

Package ‘tidyrstats’

May 8, 2026

Type Package

Depends R (>= 4.1.0)

Title Tidy Common R Statistical Functions

Version 0.1.0

Maintainer Brendan Ansell <ansell.b@wehi.edu.au>

Description Provides functions to scale, log-transform and fit linear models within a 'tidyverse'-style R code framework.

Intended to smooth over inconsistencies in output of base R statistical functions, allowing ease of teaching, learning and daily use. Inspired by the tidy principles used in 'broom' Robinson (2017) <[doi:10.21105/joss.00341](https://doi.org/10.21105/joss.00341)>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Imports broom, glue, purrr, dplyr, rlang, stringr, stats

Suggests ggplot2, knitr, magrittr, rmarkdown

NeedsCompilation no

Author Brendan Ansell [aut, cre] (ORCID:
<<https://orcid.org/0000-0003-0297-897X>>)

Repository CRAN

Date/Publication 2025-06-11 13:00:02 UTC

Contents

lm_test	2
neg_log	3
scale_this	4

Index	5
--------------	----------

`lm_test`*Linear Model Testing for Grouped, Nested, or Ungrouped Data*

Description

Applies a linear model to a data frame and returns tidy model summaries. Supports ungrouped, grouped (`dplyr::group_by()`), and nested (`tidyr::nest_by()`) input data.

Usage

```
lm_test(input_data, formula)
```

Arguments

`input_data` A data frame or tibble. Can be ungrouped, grouped, or nested.
`formula` A model formula, either quoted or unquoted (e.g., `y ~ x * z`, or `"y ~ x * z"`).

Details

Designed to allow seamless 'in-line' chaining to fit linear models to columns of a tibble. Compatible with ungrouped, grouped or nested input. Compatible with native and magrittr pipe. Uses `broom::tidy()` to extract model summaries.

Value

A tibble with tidy model output sorted by p value, including:

term Model term (e.g., intercept, predictors, interactions)

estimate Estimated coefficient / beta

std.error Standard error of the estimate

statistic t-statistic

p.value p-value for the hypothesis test

If the input is grouped or nested, group identifiers are retained in the output. In the nested case, nested terms are relocated to the left-most column of the tibble.

Examples

```
library(ggplot2)
library(dplyr)

# Ungrouped
mpg |> lm_test(cty ~ hwy * cyl)

# Grouped
mpg |> group_by(class) |> lm_test(cty ~ hwy * cyl)

# Nested
```

```
mpg |> nest_by(class) |> lm_test(cty ~ hwy * cyl)
```

neg_log*Negative Logarithm (Base 10 by Default)*

Description

Computes the negative logarithm of a numeric input using base 10 by default.

Usage

```
neg_log(x, base = 10)
```

```
neglog(x, base = 10)
```

Arguments

x A numeric vector. Values must be positive.

base A numeric value specifying the base of the logarithm. Default is 10.

Details

This function returns the negative logarithm of 'x'. By default, it uses base 10, but you can specify a different base using the 'base' argument. Designed for quickly transforming p values for statistical analysis.

Value

A numeric vector of negative logarithmic values.

Examples

```
pvals <- 10^runif(10, -15, -1)
neg_log(pvals)
```

`scale_this`*Scale a numeric vector without converting to a matrix*

Description

This function scales and centres a numeric vector by subtracting the mean and dividing by the standard deviation. Unlike `scale()`, it returns a numeric vector, not a matrix. Note this function does not allow control over centering or scaling.

Usage

```
scale_this(x)
```

Arguments

`x` A numeric vector.

Value

A numeric vector of scaled values.

Examples

```
iris_dat <- head(iris$Sepal.Length)
scale_this(iris_dat)
scale_this(c(iris_dat, NA))
```

Index

`lm_test`, 2

`neg_log`, 3

`neglog(neg_log)`, 3

`scale_this`, 4