

Package: cncleanr (via r-universe)

October 18, 2026

Title Parse Compact Numbers in Chinese Data
Version 0.2.19
Description Parses compact numeric values, qualified quantities, and ranges commonly found in Chinese tables and spreadsheets. It handles Chinese magnitude suffixes, currencies, percentages, full-width characters, financial negatives, and configurable missing-value markers while reporting values that cannot be parsed safely.
License MIT + file LICENSE
URL <https://github.com/Jorungandr/cncleanr>
BugReports <https://github.com/Jorungandr/cncleanr/issues>
Encoding UTF-8
NeedsCompilation yes
Language en-US
Roxygen list(markdown = TRUE)
Suggests spelling, testthat (>= 3.0.0)
Config/testthat/edition 3
Config/roxygen2/version 8.1.0
Repository <https://jorungandr.r-universe.dev>
Date/Publication 2026-10-18 03:12:46 UTC
RemoteUrl <https://github.com/jorungandr/cncleanr>
RemoteRef HEAD
RemoteSha b21289ed93cd8f926b9e6d05ee552eeae81b941b

Contents

cn_problems	2
cn_subset	2
cn_summary	3

parse_cn_number	4
parse_cn_quantity	5
parse_cn_range	6

Index	8
--------------	----------

cn_problems	<i>Extract Problems From a cncleanr Result</i>
-------------	--

Description

Returns parsing failures attached to an object produced by a cncleanr parser. Successful results return an empty data frame with the same stable schema. Use [cn_subset\(\)](#) to select or reorder rows with their diagnostics. Other base R transformations may drop the attribute or retain stale indices.

Usage

```
cn_problems(x)
```

Arguments

x An object returned by a cncleanr parsing function.

Value

A data frame with columns index, value, and reason.

Examples

```
result <- suppressWarnings(parse_cn_number(c("bad", "2\u4e07")))  
cn_problems(result)
```

cn_subset	<i>Select Parsed Results With Their Problems</i>
-----------	--

Description

Explicitly subsets or reorders an unmodified parser result and remaps its problem indices to positions in the selected result. Repeated selections repeat the corresponding problems. Ordinary `[]` does not track problems.

Usage

```
cn_subset(x, rows)
```

Arguments

x	An unmodified result from a cncleanr parser, or from <code>cn_subset()</code> .
rows	Integer-valued numeric indices (positive, zero, or negative using base R rules), or a logical vector of exactly the input length/row count. Missing, nonfinite, fractional and out-of-range indices are rejected. Defaults to all rows. Use <code>integer()</code> to select no rows. Classed numeric indices use <code>as.double()</code> once before validation.

Details

Call this before other transformations. It cannot recover problems lost or made stale by ordinary subsetting, mutation, sorting or binding. It does not reparse values or emit another parsing warning.

Value

The selected numeric vector or data frame, without a new class. Problem indices refer to the selected result, not the original input.

Examples

```
result <- suppressWarnings(parse_cn_number(c("bad", "2", "wrong")))
selected <- cn_subset(result, c(3, 2, 1, 3))
cn_problems(selected)
cn_subset(result, !is.na(result))
```

cn_summary

*Summarize Parsing Outcomes and Failure Reasons***Description**

Counts successful, missing and failed rows without reparsing source data.

Usage

```
cn_summary(x)
```

Arguments

x	An unmodified result from a cncleanr parser or cn_subset() .
---	--

Details

NA and NaN rows count as missing unless their positions occur in the attached problems table, in which case they count as failed. Numeric infinities and textual open range bounds are valid parsed values. Obvious malformed results or stale problem indices cause an error. Plain R results do not prove provenance: diagnostics removed or made stale by other transformations cannot always be detected or recovered. Summarize before ordinary subsetting, mutation, binding or CSV export, or use [cn_subset\(\)](#). Only this explicit summary scans result values; existing parsers do no extra summary work. No parsing warnings are repeated and x is not modified.

Value

A list with two ordinary data frames. counts has one row with columns total, parsed, missing, failed. reasons has columns reason and count, in first-occurrence order. Empty results have zero counts and a zero-row reason table. Repeated selections count repeatedly.

Examples

```
result <- suppressWarnings(parse_cn_number(c("3\u4e07", "bad", "\u6682\u65e0")))
cn_summary(result)
cn_summary(cn_subset(result, c(2, 1, 2)))
```

parse_cn_number	<i>Parse Compact Numbers Used in Chinese Data</i>
-----------------	---

Description

Converts character values such as "1.25\u4e07", "\uffe53\u4ebf", "1.2e5", and "12.5%" to doubles. Parsing is deliberately conservative: every non-missing value must match the supported syntax in full. Spaces between digits are accepted only when they form valid three-digit grouping.

Usage

```
parse_cn_number(
  x,
  na = c(
    "", "NA", "N/A", "\u6682\u65e0", "\u672a\u516c\u5e03",
    "\u2014", "\u2013", "...", "\u2026"
  ),
  strict = FALSE,
  unit = NULL
)
```

Arguments

x	A character, factor, or numeric vector. When unit = NULL, numeric input is converted to double and otherwise passed through unchanged, including Inf, -Inf, and NaN. Numeric input uses as.double() (including applicable S3 methods) before header-unit scaling and missing-value detection. Conversion methods must return a plain double vector of the original length, without a class or dimensions; malformed results cause an error. SPSS user-defined missing meta-data is respected. Objects with class units are rejected: convert units explicitly, then call units::drop_units().
na	A character vector containing values that should be interpreted as missing. Matching happens after whitespace and full-width normalization.
strict	A single logical value. If FALSE, invalid values become NA_real_, a warning is emitted, and a data frame is stored in the problems attribute. If TRUE, invalid values cause an error.

unit A single optional table-header unit: yuan, thousand yuan, ten thousand, hundred million, or trillion, optionally in yuan. Use the spellings `"\u5143"`, `"\u5343\u5143"`, `"\u4e07"`, `"\u4e07\u5143"`, `"\u4ebf"`, `"\u4ebf\u5143"`, `"\u4e07\u4ebf"`, or `"\u4e07\u4ebf\u5143"`. Bare values inherit this unit; explicit units must have the same multiplier and are not scaled twice. Percentages conflict with a header unit. With a header unit, numeric input follows character parsing: missing values remain missing, while infinities and overflow are reported as problems. Defaults to NULL (no header unit).

Details

Finite-range validation applies to parsed character and factor input. With `unit = NULL`, existing numeric Inf, -Inf, and NaN values are preserved. Results use double precision, not exact integer or decimal arithmetic. Nonzero character values rounded to zero after final unit conversion are reported as underflow; representable subnormal values are retained.

Value

A double vector with the same length and names as `x`. When invalid values occur in non-strict mode, the result has a `problems` attribute with columns `index`, `value`, and `reason`.

Examples

```
parse_cn_number(c("1.25\u4e07", "3\u4ebf\u5143", "12.5%", "\u6682\u65e0"))
parse_cn_number(c("\uff11\uff12\uff0e\uff15\uff05", "(2.5\u4e07)"))
parse_cn_number(c("123.5", "1.5\u5343\u5143"), unit = "\u5343\u5143")
```

parse_cn_quantity

Parse Qualified Quantities Used in Chinese Data

Description

Parses exact values and values qualified by language or symbols, such as `"\u7ea6\u4e07"`, `"\u5927\u4e8e\u4ebf"`, `"\u4e0d\u5c0f\u4e8e\u4e07"`, `"10\u4e07+"`, `"50\u4f59"`, or `"50\u4f59\u4e07\u5143"`. The qualifier is retained instead of silently discarded. One approximation prefix combined with an approximation suffix is accepted, for example `"\u7ea615%\u5de6\u53f3"`. Repeated prefixes or suffixes and mixed inequality qualifiers are rejected.

Usage

```
parse_cn_quantity(
  x,
  na = c(
    "", "NA", "N/A", "\u6682\u65e0", "\u672a\u516c\u5e03",
    "\u2014", "\u2013", "\u2014", "\u2013", "\u2014", "\u2013"
  ),
  strict = FALSE
)
```

Arguments

x	A character, factor, or numeric vector. Numeric input is converted to double and otherwise passed through unchanged, including Inf, -Inf, and NaN. Conversion methods must return a plain double vector of the original length, without a class or dimensions; malformed results cause an error. SPSS user-defined missing metadata is respected. Objects with class units are rejected: convert units explicitly, then call <code>units::drop_units()</code> .
na	A character vector containing values that should be interpreted as missing. Matching happens after whitespace and full-width normalization.
strict	A single logical value. If FALSE, invalid values become <code>NA_real_</code> , a warning is emitted, and a data frame is stored in the <code>problems</code> attribute. If TRUE, invalid values cause an error.

Value

A data frame with double column `value` and character column `qualifier`. Qualifiers are `exact`, `approx`, `greater_than`, `at_least`, `less_than`, and `at_most`. Numeric finite and infinite values have the qualifier `exact`; NA and NaN have a missing qualifier. Numeric values, including Inf, -Inf, and NaN, are preserved in `value`.

Examples

```
parse_cn_quantity(c(
  "\u7ea663\u4e07", "\u5927\u4e8e2\u4ebf", "10\u4e07+", "50\u4f59", "50\u4f59\u4e07\u5143"
))
```

parse_cn_range	<i>Parse Numeric Ranges Used in Chinese Data</i>
----------------	--

Description

Converts closed ranges such as `"3\u4e07-5\u4e07"` and open bounds such as `"10\u4e07\u5143\u4ee5\u4e0a"` into explicit lower and upper bounds. A shared suffix in `"3-5\u4e07"` is applied to both endpoints. Signs used by negative numbers and scientific notation are distinguished from range separators. Digit grouping is validated independently at each range endpoint, so `"1 234-2 345"` is accepted but `"1 23-2 345"` is not. Endpoint missing markers are checked before and after shared-unit propagation.

Usage

```
parse_cn_range(
  x,
  na = c(
    "", "NA", "N/A", "\u6682\u65e0", "\u672a\u516c\u5e03",
    "\u2014", "\u2013", "...", "\u2026"
  )
)
```

```

    ),
    strict = FALSE
)

```

Arguments

<code>x</code>	A character, factor, or numeric vector. Numeric input is converted to double and otherwise passed through unchanged, including Inf, -Inf, and NaN. Conversion methods must return a plain double vector of the original length, without a class or dimensions; malformed results cause an error. SPSS user-defined missing metadata is respected. Objects with class units are rejected: convert units explicitly, then call <code>units::drop_units()</code> .
<code>na</code>	A character vector containing values that should be interpreted as missing. Matching happens after whitespace and full-width normalization.
<code>strict</code>	A single logical value. If FALSE, invalid values become <code>NA_real_</code> , a warning is emitted, and a data frame is stored in the <code>problems</code> attribute. If TRUE, invalid values cause an error.

Value

A data frame with columns `lower`, `upper`, `lower_inclusive`, and `upper_inclusive`. Numeric input becomes an inclusive point range, including numeric Inf and -Inf; NA and NaN have missing inclusivity. Infinite bounds created from textual inequalities represent open-ended bounds and are distinct from numeric point inputs.

Examples

```
parse_cn_range(c("3\u4e07-5\u4e07", "3-5\u4e07", "10\u4e07\u5143\u4ee5\u4e0a"))
```

Index

`cn_problems`, [2](#)

`cn_subset`, [2](#)

`cn_subset()`, [2](#), [3](#)

`cn_summary`, [3](#)

`parse_cn_number`, [4](#)

`parse_cn_quantity`, [5](#)

`parse_cn_range`, [6](#)