Package 'regextable'

May 9, 2026

Title Pattern-Based Text Extraction and Standardization with Lookup Tables

Version 0.1.1

Description Extracts information from text using lookup tables of regular expressions. Each text entry is compared against all patterns, and all matching patterns and their corresponding substrings are returned. If a text entry matches multiple patterns, multiple rows are generated to capture each match. This approach enables comprehensive pattern coverage when processing large or complex text datasets.

LazyData true

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports chk, dplyr, stringi, stringr, pbapply, stats

Suggests kableExtra, knitr, rmarkdown, spelling, testthat (>= 3.0.0)

VignetteBuilder knitr

URL https://github.com/judgelord/regextable, https://judgelord.github.io/regextable/

BugReports https://github.com/judgelord/regextable/issues

Config/testthat/edition 3

Depends R (>= 4.1)

Language en-US

NeedsCompilation no

Author Shirlyn Dong [aut, cre], Devin Judge-Lord [aut]

Maintainer Shirlyn Dong <shirlynd@umich.edu>

Repository CRAN

Date/Publication 2026-02-05 09:10:02 UTC

Contents

`clean_text` *Clean Text*

Description

Cleans a character vector by converting text to lowercase, removing selected punctuation (plus signs, em dashes, exclamation points), normalizing commas, and removing whitespace.

Usage

```
clean_text(text)
```

Arguments

`text`	Character vector to clean.

Value

Cleaned character vector.

Examples

```
clean_text(c("Hello  World!", "This is\tR"))
```

`cr2007_03_01` *cr2007_03_01 dataset*

Description

Sample text dataset used for demonstration of `regextable`.

Format

A tibble with 5 columns:

date Date of the record (YYYY-MM-DD)

speaker Speaker name in the text

header Header or title of the speech

url Original URL of the source text

url_txt Full text content from the source

Source

Generated for the `regextable` package.

`extract` *Extract pattern matches from text*

Description

Uses a regex lookup table to extract **all** pattern matches.

Usage

```
extract(
  data,
  col_name = "text",
  regex_table,
  pattern_col = "pattern",
  data_return_cols = NULL,
  regex_return_cols = NULL,
  date_col = NULL,
  date_start = NULL,
  date_end = NULL,
  remove_acronyms = FALSE,
  do_clean_text = TRUE,
  verbose = TRUE,
  cl = NULL
)
```

Arguments

`data`	A data frame or character vector containing the text to search.
`col_name`	Column name in data frame containing text to search through.
`regex_table`	A regex lookup table with a pattern column.
`pattern_col`	Name of the regex pattern column in regex_table.
`data_return_cols`	Optional vector of column names to include from 'data'.
`regex_return_cols`	Optional vector of column names to include from 'regex_table'.
`date_col`	Optional column in 'data' for date filtering.
`date_start`	Optional start date for filtering 'data'.
`date_end`	Optional end date for filtering 'data'.
`remove_acronyms`	Logical; if TRUE, removes all-uppercase patterns from regex_table.
`do_clean_text`	Logical; if TRUE, applies basic text cleaning to the input before matching.

verbose	Logical; if TRUE, displays progress messages.
cl	A cluster object created by parallel::makeCluster(), or an integer to indicate number of child-processes (integer values are ignored on Windows) for parallel evaluations. Passed to pbapply::pblapply().

Details

Pattern matching is performed using R's regular expression engine and is case-insensitive by default. For each input row, the function checks every pattern in regex_table and returns the first match of each pattern.

The output contains one row per pattern match per input row. If multiple patterns match the same text, multiple rows will be returned for that text.

Value

A tibble (data frame) with columns:

- row_id Integer row identifier corresponding to the input data
- Additional columns from data if data_return_cols specified
- Additional columns from regex_table if regex_return_cols specified
- pattern The matched regex pattern(s)
- match The specific text extracted from the data (original casing preserved)

Examples

```
# Create sample data
data <- data.frame(
  id = 1:3,
  text = c("I love apples", "Bananas are great", "Oranges and apples"),
  stringsAsFactors = FALSE
)

# Create regex patterns
patterns <- data.frame(
  pattern = c("apples", "bananas", "oranges"),
  category = c("fruit", "fruit", "fruit")
)

# Extract matches
extract(data, "text", patterns)
```

`members` *members dataset*

Description

Lookup table of member names and metadata for regex matching.

Format

A tibble with 9 columns:

congress Congress number (numeric)

chamber Chamber (House/President/Senate)

bioname Full bio name of the member

pattern Regex pattern to match this member's name

icpsr Numeric ICPSR identifier

state_abbrev Two-letter state abbreviation

district_code District number (0 for President)

first_name First name of the member

last_name Last name of the member

Source

Generated for the `regextable` package.

Index