

Package ‘ptvalue’

October 31, 2025

Type Package

Title Working with Precision Teaching Values

Version 0.2.0

Description An implementation of an S3 class based on a double vector for storing and displaying precision teaching measures, representing a growing or a decaying (multiplicative) change between two frequencies. The main format method allows researchers to display measures (including data.frame) that respect the established conventions in the precision teaching community (i.e., prefixed multiplication or division symbol, displayed number ≤ 1). Basic multiplication and division methods are allowed and other useful functions are provided for creating, converting or inverting precision teaching measures. For more details, see Pennypacker, Gutierrez and Lindsley (2003, ISBN: 1-881317-13-7).

License MIT + file LICENSE

URL <https://github.com/agkamel/ptvalue>

BugReports <https://github.com/agkamel/ptvalue/issues>

Imports rlang, vctrs

Suggests testthat ($\geq 3.0.0$)

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation no

Author Alexandre Gellen-Kamel [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0002-0382-2478>>)

Maintainer Alexandre Gellen-Kamel <kamel.ag@outlook.com>

Repository CRAN

Date/Publication 2025-10-31 18:10:02 UTC

Contents

invert_sign	2
ptvalue	3
times	4

invert_sign	<i>Basic functions for converting pvalues</i>
-------------	-----------------------------------------------

Description

- `invert_sign()` inverts pvalues' sign.
- `abs_sign()` converts pvalues' sign to an 'absolute' sign, times or div.
- `as_times()` converts pvalues' sign to all times.
- `as_div()` converts pvalues' sign to all div.

Usage

```
invert_sign(x = double())
```

```
abs_sign(x = double(), sign = "times")
```

```
as_times(x = double())
```

```
as_div(x = double())
```

Arguments

- | | |
|-------------------|-----------------------------------------------------------------------------------------------------------------------|
| <code>x</code> | A vector of class ptvalue or a numeric vector. If a numeric vector is provided, values must be greater than 0. |
| <code>sign</code> | Either "times" or "div". Default to "times". |

Details

The function `invert_sign()` is the same as applying $\frac{1}{x}$ with the underlying numeric values of pvalues. Times values (\times) will be converted to div (\div) and div to times.

The function `abs_sign()` finds the multiplicative absolute values where all times and div values are converted to times values (by default). It can also convert to all div values by specifying `sign = "div"`.

Functions `as_times()` and `as_div()` are wrappers of `abs_sign()` where `sign = "times"` is specified for `as_times()` and `sign = "div"` is specified for `as_div()`.

All functions will return a vector of class **ptvalue** even when providing a numeric vector for `x` as it is the expected use. To see the underlying numeric vector, you can use `unclass()` or `as.double()`.

Value

A vector of class `ptvalue`.

Examples

```
x <- ptvalue(c(0.25, 0.5, 1, 2, 4))
x

abs_sign(x)
invert_sign(x)
as_times(x)
as_div(x)
```

ptvalue

ptvalue: Working with precision teaching values

Description

This class allow to print precision teaching measures with the times or the division symbols (ex. $\times 2$, $\div 1.4$) by converting numeric values to precision teaching values. More specifically, providing:

- Values between $]0, 1[$ will return output values ≥ 1 with a prefixed div (\div) symbol (ex. $\div 1$).
- Values between $[1, \infty[$ will return output values ≥ 1 with a prefixed times (\times) symbol.
- Values of 0 will return $\div \text{Inf}$.
- Inf values will return $\times \text{Inf}$.
- NA values will return NA.

Usage

```
ptvalue(x = double())

is_ptvalue(x)

as_ptvalue(x, ...)

## Default S3 method:
as_ptvalue(x, ...)
```

Arguments

`x` A numeric vector. Values must be ≥ 0 .

`...` Other values passed to method.

Value

A numeric vector of class **ptvalue** that represents precision teaching measures.

See Also

[times\(\)](#), [div\(\)](#)

Examples

```

# Basic examples
x <- c(0.5, 0.8, 1, 1.25, 2)
ptvalue(x)

ptvalue(0)
ptvalue(NA)
ptvalue(Inf)

# For convenience, `div()` can be used to
# create decaying values without using decimal values
ptvalue(c(0.5, 0.8))
div(c(2, 1.25))

x <- ptvalue(2)
is_ptvalue(x)

x <- 2
is_ptvalue(x)

x <- c(0.5, 1, 2)
as_ptvalue(x)

```

times

*Create times or div vector of class **ptvalue***

Description

`times()` and `div()` are convenient and stricter functions for creating growing or decaying precision teaching values with numeric values greater or equal than 1 (or otherwise raise an error).

Usage

```
times(x = double())
```

```
div(x = double())
```

Arguments

`x` A numeric vector. Values must be greater or equal than 1.

Details

Note that providing a vector of class **ptvalue** to `times()` or `div()` will raise an error as these functions are stricter.

Value

A numeric vector of class **ptvalue** that represent precision teaching measures.

Examples

```
x <- c(1, 2, 4)
times(x)
div(x)
```

```
# `div()` can be useful for specifying div values without
# the need to convert them first into values under zero
ptvalue(c(0.25, 0.5, 1, 2, 4))
ptvalue(c(div(4), div(2), 1, 2, 4))
```

Index

`abs_sign (invert_sign)`, 2
`as_div (invert_sign)`, 2
`as_ptvalue (ptvalue)`, 3
`as_times (invert_sign)`, 2

`div (times)`, 4
`div()`, 3

`invert_sign`, 2
`is_ptvalue (ptvalue)`, 3

`ptvalue`, 3

`times`, 4
`times()`, 3