Package 'prepkit'

January 23, 2026

Type Package

Title Data Normalization and Transformation

Version 0.1.1

Description Provides functions for data normalization and transformation in preprocessing stages. Implements scaling methods (min-max, Z-score, L2 normalization) and power transformations (Box-Cox, Yeo-Johnson). Box-Cox transformation is described in Box and Cox (1964) <doi:10.1111/j.2517-6161.1964.tb00553.x>, Yeo-Johnson transformation in Yeo and Johnson (2000) <doi:10.1093/biomet/87.4.954>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

URL https://gonrui.github.io/prepkit/, https://github.com/Gonrui/prepkit/

BugReports https://github.com/Gonrui/prepkit/issues/

Language en-US

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Imports ggplot2

Depends R (>= 3.5)

NeedsCompilation no

Author Rui Gong [aut, cre] (ORCID: <https://orcid.org/0000-0001-5112-5696>)

Maintainer Rui Gong <gongrui4432@gmail.com>

Repository CRAN

Date/Publication 2026-01-23 17:00:02 UTC

Contents

`norm_decimal` *Decimal Scaling Normalization*

Description

Normalizes a numeric vector by moving the decimal point of values of attribute A. The number of decimal points moved depends on the maximum absolute value of A.

Usage

```
norm_decimal(x, na.rm = TRUE)
```

Arguments

`x`	A numeric vector.
`na.rm`	Logical. Should NA values be ignored when determining the scaling factor? Default is `TRUE`.

Details

Formula: $x' = \frac{x}{10^j}$ where j is the smallest integer such that $\max(|x'|) < 1$.

Value

A numeric vector with values typically in the range (-1, 1).

References

Han, J., Kamber, M., & Pei, J. (2011). *Data mining: concepts and techniques* (3rd ed.). Morgan Kaufmann.

Examples

```
# Max value is 980, so j=3 (divides by 1000) -> 0.98
norm_decimal(c(10, 500, 980))

# Works with negative numbers
norm_decimal(c(-50, 50, 200))
```

`norm_l2` *L2 Normalization (Unit Vector)*

Description

Scales the vector so that its Euclidean norm (L2 norm) is 1. This technique is often used in text mining and high-dimensional clustering, and is related to spatial sign prepkitocessing in robust statistics.

Usage

```
norm_l2(x, na.rm = TRUE)
```

Arguments

`x`	A numeric vector.
`na.rm`	Logical. Remove NAs for norm calculation? Default is `TRUE`.

Details

Formula: $x' = \frac{x}{\sqrt{\sum x^2}}$

Value

A numeric vector with an L2 norm of 1.

References

Serneels, S., De Nages, E., & Van Espen, P. J. (2006). Spatial sign prepkitocessing: a simple way to impart moderate robustness to multivariate estimators. *Journal of Chemical Information and Modeling*, 46(3), 1402-1409. doi:10.1021/ci050498u

Han, J., Kamber, M., & Pei, J. (2011). *Data mining: concepts and techniques* (3rd ed.). Morgan Kaufmann.

Examples

```
# Convert a vector to unit length
x <- c(3, 4)
norm_l2(x) # Returns c(0.6, 0.8)
```

`norm_mean` *Mean Normalization*

Description

Scales a numeric vector by centering it around its mean and scaling it by its range. The resulting vector has a mean of 0 and values typically within [-1, 1].

Usage

```
norm_mean(x, na.rm = TRUE)
```

Arguments

`x`	A numeric vector.
`na.rm`	Logical. Should NA values be removed during calculation? Default is `TRUE`.

Details

Formula: $x' = \frac{x - \text{mean}(x)}{\max(x) - \min(x)}$

Value

A numeric vector. If the range is 0 (all values are identical), returns a centered vector (zeros).

References

Han, J., Kamber, M., & Pei, J. (2011). *Data mining: concepts and techniques* (3rd ed.). Morgan Kaufmann.

Examples

```
# Result ranges from approx -0.5 to 0.5, mean is 0
norm_mean(c(1, 2, 3, 4, 5))

# Handles negative values
norm_mean(c(-10, 0, 10))
```

`norm_minmax` *Min-Max Normalization*

Description

Scales a numeric vector to a specific range, typically [0, 1]. This method is sensitive to outliers.

Usage

```
norm_minmax(x, min_val = 0, max_val = 1, na.rm = TRUE)
```

Arguments

`x`	A numeric vector.
`min_val`	The minimum value of the target range. Default is 0.
`max_val`	The maximum value of the target range. Default is 1.
`na.rm`	Logical. Should NA values be removed during min/max calculation? Default is `TRUE`.

Details

Formula: $x' = \frac{x - \min(x)}{\max(x) - \min(x)} \times (\text{max\_val} - \text{min\_val}) + \text{min\_val}$

Value

A numeric vector scaled to the range [min_val, max_val].

References

Han, J., Kamber, M., & Pei, J. (2011). *Data mining: concepts and techniques* (3rd ed.). Morgan Kaufmann.

Examples

```
norm_minmax(c(1, 2, 3, 4, 5))
norm_minmax(c(1, 2, 3), min_val = -1, max_val = 1)
```

`norm_mode_range` *M-Score (Mode-Range Normalization)*

Description

Unlike Z-Score or Min-Max, the M-Score algorithm identifies the "Mode Range" (the most frequent value range) and maps it to 0. This effectively suppresses the noise of daily routine (e.g., stable step counts) and amplifies anomalies (e.g., frailty or sudden activity).

It maps:

- **Mode Range**: $[k_L, k_R] \to 0$ (Baseline/Routine)
- **Left Tail**: $[min, k_L) \to [-1, 0)$ (Decline/Frailty)
- **Right Tail**: $(k_R, max] \to (0, 1]$ (Surge/Hyperactivity)

Usage

```
norm_mode_range(x, tau = 0.8, digits = 0)
```

Arguments

`x`	A numeric vector.
`tau`	A numeric value (0 to 1). The threshold ratio for defining the mode plateau. Bins with `freq >= tau * max_freq` are considered part of the routine. Default is 0.8.
`digits`	Integer or NULL. If not NULL, values are rounded to this many decimal places **solely for identifying the mode**. This makes the algorithm robust against sensor noise (e.g., 1.0001 vs 1.0002). Default is 0 (rounds to integer), which is ideal for step counts or heart rates. Set to `NULL` to disable rounding.

Details

A robust normalization method designed for longitudinal behavioral data with a "routine plateau". Also known as Mode-Range Normalization (MRN).

Value

A numeric vector in the range [-1, 1].

References

Gong, R. (2026). M-Score: A Robust Normalization Method for Detecting Anomalies in Longitudinal Behavioral Data. *arXiv prepkitint*. (Submitted)

Examples

```
# Scenario 1: Integer data (Standard)
steps <- c(3000, 3000, 200, 5000)
norm_mode_range(steps)

# Scenario 2: Noisy Sensor Data (Floating point)
# Without 'digits', these would be seen as different values.
# With digits=1, they are grouped into the same mode.
sensor_data <- c(9.81, 9.82, 9.80, 2.5, 15.0)
norm_mode_range(sensor_data, digits = 1)
```

---

norm_robust *Robust Standardization (Median-MAD)*

Description

Standardizes a numeric vector using robust statistics: median and median absolute deviation (MAD). This method is less sensitive to outliers compared to Z-score standardization.

Usage

```
norm_robust(x, na.rm = TRUE, constant = 1.4826)
```

Arguments

x	A numeric vector.
na.rm	Logical. Should NA values be removed? Default is TRUE.
constant	A scale factor for MAD calculation. Default is 1.4826, which ensures consistency with the standard deviation for normal distributions.

Details

Formula: $x' = \frac{x - \text{median}(x)}{\text{mad}(x)}$

Value

A numeric vector. If MAD is 0 (e.g., more than 50 returns a centered vector (x - median) and issues a warning.

References

Huber, P. J. (1981). *Robust Statistics*. Wiley. ISBN: 978-0-471-41805-4.

Hampel, F. R. (1974). The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346), 383-393.

Examples

```
# Data with an outlier
x <- c(1, 2, 3, 4, 100)

# Z-score is heavily affected by the outlier
norm_zscore(x)

# Robust scaler handles it better
norm_robust(x)
```

`norm_zscore` *Z-Score Standardization*

Description

Standardizes a numeric vector by centering it to have a mean of 0 and scaling it to have a standard deviation of 1.

Usage

```
norm_zscore(x, na.rm = TRUE)
```

Arguments

`x`	A numeric vector.
`na.rm`	Logical. Should NA values be removed during mean/sd calculation? Default is `TRUE`.

Details

Formula: $z = \frac{x - \mu}{\sigma}$

Value

A numeric vector. If the input vector has zero variance (all values are identical), the function returns a centered vector (all zeros) and issues a warning.

References

Han, J., Kamber, M., & Pei, J. (2011). *Data mining: concepts and techniques* (3rd ed.). Morgan Kaufmann.

Examples

```
# Standard usage
norm_zscore(c(1, 2, 3, 4, 5))

# Edge case: Zero variance
norm_zscore(c(5, 5, 5))
```

pp_plot *Visualize Distribution: Before vs After*

Description

Creates a comparison plot to visualize the effect of a transformation. It displays histograms and density curves for both the original and transformed data.

Usage

```
pp_plot(x, y, title = "Distribution Comparison")
```

Arguments

x	Numeric vector. The original data.
y	Numeric vector. The transformed data.
title	String. The main title of the plot.

Value

A ggplot object.

Examples

```
# 1. Generate skewed data
x <- rchisq(1000, df = 2)

# 2. Transform it
y <- trans_boxcox(x)

# 3. Visualize
pp_plot(x, y, title = "Box-Cox Transformation Effect")
```

sim_gait_data *Simulated Geriatric Gait Data*

Description

A synthetic longitudinal dataset representing daily step counts of an older adult. Used to demonstrate the "Vanishing Variance" problem.

Usage

```
data(sim_gait_data)
```

Format

A data frame with 200 rows and 2 variables:

day Integer. Time index (Days 1-200).

steps Numeric. Daily step count with habitual plateau and anomalies.

Source

Generated via simulation logic in data-raw/.

`trans_boxcox` *Box-Cox Transformation*

Description

Applies the Box-Cox transformation to normalize the data distribution. It automatically handles non-positive values by shifting the data. The optimal lambda parameter is estimated using Maximum Likelihood Estimation (MLE).

Usage

```
trans_boxcox(x, lambda = "auto", force_pos = TRUE)
```

Arguments

`x`	A numeric vector.
`lambda`	A numeric value for the transformation power. If `"auto"` (default), the optimal lambda is estimated within the interval [-2, 2].
`force_pos`	Logical. If `TRUE` (default), automatically shifts data to be positive if non-positive values are present.

Value

A numeric vector with the transformed values. The used `lambda` and `shift` amount are attached as attributes: `attr(res, "lambda")` and `attr(res, "shift")`.

References

Box, G. E. P., & Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological)*, 26(2), 211-243. https://www.jstor.org/stable/2984418

trans_log *Logarithmic Transformation*

Description

Applies a logarithmic transformation with an offset. Useful for handling right-skewed data.

Usage

```
trans_log(x, base = exp(1), offset = 1)
```

Arguments

x	A numeric vector.
base	A positive number. The base of the logarithm. Default is exp(1).
offset	A numeric value to add before taking the log. Default is 1.

Value

A numeric vector.

References

Bartlett, M. S. (1947). The use of transformations. *Biometrics*, 3(1), 39-52.

trans_yeojohnson *Yeo-Johnson Transformation*

Description

A power transformation similar to Box-Cox but supports both positive and negative values. Automatically estimates the optimal lambda using MLE.

Usage

```
trans_yeojohnson(x, lambda = "auto")
```

Arguments

x	A numeric vector.
lambda	A numeric value or "auto".

Value

A numeric vector with attribute "lambda".

References

Yeo, I.-K., & Johnson, R. A. (2000). A new family of power transformations to improve normality or symmetry. Biometrika.

Index