# Package 'ymd'

November 26, 2024

**Title** Parse 'YMD' Format Number or String to Date

**Version** 0.1.4

**Maintainer** Xianying Tan <shrektan@126.com>

**Description** Convert 'YMD' format number or string to Date efficiently, using Rust's standard library. It also provides helper functions to handle Date, e.g., quick finding the beginning or end of the given period, adding months to Date, etc.

**License** MIT + file LICENSE

**URL** https://shrektan.github.io/ymd/, https://github.com/shrektan/ymd

**BugReports** https://github.com/shrektan/ymd/issues

**SystemRequirements** Cargo (Rust's package manager), rustc, xz

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Biarch** true

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Config/rextendr/version** 0.3.1

**NeedsCompilation** yes

**Author** Xianying Tan [aut, cre] (<https://orcid.org/0000-0002-6072-3521>),
Hiroaki Yutani [ctb] (<https://orcid.org/0000-0002-3385-7233>,
configure, configure.win, tools/configure.R),
The authors of the dependency Rust crates [ctb] (see inst/AUTHORS file for details)

**Repository** CRAN

**Date/Publication** 2024-11-26 16:00:02 UTC

## Contents

---

beop *Find the Beginning or End of Period*

---

## Description

Each of bop and eop contains a list of functions, whose names all consist of two letters, the first of which stands for **l**ast, **t**his, **n**ext while the second stands for **y**ear, **q**uarter, **m**onth, **w**eek. For example, `eop$ty()` means "the **e**nding **o**f **p**eriod of **t**his **y**ear" and `bop$lm()` means "the **b**eginning **o**f **p**eriod of **l**ast **m**onth".

## Details

All functions' signatures are the same, with only one argument `x`, which could be a `Date` or values that can be converted to `Date` via `ymd()`.

## Examples

```
bop$ty(as.Date("2021-03-02"))
## supports 'YMD' formatted integer or string
bop$ty(210302)
eop$tm(200201)
```

---

date_part *Fast Date Part Extracting*

---

## Description

These date helper functions provide the similar functionalities like in `data.table` or `lubridate` package. They are implemented by the Rust Lang's standard library and very fast.

## Usage

```
year(ref_date)

month(ref_date)

quarter(ref_date)

isoweek(ref_date)

isowday(ref_date)

wday(ref_date)
```

```
mday(ref_date)

yday(ref_date)
```

## Arguments

| | |
|---|---|
| ref_date | a Date vector. It will try to convert the input to date via ymd(), if the input is not a Date. |

## Details

- year, month, quarter: get the year, month, quarter part
- yday: the day of year
- mday: the day of month
- wday: the day of the week (Sunday is 1)
- isoweek: ISO 8601 week
- isowday: the day of week (ISO 8601 weekday number, Monday is 1)

## Value

an integer vector

## References

ISO week day, https://en.wikipedia.org/wiki/ISO_week_date ISO 8601, https://en.wikipedia.org/wiki/ISO_8601

## Examples

```
year(210205)
month(210205)
quarter(210205)
yday(210205)
mday(210205)
wday(210117)
isowday(210117)
isoweek(210101)
```

---

# edate *Calculate the date before / after months*

---

## Description

Calculate the date before / after months

## Usage

```
edate(ref_date, months)
```

### Arguments

| | |
|---|---|
| ref_date | a Date vector |
| months | the number of months that's added to ref_date |

### Note

The function name is the same as the Excel function EDATE() and does the same. It returns the date that is the indicated number of months before or after the ref date.

### Examples

```
edate(as.Date("2020-01-31"), 1)
## supports 'YMD' formatted integer or string
edate(200131, 1)
edate(200229, -12)
```

---

## ymd *Convert 'YMD' format integer or string to Date*

---

### Description

Transform integer or strings vectors in 'YMD' format to Date objects. It intends to only support limited formats (no separator or one of '.', ' ', '-' and '/' separators). See the possible formats in examples.

### Usage

```
ymd(x, ...)
```

### Arguments

| | |
|---|---|
| x | An integer or string vector in 'YMD' format. Double values without the decimal part are allowed. |
| ... | The same as x. It will be merged into one vector with x. It's convinient for interactive use. |

### Value

A Date object. When the parse fails for certain input, the value returned would be NA, silently.

### Examples

```
ymd(c(210326, 19981225))
ymd(c("2020/1/8", "20 1 7", "1998.7.1", "1990-02-03"))
ymd(210420, 180322)
```

# Index