Package 'date'

February 18, 2026

Version 1.2-43

Title Functions for Handling Dates

Description Functions for handling dates.

Imports graphics

License GPL-2

NeedsCompilation yes

Author Terry Therneau [aut] (S original),
Thomas Lumley [trl] (R port),
Kjetil Halvorsen [trl] (R port),
Kurt Hornik [trl, aut, cre] (ORCID:
<https://orcid.org/0000-0003-4198-9911>, R port),
R Core Team [ctb]

Maintainer Kurt Hornik <Kurt.Hornik@R-project.org>

Repository CRAN

Date/Publication 2026-02-18 08:46:22 UTC

Contents

`as.date` *Coerce Data to Dates*

Description

Converts any of the following character forms to a Julian date: 8/31/56, 8-31-1956, 31 8 56, 083156, 31Aug56, or August 31 1956.

Usage

```
as.date(x, order = "mdy", ...)
```

Arguments

`x`	input data vector.
`order`	if `x` is character, defines the order in which the terms are assumed to appear in a xx/xx/xx date. The default is month/day/year; any permutation of mdy is legal.
`...`	if `x` is character, then any other arguments from `mdy.date()` can be used as well.

Details

If `x` is numeric, then `floor(x)` is returned, e.g., `as.date(35)` is the same as `as.date(35.2)` and gives February 5, 1960 ('`5Feb60`'). If `x` is character, the program attempts to parse it.

Value

For each date, the number of days between it and January 1, 1960. The date will be missing if the string is not interpretable.

See Also

`mdy.date`, `date.mmddyy`, `date.ddmmmyy`

Examples

```
as.date(c("1jan1960", "2jan1960", "31mar1960", "30jul1960"))
```

date.ddmmmyy *Format a Julian date*

Description

Given a vector of Julian dates, this returns them in the form "10Nov89", "28Jul54", etc.

Usage

```
date.ddmmmyy(sdate)
```

Arguments

sdate	A vector of Julian dates, e.g., as returned by `mdy.date()`.

Value

A vector of character strings containing the formatted dates.

See Also

`mdy.date`, `date.mdy`

Examples

```
date.ddmmmyy(1:10)
```

date.mdy *Convert from Julian Dates to Month, Day, and Year*

Description

Convert a vector of Julian dates to a list of vectors with the corresponding values of month, day and year, and optionally weekday.

Usage

```
date.mdy(sdate, weekday = FALSE)
```

Arguments

sdate	a Julian date value, as returned by `mdy.date()`, number of days since 1/1/1960.
weekday	if `TRUE`, then the returned list also will contain the day of the week (Sunday=1, Saturday=7).

Value

A list with components `month`, `day`, and `year`.

References

Press WH, Teukolsky SA, Vetterling WT, Flannery BP (1992). *Numerical Recipes in C: The Art of Scientific Computing*, Second edition. Cambridge University Press. ISBN 978-0521431088.

Examples

```
day <- 7
temp <- date.mdy(mdy.date(month = 7, day = day, year = 1960))
## Check for illegal dates, such as 29 Feb in a non leap year
if (temp$day != day) {
  cat("Some illegal dates\n")
} else {
  cat("All days are legal\n")
}
```

`date.mmddyy` *Format a Julian date*

Description

Given a vector of Julian dates, this returns them in the form "10/11/89", "28/7/54", etc.

Usage

```
date.mmddyy(sdate, sep = "/")
```

Arguments

`sdate`	A vector of Julian dates, e.g., as returned by `mdy.date()`.
`sep`	Character string used to separate the month, day, and year portions of the returned string.

Value

A vector of character strings containing the formatted dates.

See Also

`date.mdy`, `mdy.date`, `date.ddmmmyy`

Examples

```
date.mmddyy(as.date(10))
```

date.mmddyyyy *Format a Julian date*

Description

Given a vector of Julian dates, this returns them in the form "10/11/1989", "28/7/1854", etc.

Usage

```
date.mmddyyyy(sdate, sep = "/")
```

Arguments

sdate	A vector of Julian dates, e.g., as returned by `mdy.date()`.
sep	Character string used to separate the month, day, and year portions of the returned string.

Value

A vector of character strings containing the formatted dates.

See Also

`date.mdy`, `mdy.date`, `date.ddmmmyy`

Examples

```
date.mmddyyyy(as.date(1:10))
```

date.object *Date Objects*

Description

Objects of class `"date"`.

Usage

```
is.date(x)
```

Arguments

x	any R object.

Details

Dates are stored as the number of days since 1/1/1960, and are kept in integer format. (This is the same baseline value as is used by SAS). The numerical methods for dates treat `date - date` as a numeric, and `date +- numeric` as a date.

`is.date` returns `TRUE` if `x` has class `"date"`, and `FALSE` otherwise. Its behavior is unaffected by any attributes of `x`; for example, `x` could be a date array (in contrast to the behavior of `is.vector`).

`as.date` returns `x` if `x` is a simple object of class `"date"`, and otherwise a date vector of the same length as `x` and with data resulting from coercing the elements of `x` to class `"date"`. See the manual page for `as.date()` for details.

Logical operations as well as the numeric functions `exp()`, `log()`, and so on are invalid.

Other methods exist for missing value, `as.character()`, printing, and summarizing.

See Also

`date.mdy`, `mdy.date`, `date.ddmmmyy`, `as.date`.

---

mdy.date *Convert to Julian Dates*

Description

Given a month, day, and year, returns the number of days since January 1, 1960.

Usage

```
mdy.date(month, day, year, nineteen = TRUE, fillday = FALSE,
         fillmonth = FALSE)
```

Arguments

`month`	vector of months.
`day`	vector of days.
`year`	vector of years.
`nineteen`	if `TRUE`, year values between 0 and 99 are assumed to be in the 20th century A.D.; otherwise, if `FALSE`, they are assumed to be in the 1st century A.D.
`fillday`	if `TRUE`, then missing days are replaced with 15.
`fillmonth`	if `TRUE`, then a missing month causes the month and day to be set to 7/1.

Details

The date functions are particularly useful in computing time spans, such as number of days on test, and similar functions can be found in other statistical packages. The baseline date of Jan 1, 1960 is, of course, completely arbitrary (it is the same one used by SAS).

The `fillday` and `fillmonth` options are perhaps useful only to the author and a very few others: we sometimes deal with patients whose birth date was in the 1800's, and only the month or even only the year is known. When the interval is greater than 80 years, a filler seems defensible.

Value

a vector of Julian dates.

References

Press WH, Teukolsky SA, Vetterling WT, Flannery BP (1992). *Numerical Recipes in C: The Art of Scientific Computing*, Second edition. Cambridge University Press. ISBN 978-0521431088.

See Also

`date.mmddyy`, `date.ddmmmyy`, `date.mmddyyyy`

Examples

```
mdy.date(3, 10, 53)
xzt <-1:10
xzy <- as.date(xzt)
test <- data.frame(x = xzt, date = xzy)
summary(test)
```

Index