

# Package ‘cnum’

July 22, 2025

**Type** Package

**Title** Chinese Numerals Processing

**Version** 0.1.5

**Maintainer** Elgar Teo <elgarteo@connect.hku.hk>

**URL** <https://github.com/elgarteo/cnum/>

**BugReports** <https://github.com/elgarteo/cnum/issues>

**Description** Chinese numerals processing in R, such as conversion between Chinese numerals and Arabic numerals as well as detection and extraction of Chinese numerals in character objects and string. This package supports the casual scale naming system and the respective SI prefix systems used in mainland China and Taiwan:  
`` The State Council's Order on the Unified Implementation of Legal Measurement Units in Our Country"  
The State Council of the People's Republic of China (1984)  
`` Names, Definitions and Symbols of the Legal Units of Measurement and the Decimal Multiples and Submultiples"  
Ministry of Economic Affairs (2019) <<https://gazette.nat.gov.tw/egFront/detail.do?metaid=108965>>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Depends** R(>= 2.10)

**Imports** stringr, Rcpp

**Suggests** magrittr

**LinkingTo** Rcpp, BH

**RoxygenNote** 7.3.2

**NeedsCompilation** yes

**Author** Elgar Teo [aut, cre]

**Repository** CRAN

**Date/Publication** 2025-01-11 22:00:02 UTC

## Contents

conversion . . . . .	2
default_cnum_lang . . . . .	4
tools . . . . .	5

<b>Index</b>	<b>8</b>
--------------	----------

---

conversion	<i>Chinese Numerals Conversion</i>
------------	------------------------------------

---

## Description

Functions to convert between Chinese and Arabic numerals.

## Usage

```
c2num(
  x,
  lang = default_cnum_lang(),
  mode = "casual",
  financial = FALSE,
  literal = FALSE
)
```

```
num2c(
  x,
  lang = default_cnum_lang(),
  mode = "casual",
  financial = FALSE,
  literal = FALSE,
  single = FALSE
)
```

## Arguments

x	the Arabic/Chinese numerals to be converted, or a vector of them. The absolute value must not be greater than 1e+18.
lang	the language of the Chinese numerals. "tc" for Traditional Chinese. "sc" for Simplified Chinese. The default is "tc", but this can be changed by setting <code>options(cnum.lang = "sc")</code> .
mode	the scale naming system to be enforced. See the 'Details' section for the list of supported modes.
financial	logical: should the financial numerals be used ( <i>daxie shuzi</i> )?
literal	logical: should the numerals be converted literally? (e.g. 721 to be converted to "qi er yi" instead of "qibai ershiyi" and vice versa)
single	logical: should the return result with one scale character only? (e.g. 1.5e+08 as "yi dian wuyi" instead of "yiyi wuqianwan")

## Value

c2num returns a numeric vector.

num2c returns a character vector.

## Functions

- c2num(): Convert Chinese Numerals to Arabic Numerals.
- num2c(): Convert Arabic Numerals to Chinese Numerals.

## Details

The following scale naming systems are supported:

- "casual": the casual naming system used outside of mainland China, i.e.  $1e+09$  is referred to as "yi zhao".
- "casualPRC": the casual naming system used in mainland China, i.e.  $1e+9$  is referred to as "yi wanyi".
- "SIprefix": the SI prefix system used in Taiwan as stipulated in the document *Names, Definitions and Symbols of the Legal Units of Measurement and the Decimal Multiples and Submultiples*.
- "SIprefixPRC": the SI prefix system used in mainland China as stipulated in the document *China Statutory Measurement Units*.
- "SIprefixPRClong": a variant of "SIprefixPRC" with long prefixes, e.g.  $1e+09$  is referred to as "yi jika" instead of "yi ji".

## Warnings

The modes "casual" and "casualPRC" implements a "myriad scale" with an interval of  $1e+04$  for large numbers, i.e. "yi" is 10,000 times of "wan", which is different from some of the interval systems used in ancient Chinese writings.

This package supports conversion of numbers with absolute value not greater than  $1e+18$ . Note that numbers in R are in double precision that carries approximately 16 significant digits. The conversion accuracy for numbers beyond this limit is therefore not guaranteed.

## References

The standard for mode "SIprefix" *Names, Definitions and Symbols of the Legal Units of Measurement and the Decimal Multiples and Submultiples* is available from <https://gazette.nat.gov.tw/egFront/detail.do?metaid=108965> (in Traditional Chinese).

The standard for mode "SIprefixPRC" *The State Council's Order on the Unified Implementation of Legal Measurement Units in Our Country* is available from the PRC State Council's website (in Simplified Chinese).

## See Also

[Functions for detection and extraction](#)

## Examples

```
c2num("EXAMPLE CHECK")

num2c(721)
num2c(-6)
num2c(3.14)
num2c(721, literal = TRUE)
num2c(1.45e4, financial = TRUE)
num2c(6.85e4, lang = "sc", mode = "casualPRC")
num2c(1.5e4, mode = "SIprefix", single = TRUE)
```

---

default_cnum_lang	<i>Default Language for cnum</i>
-------------------	----------------------------------

---

## Description

Function to check the default language for cnum functions.

## Usage

```
default_cnum_lang()
```

## Details

This package supports Traditional Chinese and Simplified Chinese. The language can be specified with the lang parameter in every function, with "tc" for Traditional Chinese and "sc" for Simplified Chinese. The default is "tc", but this can be changed by setting `options(cnum.lang = "sc")`.

## Value

The default language for cnum functions.

## See Also

- [Functions for conversion](#)
- [Functions for detection and extraction](#)

## Examples

```
# Set the default language to Simplified Chinese
options(cnum.lang = "sc")

default_cnum_lang()
```

## Description

Functions to detect and extract Chinese numerals in character object and string.

## Usage

```
is_cnum(  
  x,  
  lang = default_cnum_lang(),  
  mode = "casual",  
  financial = FALSE,  
  literal = FALSE,  
  strict = FALSE,  
  ...  
)
```

```
has_cnum(  
  x,  
  lang = default_cnum_lang(),  
  mode = "casual",  
  financial = FALSE,  
  ...  
)
```

```
extract_cnum(  
  x,  
  lang = default_cnum_lang(),  
  mode = "casual",  
  financial = FALSE,  
  prefix = NULL,  
  suffix = NULL,  
  ...  
)
```

## Arguments

<code>x</code>	the character object or string to be tested or to extract from.
<code>lang</code>	the language of the Chinese numerals. "tc" for Traditional Chinese. "sc" for Simplified Chinese. The default is "tc", but this can be changed by setting <code>options(cnum.lang = "sc")</code> .
<code>mode</code>	the scale naming system to be enforced. See the 'Details' section for the list of supported modes.
<code>financial</code>	logical: should the financial numerals be used ( <i>daxie shuzi</i> )?

literal	logical: should the numerals be converted literally? (e.g. 721 to be converted to "qi er yi" instead of "qibai ershiyi" and vice versa)
strict	logical: Should the Chinese numerals format be strictly enforced? A casual test only checks if x contains Chinese numerals characters. A strict test checks if x is valid Chinese numerals. (e.g. "yi bai yi" will pass the casual test and fail the strict test)
...	optional arguments to be passed to <code>grepl</code> (for <code>is_cnum</code> and <code>has_cnum</code> ) or <code>str_extract_all</code> (for <code>extract_cnum</code> ). Disregarded when <code>strict = TRUE</code> .
prefix	the prefix of the Chinese numerals. Only numerals with the designated prefix are extracted. Supports <a href="#">regular expression(s)</a> .
suffix	the suffix of the Chinese numerals. Only numerals with the designated suffix are extracted. Supports <a href="#">regular expression(s)</a> .

### Value

`is_cnum` returns a logical vector indicating is Chinese numerals or not for each element of `x`).

`has_cnum` returns a logical vector indicating contains Chinese numerals or not for each element of `x`.

`extract_cnum` returns a list of character vectors containing the extracted Chinese numerals.

### Functions

- `is_cnum()`: Test if character object is Chinese numerals. A wrapper around `grepl`.
- `has_cnum()`: Test if string contains Chinese numerals. A wrapper around `grepl`.
- `extract_cnum()`: Extracts Chinese numerals from string. A wrapper around `str_extract_all` from `stringr`.

### Details

The following scale naming systems are supported:

- "casual": the casual naming system used outside of mainland China, i.e.  $1e+09$  is referred to as "yi zhao".
- "casualPRC": the casual naming system used in mainland China, i.e.  $1e+9$  is referred to as "yi wanyi".
- "SIprefix": the SI prefix system used in Taiwan as stipulated in the document *Names, Definitions and Symbols of the Legal Units of Measurement and the Decimal Multiples and Sub-multiples*.
- "SIprefixPRC": the SI prefix system used in mainland China as stipulated in the document *China Statutory Measurement Units*.
- "SIprefixPRClong": a variant of "SIprefixPRC" with long prefixes, e.g.  $1e+09$  is referred to as "yi jika" instead of "yi ji".

## References

The standard for mode "SIprefix" *Names, Definitions and Symbols of the Legal Units of Measurement and the Decimal Multiples and Submultiples* is available from <https://gazette.nat.gov.tw/egFront/detail.do?metaid=108965> (in Traditional Chinese).

The standard for mode "SIprefixPRC" *The State Council's Order on the Unified Implementation of Legal Measurement Units in Our Country* is available from the PRC State Council's website (in Simplified Chinese).

## See Also

[Functions for conversion](#)

## Examples

```
is_cnum("yibai ershiyi")
```

```
has_cnum("yibai bashi yuan")
```

```
extract_cnum("shisiyi ren")
```

# Index

`c2num` (conversion), [2](#)  
conversion, [2](#)

`default_cnum_lang`, [4](#)

`extract_cnum` (tools), [5](#)

Functions for conversion, [4, 7](#)

Functions for detection and  
extraction, [3, 4](#)

`grepl`, [6](#)

`has_cnum` (tools), [5](#)

`is_cnum` (tools), [5](#)

`num2c` (conversion), [2](#)

options, [2, 4, 5](#)

regular expression, [6](#)

`str_extract_all`, [6](#)

tools, [5](#)