Package: doseminer (via r-universe)

October 21, 2026

Type Package

Title Extract Drug Dosages from Free-Text Prescriptions

Version 0.2.1.9000

Description Utilities for converting unstructured electronic prescribing instructions into structured medication data. Extracts drug dose, units, daily dosing frequency and intervals from English-language prescriptions. Based on Karystianis et al. (2015) <doi:10.1186/s12911-016-0255-x>.

BugReports https://github.com/Selbosh/doseminer/issues

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports magrittr(>= 2.0.1), stringr(>= 1.4.0)

Suggests rmarkdown, knitr, testthat, tidyr, dplyr, ggplot2, prettydoc

Depends R (>= 3.5.0)

VignetteBuilder knitr

Language en-GB

Config/roxygen2/version 8.0.0

URL https://github.com/Selbosh/doseminer

Config/pak/sysreqs libicu-dev

Repository https://selbosh.r-universe.dev

Date/Publication 2026-07-23 08:00:24 UTC

RemoteUrl https://github.com/selbosh/doseminer

RemoteRef HEAD

RemoteSha d38076f6b3b9ba1ee8b26801c8dbd02493aea5c6

Contents

clean_prescription_text . . . 2
cprd . . . 3
drug_units . . . 3
example_cprd . . . 4
example_prescriptions . . . 4
extract_dose_unit . . . 4
extract_from_prescription . . . 5
hourly_to_daily . . . 6
latin_medical_terms . . . 6
multiply_dose . . . 7
numb_replacements . . . 7
regex_numbers . . . 8
replace_numbers . . . 9
weekly_to_daily . . . 9
words2number . . . 10

Index **11**

`clean_prescription_text` *Clean up raw prescription freetext*

Description

Clean up raw prescription freetext

Usage

```
clean_prescription_text(txt)
```

Arguments

`txt`	a character vector

Value

a character vector the same length as `txt`

Examples

```
clean_prescription_text(example_prescriptions)
```

cprd *Sample electronic prescribing dataset*

Description

A dataset containing product codes, patient identifiers, quantities, dates and free-text dose instructions, similar to data provided by the Clinical Practice Research Datalink (CPRD).

Usage

```
cprd
```

Format

An object of class `data.frame` with 714 rows and 6 columns.

Details

Variables in the data include

id record identifier

patid patient identifier

date date of start of prescription

prodcode product code; identifier for the prescribed medication

qty total quantity of medication prescribed

text free text prescribing instructions

drug_units *Medication dosage units*

Description

A named character vector. Names represent patterns to match dose units and values represent standardised names for those units.

Usage

```
drug_units
```

Details

Use with a function like `str_replace_all` to standardise a freetext prescription. Used internally in `extract_from_prescription`.

example_cprd *Example freetext prescriptions*

Description

Adapted from CPRD common dosages

Usage

```
example_cprd
```

See Also

example_prescriptions

example_prescriptions *Example freetext prescriptions*

Description

Various examples of how prescription data may be represented in free text.

Usage

```
example_prescriptions
```

See Also

example_cprd

extract_dose_unit *Extract units of dose from freetext prescriptions.*

Description

A function used internally in extract_from_prescription to parse the dosage units, such as millilitres, tablets, grams and so on. If there are multiple units mentioned in a string, only the first is returned.

Usage

```
extract_dose_unit(txt)
```

Arguments

txt	a character vector

Value

A character vector the same length as txt, containing standardised units, or NA if no units were found in the prescription.

A simple wrapper around str_replace_all and str_extract. Based on add_dose_unit.py from original Python/Java algorithm.

See Also

extract_from_prescription

---

extract_from_prescription

Extract dosage information from free-text English-language prescriptions

---

Description

This is the main workhorse function for the doseminer package. Pass in a character vector of prescribing instructions and it will extract structured dosage information.

Usage

```
extract_from_prescription(txt)
```

Arguments

txt	A character vector of freetext prescriptions

Details

To avoid redundant computation, it is recommended to remove duplicate elements from the input vector. The results can be joined back to the original data using the raw column.

Value

A data.frame with seven columns:

raw the input character vector

output a residual character vector of 'non-extracted' text. For debugging.

freq number of doses administered per day

itvl number of days between doses

dose quantity of medication in each dose

unit unit of measurement of medication, if any

optional integer. Can the dose be zero? 1 if yes, otherwise 0

Examples

```
extract_from_prescription(example_prescriptions)
```

---

hourly_to_daily *Convert hourly to daily frequency*

Description

Convert hourly to daily frequency

Usage

```
hourly_to_daily(txt)
```

Arguments

txt	String of the form 'every n hours'

Value

An equivalent string of the form 'x / day'

---

latin_medical_terms *List of Latin medical and pharmaceutical abbreviations*

Description

A named character vector. Names represent Latin terms and values the English translations. Used for converting terms like "q4h" into "every 4 hours", which can then be parsed into a dosage frequency/interval.

Usage

```
latin_medical_terms
```

Details

Use with a function like `str_replace_all` to translate a prescription from Latin to English (thence to numbers).

Source

https://en.wikipedia.org/wiki/List_of_abbreviations_used_in_medical_prescriptions

Examples

```
stringr::str_replace_all('Take two tablets q4h', latin_medical_terms)
```

multiply_dose *Evaluate a multiplicative plaintext expression*

Description

Replaces written phrases like "2 x 5" with their arithmetic result (i.e. 10)

Usage

```
multiply_dose(axb)
```

Arguments

axb	An string expression of the form 'A x B' where A, B are numeric

Value

An equivalent string giving the product of A and B. If A is a range of values, a range of values is returned.

See Also

Used internally within extract_from_prescription

numb_replacements *Dictionary of English names of numbers*

Description

For internal use in words2number. When passed as a replacement to a function like str_replace_all, it turns the string into an arithmetic expression that can be evaluated to give an integer representation of the named number.

Usage

```
numb_replacements
```

Details

Lifted from Ben Marwick's words2number package and converted into a named vector (previously a chain of gsub calls).

Note

Does not yet fully support decimals, fractions or mixed fractions. Some limited support for 'half' expressions, e.g. 'one and a half'.

Source

https://github.com/benmarwick/words2number

Examples

```
stringr::str_replace_all('one hundred and forty-two', doseminer:::numb_replacements)
```

`regex_numbers` *Regular expression to match numbers in English*

Description

A regex pattern to identify natural language English number phrases, such as "one hundred and fifty" or "thirty-seven". Used internally by `replace_numbers` to identify substrings to replace with their decimal representation.

Usage

```
regex_numbers
```

Details

This is a PCRE (Perl type) regular expression, so it must be used with `perl = TRUE` in base R regex functions. The packages `stringr` and `stringi` are based on the alternative ICU regular expression engine, so they cannot use this pattern.

Note

There is limited support for fractional expressions like "one half". The original pattern did not support expressions like "a thousand", but it has been adapted to offer (experimental) support for this. Phrases like "million" or "thousand" with no prefix will *not* match.

Source

https://www.rexegg.com/regex-trick-numbers-in-english.php

replace_numbers *Replace English number phrases with their decimal representations*

Description

Uses `numb_replacements` to match parts of a string corresponding to numbers, then invokes `words2number` to convert these substrings to numeric. The rest of the string (the non-number words) is left intact.

Usage

```
replace_numbers(string)
```

Arguments

`string`	A character vector. Can contain numbers and other text

Details

Works on non-negative integer numbers under one billion (one thousand million). Does not support fractions or decimals (yet).

Value

A character vector the same length as `string`, with words replaced by their decimal representations.

See Also

`words2number`, for use on cleaned text that does not contain any non-number words

Examples

```
replace_numbers('Two plus two equals four')
replace_numbers('one hundred thousand dollars!')
replace_numbers(c('A vector', 'containing numbers', 'like thirty seven'))
```

weekly_to_daily *Convert weekly interval to daily interval*

Description

Convert weekly interval to daily interval

Usage

```
weekly_to_daily(Dperweek)
```

Arguments

Dperweek	String of the form 'n / week'

Value

An equivalent string of the form 'x / day'

---

words2number *Convert English names of numbers to their numerical values*

Description

Convert English names of numbers to their numerical values

Usage

```
words2number(txt)
```

Arguments

txt	A character vector containing names of numbers (only).

Value

A named numeric vector of the same length as phrase.

Source

Originally adapted from the words2number package by Ben Marwick.

Examples

```
words2number('seven')
words2number('forty-two')
words2number(c('three', 'one', 'twenty two thousand'))
```

Index

∗ **datasets**
cprd, 3

clean_prescription_text, 2
cprd, 3

drug_units, 3

example_cprd, 4, *4*
example_prescriptions, 4, *4*
extract_dose_unit, 4
extract_from_prescription, *3*, *4*, 5, *5*, *7*

gsub, *7*

hourly_to_daily, 6

latin_medical_terms, 6

multiply_dose, 7

numb_replacements, 7, *9*

regex_numbers, 8
replace_numbers, *8*, 9

str_extract, *5*
str_replace_all, *3*, *5–7*

weekly_to_daily, 9
words2number, *7*, *9*, 10