEAD

RemoteSha 2dcb481adcd3084244042b42595b02f11bfe4e62

Contents

expect_diagonal 2
expect_idempotent 2
expect_involutory 3
expect_n_idempotent 4
expect_n_involutory 5
expect_orthogonal 6
expect_periodic 6
expect_psd 7
expect_spd 8
expect_symmetric 9

Index **10**

`expect_diagonal` *Do you expect a matrix to be diagonal?*

Description

Do you expect a matrix to be diagonal?

Usage

```
expect_diagonal(matrix, tolerance = testthat::testthat_tolerance(), ...)
```

Arguments

`matrix`	matrix. The matrix you are checking to be diagonal.
`tolerance`	numeric. The tolerance of difference between compared entries. Defaults to `testthat`'s default tolerance.
`...`	Additional arguments passed to `waldo::compare()`.

See Also

Other matrix properties: `expect_orthogonal()`, `expect_psd()`, `expect_spd()`, `expect_symmetric()`

Examples

```
mat_diag <- diag(c(1, 5, 3))
mat_not_diag <- matrix(c(1, 2, 0, 3), nrow = 2)

expect_diagonal(mat_diag)
testthat::expect_failure(expect_diagonal(mat_not_diag))
```

`expect_idempotent` *Do you expect a function to be idempotent?*

Description

This function tests whether applying a function twice yields the same result as applying it once.

Usage

```
expect_idempotent(f, x, tolerance = testthat::testthat_tolerance(), ...)
```

Arguments

f	function. The function you are testing.
x	parameter(s). The value under which you test the function.
tolerance	Passed on to waldo::compare(). If non-null, ignores small floating point differences
...	Additional arguments passed into waldo::compare()

See Also

Other function properties: expect_involutory(), expect_n_idempotent(), expect_n_involutory(), expect_periodic()

Examples

```
x <- c(2, 1, 3)
expect_idempotent(mean, x)
```

---

expect_involutory *Do you expect a function to be involutory?*

Description

This function tests whether a function, f is its own inverse for a given value of x.

Usage

```
expect_involutory(f, x, tolerance = testthat::testthat_tolerance(), ...)
```

Arguments

f	function. The function you are testing to be involutory
x	parameter(s). The value under which you test the function.
tolerance	Passed on to waldo::compare(). If non-null, ignores small floating point differences
...	Additional arguments passed into waldo::compare()

Details

Tests that $f(f(x)) = x$. A function is called involutory (or a self-inverse function) if this applies at each element in its domain.

See Also

Other function properties: expect_idempotent(), expect_n_idempotent(), expect_n_involutory(), expect_periodic()

Examples

```
x <- c(2, 1, 3)
expect_involutory(rev, x) # passes
testthat::expect_failure(expect_involutory(sort, x)) # fails
```

`expect_n_idempotent` *Do you expect a function to be n-idempotent?*

Description

This function tests whether a function, f, applied n times on x yields the same result as applying it a single time.

Usage

```
expect_n_idempotent(f, x, n, tolerance = testthat::testthat_tolerance(), ...)
```

Arguments

`f`	function. The function you are testing.
`x`	parameter(s). The value under which you test the function `f`.
`n`	integer. The expected order of idempotence.
`tolerance`	Passed on to `waldo::compare()`. If non-null, ignores small floating point differences
`...`	Additional arguments passed into `waldo::compare()`

Details

Tests that $f^n(x) = f(x)$. For `n = 2`, this is equivalent to `expect_idempotent`.

See Also

Other function properties: `expect_idempotent()`, `expect_involutory()`, `expect_n_involutory()`, `expect_periodic()`

Examples

```
# A function that reaches its fixed/stable state after 3 steps
custom_func <- function(x) {
  pmin(pmax(x, 0), 1) # e.g., clipping logic
}

expect_n_idempotent(custom_func, -5:5, n = 3)
```

`expect_n_involutory` *Do you expect a function to be n-involutory?*

Description

This function tests whether a function, f, applied n times on x returns x.

Usage

```
expect_n_involutory(f, x, n, tolerance = testthat::testthat_tolerance(), ...)
```

Arguments

`f`	function. The function you are testing.
`x`	parameter(s). The value under which you test the function `f`.
`n`	integer. The expected order of involution.
`tolerance`	Passed on to `waldo::compare()`. If non-null, ignores small floating point differences
`...`	Additional arguments passed into `waldo::compare()`

Details

Tests that $f^n(x) = x$. For `n = 2`, this is equivalent to `expect_involutory`.

See Also

Other function properties: `expect_idempotent()`, `expect_involutory()`, `expect_n_idempotent()`, `expect_periodic()`

Examples

```
mult_by_im <- function(x) {
  x * 1i
}

expect_n_involutory(mult_by_im, 1, n = 4)
```

expect_orthogonal *Do you expect a matrix to be orthogonal?*

Description

Do you expect a matrix to be orthogonal?

Usage

```
expect_orthogonal(matrix, tolerance = testthat::testthat_tolerance(), ...)
```

Arguments

matrix	matrix. The matrix you are checking for orthogonality.
tolerance	numeric. The tolerance of difference between compared entries. Defaults to testthat's default tolerance.
...	Additional arguments passed to waldo::compare().

See Also

Other matrix properties: expect_diagonal(), expect_psd(), expect_spd(), expect_symmetric()

Examples

```
# Rotation matrix is orthogonal
theta <- pi / 4
R <- matrix(c(cos(theta), -sin(theta), sin(theta), cos(theta)), nrow = 2)
expect_orthogonal(R)

# Non-orthogonal matrix
mat_bad <- matrix(1:4, nrow = 2)
testthat::expect_failure(expect_orthogonal(mat_bad))
```

expect_periodic *Do you expect a function to be periodic?*

Description

This function tests whether a function, f, is periodic on x with period T

Usage

```
expect_periodic(f, x, period, tolerance = testthat::testthat_tolerance(), ...)
```

Arguments

f	function. The function you are testing to be periodic.
x	parameter(s). The value under which you test the function f.
period	numeric. The expected period of $f(x)$
tolerance	Passed on to waldo::compare(). If non-null, ignores small floating point differences
...	Additional arguments passed into waldo::compare()

Details

Tests that $f(x + T) = f(x)$

See Also

Other function properties: expect_idempotent(), expect_involutory(), expect_n_idempotent(), expect_n_involutory()

Examples

```
expect_periodic(cos, c(0, 1, 2), 2*pi)
expect_periodic(cos, c(0, 1, 2), 4*pi)
expect_periodic(cos, c(pi/2), pi)
```

---

expect_psd *Do you expect a matrix to be positive semidefinite?*

Description

Do you expect a matrix to be positive semidefinite?

Usage

```
expect_psd(matrix, tolerance = testthat::testthat_tolerance())
```

Arguments

matrix	matrix. The matrix you are checking to be PSD.
tolerance	numeric. The tolerance for symmetry checks and eigenvalues. Defaults to testthat's default tolerance.

Details

In addition to checking whether the matrix is symmetric, it also checks if the eigenvalues of matrix are all positive. Equivalent to checking if $x^T Ax \geq 0$.

See Also

Other matrix properties: expect_diagonal(), expect_orthogonal(), expect_spd(), expect_symmetric()

Examples

```
# Matrix with zero eigenvalue (PSD, but not strictly PD)
mat_psd <- matrix(c(1, 1, 1, 1), nrow = 2)
expect_psd(mat_psd)

# Indefinite matrix fails PSD
mat_bad <- matrix(c(1, 2, 2, 1), nrow = 2)
testthat::expect_failure(expect_psd(mat_bad))
```

---

expect_spd *Do you expect a matrix to be symmetric positive definite?*

Description

Do you expect a matrix to be symmetric positive definite?

Usage

```
expect_spd(matrix, tolerance = testthat::testthat_tolerance())
```

Arguments

matrix	matrix. The matrix you are checking to be SPD.
tolerance	numeric. The tolerance for symmetry checks and eigenvalues. Defaults to testthat's default tolerance.

Details

In addition to checking whether the matrix is symmetric, it also checks if the eigenvalues of matrix are all strictly positive. Equivalent to checking if $x^T Ax > 0$.

See Also

Other matrix properties: expect_diagonal(), expect_orthogonal(), expect_psd(), expect_symmetric()

Examples

```
mat_spd <- matrix(c(2, -1, -1, 2), nrow = 2) # spd
mat_not_spd <- matrix(c(1, 2, 2, 1), nrow = 2) # symmetric, but not positive definite

expect_spd(mat_spd)
testthat::expect_failure(expect_spd(mat_not_spd))
```

`expect_symmetric` *Do you expect a matrix to be symmetric?*

Description

Do you expect a matrix to be symmetric?

Usage

```
expect_symmetric(matrix, tolerance = testthat::testthat_tolerance())
```

Arguments

`matrix`	matrix. The matrix you are checking for symmetry.
`tolerance`	numeric. The tolerance of difference between compared entries. Defaults to `testthat`'s default tolerance

Details

Checks whether $A_{ij} = A_{ji}$ for all i, j. As soon as an infringement is found, it breaks early.

See Also

Other matrix properties: `expect_diagonal()`, `expect_orthogonal()`, `expect_psd()`, `expect_spd()`

Examples

```
mat1 <- matrix(c(7, 4, 4, 5), nrow = 2, ncol = 2) # symmetric
mat2 <- matrix(c(4, 1, 3, 9), nrow = 2, ncol = 2) # not symmetric

expect_symmetric(mat1)
testthat::expect_failure(expect_symmetric(mat2))
```

Index

∗ **function properties**
    expect_idempotent, 2
    expect_involutory, 3
    expect_n_idempotent, 4
    expect_n_involutory, 5
    expect_periodic, 6
∗ **matrix properties**
    expect_diagonal, 2
    expect_orthogonal, 6
    expect_psd, 7
    expect_spd, 8
    expect_symmetric, 9

expect_diagonal, 2
expect_diagonal(), *6*, *8, 9*
expect_idempotent, 2
expect_idempotent(), *3–5*, *7*
expect_involutory, 3
expect_involutory(), *3–5*, *7*
expect_n_idempotent, 4
expect_n_idempotent(), *3*, *5*, *7*
expect_n_involutory, 5
expect_n_involutory(), *3, 4*, *7*
expect_orthogonal, 6
expect_orthogonal(), *2*, *8, 9*
expect_periodic, 6
expect_periodic(), *3–5*
expect_psd, 7
expect_psd(), *2*, *6*, *8, 9*
expect_spd, 8
expect_spd(), *2*, *6*, *8, 9*
expect_symmetric, 9
expect_symmetric(), *2*, *6*, *8*

waldo::compare(), *3–5*, *7*