Package 'normalize'

November 12, 2026

Type Package

Title Centering and Scaling of Numeric Data

Version 0.2.0

Description Centers and scales numeric data in vectors, matrices, data frames, time series, and lists of these objects. Rows or columns can be left unchanged or be normalized jointly. The values used for centering and scaling are returned as attributes.

License GPL (>= 3)

Encoding UTF-8

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Imports stats

URL https://github.com/loelschlaeger/normalize

BugReports https://github.com/loelschlaeger/normalize/issues

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Lennart Oelschläger [aut, cre]

Maintainer Lennart Oelschläger <oelschlaeger.lennart@gmail.com>

Repository CRAN

Date/Publication 2026-11-12 14:50:02 UTC

Contents

normalize 2

Index **5**

normalize *Normalize numeric data*

Description

Center and scale vectors, matrices, data frames, time series, or lists of these objects, i.e., subtract the mean and divide by the standard deviation.

Usage

```
normalize(x, center = TRUE, scale = TRUE, ...)

## Default S3 method:
normalize(x, ...)

## S3 method for class 'ts'
normalize(x, center = TRUE, scale = TRUE, ...)

## S3 method for class 'numeric'
normalize(x, center = TRUE, scale = TRUE, ...)

## S3 method for class 'logical'
normalize(x, center = TRUE, scale = TRUE, ...)

## S3 method for class 'matrix'
normalize(
  x,
  center = TRUE,
  scale = TRUE,
  byrow = FALSE,
  ignore = integer(),
  jointly = list(),
  ...
)

## S3 method for class 'data.frame'
normalize(
  x,
  center = TRUE,
  scale = TRUE,
  byrow = FALSE,
  ignore = integer(),
  jointly = list(),
  ...
)

## S3 method for class 'list'
```

```
normalize(x, center = TRUE, scale = TRUE, ...)
```

Arguments

x	An object to be normalized.
center	[logical(1)] Should values be centered to zero mean?
scale	[logical(1)] Should values be scaled to unit variance?
...	Further arguments to be passed to or from other methods.
byrow	[logical(1)] Only relevant if x has two dimensions (rows and columns). In this case, set to TRUE to normalize row-wise or FALSE to normalize column-wise (default).
ignore	[integer()] Only relevant if x has two dimensions (rows and columns). Indices to leave unchanged. With byrow = TRUE, these are row indices. With byrow = FALSE, these are column indices. This is useful, for example, to keep identifier or non-numeric columns in a data frame.
jointly	[list()] Only relevant if x has two dimensions (rows and columns). A list of disjoint index groups to normalize jointly, see details. With byrow = TRUE, these are row indices. With byrow = FALSE, these are column indices.

Details

A vector is normalized as a whole, while a matrix or data frame is normalized column-wise (or row-wise if byrow = TRUE). A univariate time series is treated like a vector and a multivariate time series like a matrix. For a list, each element is normalized separately.

Note that logical values are treated as 0 and 1, and that normalized values are of type double.

Value

A normalized version of x, preserving its class, dimensions, names, and other attributes. Lists return a list with each element normalized.

The centering and scaling values used are stored in the "center" and "scale" attributes (only if the corresponding operation was requested). For a vector, each attribute is a single number, while for a matrix or data frame, each attribute has one element per column (or per row if byrow = TRUE). The elements are named if the columns or rows have names, and they are NA for ignored columns or rows. For a list, the elements of the result have these attributes.

Missing values

Missing values are ignored when computing the mean and the standard deviation, and they stay missing in the normalized object. The standard deviation is missing if fewer than two values are available, and it is zero if all values are equal. In these cases, scaling yields values that are not finite. A warning is issued if the normalized values contain missing or infinite values.

Joint normalization

Rows or columns that are normalized jointly (see the `jointly` argument) share one centering and one scaling value. More precisely, the centering value is the mean of all values in the group, and the scaling value is the pooled standard deviation

$$\sqrt{\sum_k w_k s_k^2 / \sum_k w_k},$$

where s_k^2 denotes the variance of row or column k in the group and the weight w_k is its number of non-missing values minus one. Without missing values, this is the square root of the mean of the variances. Note that the pooled standard deviation measures the variation within the rows or columns, such that differences between their means do not enter.

Examples

```
# can normalize vectors, matrices, data.frames, and lists of such types
normalize(
  list(
    c(-3, 0, 3),
    matrix(1:12, nrow = 3, ncol = 4),
    data.frame(a = 1:3, b = 4:6, c = 7:9, d = 10:12)
  )
)

# centering and scaling values are stored as attributes
x <- normalize(c(2, 4, 9))
attr(x, "center")
attr(x, "scale")

# can ignore columns (or rows)
normalize(
  data.frame(a = 1:3, b = c("A", "B", "C"), c = 7:9, d = 10:12),
  ignore = 2
)

# can normalize columns (or rows) jointly
normalize(
  matrix(1:12, nrow = 3, ncol = 4),
  jointly = list(1:2, 3:4)
)

# can normalize time series
normalize(stats::ts(c(1, 3, 5, 7), start = c(2000, 1), frequency = 4))
```

Index

normalize, 2