Package ‘nombre’

October 13, 2022

Title Number Names

Version 0.4.1

Description Converts numeric vectors to character vectors of English number names. Provides conversion to cardinals, ordinals, numerators, and denominators. Supports negative and non-integer numbers.

License MIT + file LICENSE

URL `https://nombre.rossellhayes.com`, `https://github.com/rossellhayes/nombre`

BugReports `https://github.com/rossellhayes/nombre/issues`

Depends R (>= 2.10)

Imports fracture (>= 0.2.1)

Suggests testthat (>= 3.0.0)

Encoding UTF-8

Language en-US

RoxygenNote 7.2.0

Config/testthat/edition 3

NeedsCompilation no

Author Alexander Rossell Hayes [aut, cre, cph] (<`https://orcid.org/0000-0001-9412-0457`>), Eli Pousson [ctb]

Maintainer Alexander Rossell Hayes <`alexander@rossellhayes.com`>

Repository CRAN

Date/Publication 2022-05-23 16:20:02 UTC

R topics documented:

adverbial 2
cardinal 3
collective 4

denominator 5
numerator 6
ordinal 6
ratio 7
uncardinal 9

Index **10**

`adverbial` *Convert numbers to adverbial character vectors (once, twice, three times)*

Description

Convert numbers to adverbial character vectors (once, twice, three times)

Usage

```
adverbial(x, thrice = FALSE, ...)

nom_adv(x, thrice = FALSE, ...)

nom_times(x, thrice = FALSE, ...)
```

Arguments

`x`	A numeric vector
`thrice`	A logical of length one. If `TRUE`, the adverbial of `3` will be "thrice". If `FALSE`, the adverbial of `3` will be "three times". Defaults to `FALSE`.
`...`	Additional arguments passed to `cardinal()`

Value

A character vector of the same length as `x`

See Also

Other number names: `cardinal()`, `collective()`, `denominator()`, `numerator()`, `ordinal()`, `ratio()`

Examples

```
nom_adv(1:4)
nom_adv(1:4, thrice = TRUE)
```

`cardinal` *Convert numbers to cardinal character vectors (one, two, three)*

Description

Convert numbers to cardinal character vectors (one, two, three)

Usage

```
cardinal(x, max_n = Inf, negative = "negative", ...)

nom_card(x, max_n = Inf, negative = "negative", ...)
```

Arguments

`x`	A numeric vector
`max_n`	A numeric vector. When the absolute value of `x` is greater than `max_n`, `x` remains numeric instead of being converted to words. If `max_n` is negative, no `x`s will be converted to words. (This can be useful when `max_n` is passed by another function.) Defaults to `Inf`, which converts all `x`s to words.
`negative`	A character vector to append to negative numbers. Defaults to `"negative"`.
`...`	Arguments passed on to `fracture::frac_mat`

`denom` If `denom` is not `NULL`, all fractions will have a denominator of `denom`. This will ignore all other arguments that affect the denominator.

`base_10` If `TRUE`, all denominators will be a power of 10.

`common_denom` If `TRUE`, all fractions will have the same denominator.
If the least common denominator is greater than `max_denom`, `max_denom` is used.

`max_denom` All denominators will be less than or equal to `max_denom`.
If `base_10` is `TRUE`, the maximum denominator will be the largest power of 10 less than `max_denom`.
A `max_denom` greater than the inverse square root of machine double epsilon will produce a warning because floating point rounding errors can occur when denominators grow too large.

Value

A character vector of the same length as `x`

Fractions

Decimal components of `x` are automatically converted to fractions by `fracture::frac_mat()`.

See Also

uncardinal() to convert character vectors to numbers

Other number names: adverbial(), collective(), denominator(), numerator(), ordinal(), ratio()

Examples

```
nom_card(2)
nom_card(1:10)
nom_card(2 + 4/9)
nom_card(-2)
nom_card(-2, negative = "minus")

nom_card(5:15, max_n = 10)

paste("There are", nom_card(525600), "minutes in a year.")
paste("There are", nom_card(3.72e13), "cells in the human body.")

nom_card(1 / 2^(1:4))
nom_card(1 / 2^(1:4), common_denom = TRUE)
nom_card(1 / 2^(1:4), base_10 = TRUE)
nom_card(1 / 2^(1:4), base_10 = TRUE, common_denom = TRUE)

nom_card(1 / 2:5)
nom_card(1 / 2:5, base_10 = TRUE)
nom_card(1 / 2:5, base_10 = TRUE, max_denom = 100)
```

---

collective — *Convert numbers to collective character vectors (the, both, all three)*

Description

Convert numbers to collective character vectors (the, both, all three)

Usage

```
collective(x, all_n = TRUE, of_the = FALSE, cardinal = TRUE, ...)

nom_coll(x, all_n = TRUE, of_the = FALSE, cardinal = TRUE, ...)
```

Arguments

x	A numeric vector.
all_n	Whether to include the cardinal number after "all" for collectives of 3 or more. Defaults to TRUE.
of_the	Whether to include "of the" for collectives other than 1. Defaults to FALSE.
cardinal	Whether to convert the number after "all" with cardinal() when all_n is TRUE. Defaults to TRUE.
...	Additional arguments passed to cardinal() when cardinal is TRUE.

Value

A character vector of the same length as `x`.

See Also

Other number names: `adverbial()`, `cardinal()`, `denominator()`, `numerator()`, `ordinal()`, `ratio()`

Examples

```
paste(nom_coll(0:3), "fish")
paste(nom_coll(9:12, max_n = 10), "fish")
```

`denominator` *Convert numbers to denominator character vectors (whole, half, third)*

Description

Convert numbers to denominator character vectors (whole, half, third)

Usage

```
denominator(x, numerator = 1, quarter = TRUE, ...)

nom_denom(x, numerator = 1, quarter = TRUE, ...)
```

Arguments

`x`	A numeric vector
`numerator`	A numeric vector. The numerator(s) associated with the denominator(s). When `numerator` is not `1` or `-1`, the denominator will be pluralized.
`quarter`	A logical of length one. If `TRUE`, the denominator of `4` will be "quarter(s)". If `FALSE`, the denominator of `4` will be "fourth(s)". Defaults to `TRUE`.
`...`	Additional arguments passed to `ordinal()`

Value

A character vector of the same length as `x`

See Also

Other number names: `adverbial()`, `cardinal()`, `collective()`, `numerator()`, `ordinal()`, `ratio()`

Examples

```
nom_denom(2)
nom_denom(1:10)
nom_denom(1:10, numerator = 2)
nom_denom(1:10, numerator = 1:10)

nom_denom(4)
nom_denom(4, quarter = FALSE)

nom_denom(1:10, numerator = 2, cardinal = FALSE)
nom_denom(5:15, numerator = 2, max_n = 10)
```

`numerator` *Convert numbers to numerator character vectors (one, two, three)*

Description

`nom_numer()` and `numerator()` are equivalent to `nom_card()` and `cardinal()` for integers, but cardinals support fractional components while `numerator`s do not.

Usage

```
numerator(x, ...)

nom_numer(x, ...)
```

Arguments

`x`	A numeric vector
`...`	Additional arguments passed to `cardinal()`

See Also

Other number names: `adverbial()`, `cardinal()`, `collective()`, `denominator()`, `ordinal()`, `ratio()`

`ordinal` *Convert numbers to ordinal character vectors (first, second, third)*

Description

Adds ordinal suffixes to numbers (or a character vector of number-like words). Converts numeric vectors to cardinal numbers before adding prefixes unless `cardinal` is `FALSE`.

Usage

```
ordinal(x, cardinal = TRUE, ...)

nom_ord(x, cardinal = TRUE, ...)
```

Arguments

`x`	A numeric or character vector.
`cardinal`	Whether to convert a numeric vector with `cardinal()` before applying ordinal suffixes. When `TRUE`, 1 -> "first". When `FALSE`, 1 -> "1st". Defaults to `TRUE`.
`...`	Further arguments passed to `cardinal()` when `cardinal` is `TRUE`.

Value

A character vector of the same length as `x`

See Also

Other number names: `adverbial()`, `cardinal()`, `collective()`, `denominator()`, `numerator()`, `ratio()`

Examples

```
nom_ord(2)
nom_ord(1:10)
nom_ord(525600)

nom_ord(1:10, cardinal = FALSE)
nom_ord(5:15, max_n = 10)

nom_ord(c("n", "dozen", "umpteen", "eleventy", "one zillion"))
nom_ord(9 + 3/4)
```

`ratio` *Convert numbers to ratio character vectors (two to one, one in three, five out of ten)*

Description

Convert numbers to ratio character vectors (two to one, one in three, five out of ten)

Usage

```
ratio(x, sep = "in", max_n = Inf, negative = "negative", ...)

nom_ratio(x, sep = "in", max_n = Inf, negative = "negative", ...)
```

Arguments

`x`	A numeric vector
`sep`	A character vector separating components of the ratio. Defaults to `"in"`.
`max_n`	A numeric vector. When the absolute value of `x` is greater than `max_n`, `x` remains numeric instead of being converted to words. If `max_n` is negative, no `x`s will be converted to words. (This can be useful when `max_n` is passed by another function.) Defaults to `Inf`, which converts all `x`s to words.
`negative`	A character vector to append to negative numbers. Defaults to `"negative"`.
`...`	Arguments passed on to `fracture::frac_mat`

`denom` If `denom` is not `NULL`, all fractions will have a denominator of `denom`. This will ignore all other arguments that affect the denominator.

`base_10` If `TRUE`, all denominators will be a power of 10.

`common_denom` If `TRUE`, all fractions will have the same denominator. If the least common denominator is greater than `max_denom`, `max_denom` is used.

`max_denom` All denominators will be less than or equal to `max_denom`. If `base_10` is `TRUE`, the maximum denominator will be the largest power of 10 less than `max_denom`. A `max_denom` greater than the inverse square root of machine double epsilon will produce a warning because floating point rounding errors can occur when denominators grow too large.

Details

`x` is converted to a fraction by `fracture::frac_mat()`.

Value

A character vector of the same length as `x`

See Also

Other number names: `adverbial()`, `cardinal()`, `collective()`, `denominator()`, `numerator()`, `ordinal()`

Examples

```
paste0("Our team is outnumbered ", nom_ratio(10), ".")
paste0("The chances of winning are ", nom_ratio(1/1000000, sep = "in"), ".")

nom_ratio(c(1, 10, 100))
nom_ratio(c(0, 0.5, 1.5))
nom_ratio(c(0, 0.125, 0.625, 1), sep = "out of", common_denom = TRUE)
nom_ratio(5 / 10, sep = "in", base_10 = TRUE)
nom_ratio(6 / 25, sep = "in")
nom_ratio(6 / 25, sep = "out of", max_denom = 10)
```

`uncardinal` *Convert cardinal character vectors to numbers*

Description

This function is in experimental development. It currently only supports English cardinal integers or character vectors produced by one of nombre's functions.

Usage

```
uncardinal(x)

nom_uncard(x)
```

Arguments

`x`	A character vector of the cardinal names of numbers

Value

A numeric vector the same length as `n`. NAs will be produced for numbers with fractions or decimals or non-cardinal numbers (e.g. ordinals).

See Also

`cardinal()` to convert numeric vectors to number names

Examples

```
uncardinal("one")
uncardinal("negative one hundred fifty-seven")
uncardinal(
  c(
    "twenty-five",
    "one million two hundred thirty-four thousand five hundred sixty-seven"
  )
)
uncardinal("infinity")

card <- cardinal(25)
uncardinal(card)
ord <- ordinal(25)
uncardinal(ord)
```

Index

∗ **number names**
  adverbial, 2
  cardinal, 3
  collective, 4
  denominator, 5
  numerator, 6
  ordinal, 6
  ratio, 7

adverbial, 2, *4–8*

cardinal, *2*, 3, *5–8*
cardinal(), *2*, *4*, *6*, *7*, *9*
collective, *2*, *4*, 4, *5–8*

denominator, *2*, *4*, *5*, 5, *6–8*

fracture::frac_mat, *3*, *8*
fracture::frac_mat(), *3*, *8*

machine double epsilon, *3*, *8*

nom_adv *(*adverbial*)*, 2
nom_card *(*cardinal*)*, 3
nom_card(), *6*
nom_coll *(*collective*)*, 4
nom_denom *(*denominator*)*, 5
nom_numer *(*numerator*)*, 6
nom_ord *(*ordinal*)*, 6
nom_ratio *(*ratio*)*, 7
nom_times *(*adverbial*)*, 2
nom_uncard *(*uncardinal*)*, 9
nombre, *9*
NULL, *3*, *8*
numerator, *2*, *4*, *5*, 6, *7*, *8*

ordinal, *2*, *4–6*, 6, *8*
ordinal(), *5*

ratio, *2*, *4–7*, 7

uncardinal, 9
uncardinal(), *4*