Package 'parsedate'

July 23, 2025

Title Recognize and Parse Dates in Various Formats, Including All ISO 8601 Formats

Version 1.3.2

Maintainer Gábor Csárdi <csardi.gabor@gmail.com>

Description Parse dates automatically, without the need of specifying a format. Currently it includes the git date parser. It can also recognize and parse all ISO 8601 formats.

License GPL-2

URL https://github.com/gaborcsardi/parsedate

BugReports https://github.com/gaborcsardi/parsedate/issues

Suggests covr, testthat (>= 3.0.0), withr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation yes

Author Gábor Csárdi [aut, cre], Linus Torvalds [aut]

Repository CRAN

Date/Publication 2024-12-09 23:50:02 UTC

Contents

parsedate-package *Parse date from any format, including ISO 8601*

Description

Three useful functions to parse and format dates.

- `parse_iso_8601` recognizes and parses all valid ISO 8601 date and time formats. It can also be used as an ISO 8601 validator.
- `parse_date` can parse a date when you don't know which format it is in. First it tries all ISO 8601 formats. Then it tries git's versatile date parser. Lastly, it tries `as.POSIXct`.
- `format_iso_8601` formats a date (and time) in a specific ISO 8601 format.

See Also

Useful links:

- https://github.com/gaborcsardi/parsedate
- Report bugs at https://github.com/gaborcsardi/parsedate/issues

format_iso_8601 *Format date and time according to ISO 8601*

Description

Format a date in a fixed format that is ISO 8601 valid, and can be used to compare dates as character strings. It converts the date(s) to UTC.

Usage

```
format_iso_8601(date)
```

Arguments

`date`	The date(s) to format.

Value

Character vector of formatted dates.

Examples

```
format_iso_8601(parse_iso_8601("2013-02-08"))
format_iso_8601(parse_iso_8601("2013-02-08 09:34:00"))
format_iso_8601(parse_iso_8601("2013-02-08 09:34:00+01:00"))
format_iso_8601(parse_iso_8601("2013-W06-5"))
format_iso_8601(parse_iso_8601("2013-039"))
```

parse_date *Parse date from any format*

Description

Recognize and parse dates from a wide range of formats. The current algorithm is the following:

1. Try parsing dates using all valid ISO 8601 formats, by calling parse_iso_8601.
2. If this fails, then try parsing them using the git date parser.
3. If this fails, then try parsing them using as.POSIXct. (It is unlikely that this step will parse any dates that the first two steps couldn't, but it is still a logical fallback, to make sure that we can parse at least as many dates as as.POSIXct.

parse_date returns quickly in case of empty input elements.

Usage

```
parse_date(dates, approx = TRUE, default_tz = "UTC")
```

Arguments

dates	A character vector. An error is reported if the function cannot coerce this parameter to a character vector.
approx	Logical flag, whether the git parse should try hard(er). If this is set to TRUE, then the current time is used to fill in the missing parts of the date and time.
default_tz	Time zone to assume for dates that don't specify a time zone explicitly. Defaults to UTC, and an empty string means the local time zone.

Details

All dates are returned in the UTC time zone. If you prefer a different time zone, simply use '.POSIXct()' on the result, see examples below.

Value

A POSIXct vector. NA is returned for the dates that parse_date could not parse.

Examples

```
# Some easy examples
parse_date("2014-12-12")
parse_date("04/15/99")
parse_date("15/04/99")

# Ambiguous format, parsed assuming MM/DD/YY
parse_date("12/11/99")
parse_date("11/12/99")
```

```
# Fill in the current date and time
parse_date("03/20")
parse_date("12")

# But not for this, because this is ISO 8601
parse_date("2014")

# Handle vectors and empty input
parse_date(c("2014","2015","","2016"))

# Convert result to local time
tz <- format(Sys.time(), "%Z")
as.POSIXct(parse_date("2014-12-13T11:12:13"), tz)

# Local time zone
parse_date("2014-12-13T11:12:13", default_tz = "CET")
parse_date("2014-12-13T11:12:13", default_tz = "UTC")

# Convert results to different timezone
parse_date("2015-12-13T11:12:13")
.POSIXct(parse_date("2015-12-13T11:12:13"), tz = "CET")
```

`parse_iso_8601` *Parse date from an ISO 8601 format*

Description

See https://en.wikipedia.org/wiki/ISO_8601 and links therein for the complete standard.

Usage

```
parse_iso_8601(dates, default_tz = "UTC")
```

Arguments

`dates`	A character vector. An error is reported if the function cannot coerce this parameter to a character vector.
`default_tz`	Time zone to assume for dates that don't specify a time zone explicitly. Defaults to UTC, and an empty string means the local time zone.

Value

A `POSIXct` vector. `NA` is returned for the dates that `parse_date` could not parse.

Examples

```
# Missing fields
parse_iso_8601("2013-02-08 09")
parse_iso_8601("2013-02-08 09:30")

# Separator between date and time can be a 'T'
parse_iso_8601("2013-02-08T09")
parse_iso_8601("2013-02-08T09:30")
parse_iso_8601("2013-02-08T09:30:26")

# Fractional seconds, minutes, hours
parse_iso_8601("2013-02-08T09:30:26.123")
parse_iso_8601("2013-02-08T09:30.5")
parse_iso_8601("2013-02-08T09,25")

# Zulu time zone is UTC
parse_iso_8601("2013-02-08T09:30:26Z")

# ISO weeks, not very intuitive
parse_iso_8601("2013-W06-5")
parse_iso_8601("2013-W01-1")
parse_iso_8601("2009-W01-1")
parse_iso_8601("2009-W53-7")

# Day of the year
parse_iso_8601("2013-039")
parse_iso_8601("2013-039 09:30:26Z")
```

Index