Package 'descutils'

September 17, 2025

Title Utilities for Describing and Comparing Data

Version 1.0

Description Provides functionality that assists in tabular description and statistical comparison of data.

License GPL-3

Imports dplyr, stringr

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Andreas Leha [aut],
Fabian Kück [aut, cre]

Maintainer Fabian Kück <fabian.kueck@med.uni-goettingen.de>

Repository CRAN

Date/Publication 2025-09-17 10:40:02 UTC

Contents

decimalplaces *Count Decimal Places in a Number String*

Description

Computes the number of decimal places in the character representation of a number.

Usage

```
decimalplaces(x)
```

Arguments

x	A character string representing a number.

Value

Integer: The number of digits after the decimal point; `0` if there is no decimal point.

Examples

```
decimalplaces("3.141")   # returns 3
decimalplaces("42")      # returns 0
```

difftime_years *Calculate the Time Difference in Years*

Description

Code from Dirk Eddelbuettel via Stackoverflow (https://stackoverflow.com/a/15569373/1844418)

Usage

```
difftime_years(t2, t1)
```

Arguments

t2	end time of the interval. Will be coerced to Date
t1	starting time of the interval. Will be coerced to Date

Value

numeric. t2 - t1 [years]

Author(s)

Andreas Leha

Examples

```
difftime_years("2003-04-05", "2001-01-01")
```

---

digits2text *digits2text*

Description

English text version of number

Usage

```
digits2text(x, mult = "")
```

Arguments

x	number to convert
mult	to be appended (like a unit)

Details

Converts a number to a text version of that number

Value

character

Author(s)

Graham Williams <Graham.Williams@togawre.com> https://rattle.togaware.com/utility.R

---

formatC_mod *Modified formatC with Improved Zero and Exponential Handling*

Description

Formats a numeric vector similar to formatC, but tweaks output for certain special cases: - Attempts to remove trailing decimal points for whole numbers. - If format is 'fg' and output is "0", falls back to fixed format (f). - Converts exponential notation to plain numbers where appropriate.

Usage

```
formatC_mod(x, digits = NULL, format = NULL, ...)
```

Arguments

x	A numeric vector.
digits	Integer; desired number of digits.
format	Character string specifying the output format; see formatC. Common are "fg" (significant digits), or "f" (fixed).
...	Further arguments passed to formatC.

Value

A character vector with formatted numbers.

Examples

```
formatC_mod(c(3.14159, 42, 0), digits = 2, format = "fg")
formatC_mod(c(1e-6, 42), digits = 2, format = "fg")
```

---

formatIQR *generate [lower quartile; upper quartile]*

Description

wrapper around quantile() and format() and paste().

Usage

```
formatIQR(x, digits = 2, significant_digits = TRUE, ...)
```

Arguments

x	vector to be summarized
digits	Integer specifying the number of digits to display. Default is 2.
significant_digits	Logical. If TRUE, format numbers using significant digits; if FALSE, use fixed number of decimal digits. Default is TRUE.
...	passed onto quantile() as well as to format(). meant for arguments na.rm or digits

Value

character.

Author(s)

Dr. Andreas Leha

makeEnglishList *makeEnglishList*

Description

Print a text for English prosa

Usage

```
makeEnglishList(v, sep = ", ", lastsep = ", and ", onlysep = " and ")
```

Arguments

v	vector
sep	character. spearates all but last entries of v [", "]
lastsep	character. spearates the last entries of v [", and "]
onlysep	character. spearates the two entries of v if `length(v) == 2` [" and "]

Details

Pastes a vector and adds comma and "and" to the correct places

Value

character with plain text English prosa version

Author(s)

Andreas Leha

Examples

```
## defaut separators
makeEnglishList(c("foo", "bar", "baz"))
makeEnglishList(c("foo", "bar"))
makeEnglishList(c("foo"))

## without the 'Oxford comma'
makeEnglishList(c("foo", "bar", "baz"), lastsep = " and ")

## an 'or' list
makeEnglishList(c("foo", "bar", "baz"), lastsep = ", or ")
```

prettyCI *round and paste lower and upper (confidence) interval limits*

Description

Given 0.56 and 1.22998 this will return [0.56; 1.23].

Usage

```
prettyCI(conf.low, conf.high, digits = 2)
```

Arguments

conf.low	numeric vector. lower limits of the (confidence) interval
conf.high	numeric vector of the same length of conf.low. upper limits of the (confidence) interval.
digits	numeric. number of digits to round the limits to

Value

character of the form "[conf.low; conf.high]"

Author(s)

Dr. Andreas Leha

Examples

```
prettyCI(0.56, 1.22998)
```

prettyNumbers *prettyNumbers*

Description

Pretty print number

Usage

```
prettyNumbers(vals, digits = 5)
```

Arguments

vals	numeric
digits	how many significant digits to print

Value

character vector of pretty printed numbers

Author(s)

Fabian Kück

`prettyPvalues` *prettyPvalues*

Description

Pretty print p-values

Usage

```
prettyPvalues(
  p_vals,
  digits = 5,
  signiflev = 0.05,
  lhs = NULL,
  lhs_sep = "=",
  orgbold = TRUE,
  roundonly = FALSE
)
```

Arguments

`p_vals`	numeric
`digits`	how many digits to print
`signiflev`	print in bold face if smaller than this
`lhs`	character. left hand side of the printed 'formula'. Defaults to NULL, in which case neither lhs nor lhs_sep will be printed.
`lhs_sep`	character. Separator between lhs and pretty pvalue. Defaults to "<".
`orgbold`	boolean. Surround significant p values by '*'?. Defaults to TRUE.
`roundonly`	boolean. if TRUE, do neiter prepend any number with '<' nor do enclose any number with '*'. Defaults to FALSE

Details

Pretty print p-values with org-mode syntax for bold face

Value

character vector of pretty printed p-values

Author(s)

Andreas Leha

`wrapQuote` *wrap all elements in a vector in quotes (or other strings)*

Description

wrap all elements in a vector in quotes (or other strings)

Usage

```
wrapQuote(v, quoteChr = "'", endquoteChr = NULL)
```

Arguments

`v`	vector of elements to wrap
`quoteChr`	character. to be put around of the elements of `v`. Defaults to "'".
`endquoteChr`	character or NULL (default). If not NULL `quoteChr` is put before the elements of `v` and `endquoteChr` is put after them

Value

character vector of the elements of `v` wrapped between quotes

Author(s)

Andreas Leha

Examples

```
## default behaviour: wrap in single quotes
wrapQuote(1:10)

## change to wrap in asterisks
wrapQuote(1:10, "*")

## different front from back quotes
wrapQuote(1:10, "*", "/")

## you can also wrap with longer strings
wrapQuote(1:10, "quote")
```

Index