

# Package ‘nombre’

September 12, 2020

**Title** Number Names

**Version** 0.3.0

**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 (>= 0.1.2), pkgconfig

**Suggests** covr, rlang, testthat, waldo

**Encoding** UTF-8

**Language** en-US

**LazyData** true

**RoxygenNote** 7.1.1

**NeedsCompilation** no

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

**Maintainer** Alexander Rossell Hayes <[alexander@rossellhayes.com](mailto:alexander@rossellhayes.com)>

**Repository** CRAN

**Date/Publication** 2020-09-12 04:50:02 UTC

## R topics documented:

adverbial	2
cardinal	3
collective	4

denominator . . . . .	5
numerator . . . . .	7
ordinal . . . . .	7
set_config . . . . .	8
uncardinal . . . . .	9

<b>Index</b>	<b>11</b>
--------------	-----------

---

adverbial	<i>Convert numbers to adverbial character vectors (once, twice, three times)</i>
-----------	--

---

### Description

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

### Usage

```
adverbial(x, thrice = get_config("nombre::thrice", FALSE), ...)
```

```
nom_adv(x, thrice = get_config("nombre::thrice", FALSE), ...)
```

```
nom_times(x, thrice = get_config("nombre::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. Default can be changed with <a href="#">set_config("nombre::thrice")</a> .
...	Additional arguments passed to <a href="#">cardinal()</a>

### Value

A character vector of the same length as x

### See Also

Other number names: [cardinal\(\)](#), [collective\(\)](#), [denominator\(\)](#), [numerator\(\)](#), [ordinal\(\)](#)

### Examples

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

---

cardinal	<i>Convert numbers to cardinal character vectors (one, two, three)</i>
----------	--

---

### Description

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

### Usage

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

nom_card(
  x,
  max_n = get_config("nombre::max_n", Inf),
  negative = get_config("nombre::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 xs will be converted to words. (This can be useful when max_n is passed by another function.) Defaults to Inf, which converts all xs to words. Default can be changed with <code>set_config("nombre::max_n")</code> .
negative	A character vector to append to negative numbers. Defaults to "negative". Default can be changed with <code>set_config("nombre::negative")</code> .
...	Additional arguments passed to <code>fracture::frac_mat()</code> . See details.

### Value

A character vector of the same length as x

### Fractions:

Decimal components of x are automatically converted to fractions by `fracture::frac_mat()`. Named arguments of `cardinal()` and `nom_card()` are passed to `fracture::frac_mat()` through ... Helpful arguments include:

- `base_10 = TRUE`, which forces all fractions to use denominators that are powers of ten

- `common_denom = TRUE`, which forces all fractions to use the least common denominator when `x` is a vector
- `max_denom`, which sets the maximum allowable denominator. By default, the maximum denominator is "ten-millionths".

### See Also

[uncardinal\(\)](#) to convert character vectors to numbers

Other number names: [adverbial\(\)](#), [collective\(\)](#), [denominator\(\)](#), [numerator\(\)](#), [ordinal\(\)](#)

### 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 = get_config("nombre::all_n", TRUE),
  of_the = get_config("nombre::of_the", FALSE),
  cardinal = get_config("nombre::coll_cardinal", TRUE),
  ...
)
```

```

nom_coll(
  x,
  all_n = get_config("nombre::all_n", TRUE),
  of_the = get_config("nombre::of_the", FALSE),
  cardinal = get_config("nombre::coll_cardinal", TRUE),
  ...
)

```

### Arguments

<code>x</code>	A numeric vector.
<code>all_n</code>	Whether to include the cardinal number after "all" for collectives of 3 or more. Defaults to TRUE. Default can be changed with <code>set_config("nombre::all_n")</code> .
<code>of_the</code>	Whether to include "of the" for collectives other than 1. Defaults to FALSE. Default can be changed with <code>set_config("nombre::of")</code> .
<code>cardinal</code>	Whether to convert the number after "all" with <code>cardinal()</code> when <code>all_n</code> is TRUE. Defaults to TRUE. Default can be changed with <code>set_config("nombre::coll_cardinal")</code> .
<code>...</code>	Additional arguments passed to <code>cardinal()</code> when <code>cardinal</code> is TRUE.

### Value

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

### See Also

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

### Examples

```

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

```

---

denominator	<i>Convert numbers to denominator character vectors (whole, half, third)</i>
-------------	--

---

### Description

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

**Usage**

```
denominator(
  x,
  numerator = 1,
  quarter = get_config("nombre::quarter", TRUE),
  ...
)

nom_denom(x, numerator = 1, quarter = get_config("nombre::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. Default can be changed with <a href="#">set_config("nombre::quarter")</a> .
...	Additional arguments passed to <a href="#">ordinal()</a>

**Value**

A character vector of the same length as x

**See Also**

Other number names: [adverbial\(\)](#), [cardinal\(\)](#), [collective\(\)](#), [numerator\(\)](#), [ordinal\(\)](#)

**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 numerators 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()`

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 = get_config("nombre::ord_cardinal", TRUE), ...)
```

```
nom_ord(x, cardinal = get_config("nombre::ord_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`. Default can be changed with `set_config("nombre::ord_cardinal")`.  
`...`                 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\(\)](#)

**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)
```

---

set\_config

*Set a configuration parameter*

---

**Description**

Set a configuration parameter, for the package we are calling from. If called from the R prompt and not from a package, then it sets the parameter for global environment.

**Usage**

```
set_config(...)
```

**Arguments**

... Parameters to set, they should be all named.

**Value**

Unlike `pkgconfig::set_config()`, invisibly returns the previous values of all the parameters changed, similar to `options()`. This allows previous values to be stored in a variable and restored later.

**See Also**

[pkgconfig::get\\_config\(\)](#) and [pkgconfig::set\\_config\(\)](#)



## Examples

```
nom_adv(3)
default_adv <- set_config("nombre::thrice" = TRUE)
nom_adv(3)

nom_card(c(-1, 1e4))
default_card <- set_config("nombre::max_n" = 100, "nombre::negative" = "minus")
nom_card(c(-1, 1e4))

nom_denom(4)
default_denom <- set_config("nombre::quarter" = FALSE)
nom_denom(4)

set_config(default_adv, default_card, default_denom)
nom_adv(3)
nom_card(c(-1, 1e4))
nom_denom(4)
```

---

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, 7
  - ordinal, 7
- adverbial, 2, 4–8
- cardinal, 2, 3, 5–8  
cardinal(), 2, 5, 7, 9  
collective, 2, 4, 4, 6–8
- denominator, 2, 4, 5, 5, 7, 8
- fracture::frac\_mat(), 3
- nom\_adv (adverbial), 2  
nom\_card (cardinal), 3  
nom\_card(), 7  
nom\_coll (collective), 4  
nom\_denom (denominator), 5  
nom\_numer (numerator), 7  
nom\_ord (ordinal), 7  
nom\_times (adverbial), 2  
nom\_uncard (uncardinal), 9  
nombre, 9  
numerator, 2, 4–6, 7, 8
- options(), 8  
ordinal, 2, 4–7, 7  
ordinal(), 6
- pkgconfig::get\_config(), 8  
pkgconfig::set\_config(), 8
- set\_config, 8  
set\_config(nombre::all\_n), 5  
set\_config(nombre::coll\_cardinal), 5  
set\_config(nombre::max\_n), 3
- set\_config(nombre::negative), 3  
set\_config(nombre::of), 5  
set\_config(nombre::ord\_cardinal), 7  
set\_config(nombre::quarter), 6  
set\_config(nombre::thrice), 2
- uncardinal, 9  
uncardinal(), 4