

# Package ‘cstime’

July 22, 2025

**Title** Date and Time Functions for Public Health Purposes

**Version** 2023.5.3

**Description** Provides easy and consistent time conversion for public health purposes. The time conversion functions provided here are between date, ISO week, ISO yearweek, ISO year, calendar month/year, season, season week.

**License** MIT + file LICENSE

**URL** <https://www.csids.no/cstime/>, <https://github.com/csids/cstime>

**BugReports** <https://github.com/csids/cstime/issues>

**Encoding** UTF-8

**LazyData** true

**LazyDataCompression** xz

**RoxygenNote** 7.2.3

**Depends** R (>= 3.5.0)

**Imports** data.table, magrittr

**Suggests** rmarkdown, ggplot2, knitr, testthat (>= 3.0.0), readxl, glue, lubridate

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Chi Zhang [aut, cre] (ORCID: <<https://orcid.org/0000-0003-0501-5909>>),  
Richard Aubrey White [aut] (ORCID:  
<<https://orcid.org/0000-0002-6747-1726>>)

**Maintainer** Chi Zhang <andreachizhang@yahoo.com>

**Repository** CRAN

**Date/Publication** 2023-05-15 14:30:02 UTC

Contents

dates_by_isoyearweek . . . . .	2
date_to_calmonth_c . . . . .	3
date_to_calmonth_n . . . . .	4
date_to_calyearmonth_c . . . . .	5
date_to_calyear_c . . . . .	5
date_to_calyear_n . . . . .	6
date_to_isoweek_c . . . . .	7
date_to_isoweek_n . . . . .	8
date_to_isoyearweek_c . . . . .	8
date_to_isoyear_c . . . . .	9
date_to_isoyear_n . . . . .	10
date_to_seasonweek_n . . . . .	11
date_to_season_c . . . . .	11
isoweek_to_seasonweek_n . . . . .	12
isoyearweek_to_isoweek_c . . . . .	12
isoyearweek_to_isoweek_n . . . . .	13
isoyearweek_to_isoyear_c . . . . .	14
isoyearweek_to_isoyear_n . . . . .	14
isoyearweek_to_last_date . . . . .	15
isoyearweek_to_seasonweek_n . . . . .	16
isoyearweek_to_season_c . . . . .	16
isoyear_to_last_date . . . . .	17
isoyear_to_last_isoweek_n . . . . .	18
isoyear_to_last_isoyearweek_c . . . . .	18
keep_sundays . . . . .	19
keep_sundays_and_latest_date . . . . .	20
keep_sundays_and_latest_date_internal . . . . .	20
nor_workdays_by_date . . . . .	21
nor_workdays_by_isoyearweek . . . . .	21
now_c . . . . .	22
seasonweek_to_isoweek_c . . . . .	22
seasonweek_to_isoweek_n . . . . .	23
<b>Index</b>	<b>24</b>

---

dates_by_isoyearweek	<i>Dates of different days within isoyearweeks</i>
----------------------	--

---

Description

Dates of different days within isoyearweeks

Usage

dates\_by\_isoyearweek

**Format**

**isoyear** Isoyear.  
**isoyearweek** Isoweek-isoyear.  
**mon** Date of Monday.  
**tue** Date of Tuesday.  
**wed** Date of Wednesday.  
**thu** Date of Thursday.  
**fri** Date of Friday.  
**sat** Date of Saturday.  
**sun** Date of Sunday.  
**weekdays** List of dates from Mon-Fri  
**weekend** List of dates from Sat-Sun  
**days** List of dates from Mon-Sun

**Examples**

```
# Constructing a vector of dates without removing the Date class
do.call("c", dates_by_isoyearweek[isoyearweek %in% c("2021-01", "2021-02")]]$weekdays)
```

---

date_to_calmonth_c	<i>Date -&gt; calmonth (character)</i>
--------------------	--

---

**Description**

Date -> calmonth (character)

**Usage**

```
date_to_calmonth_c(x = lubridate::today())

## Default S3 method:
date_to_calmonth_c(x)

## S3 method for class 'character'
date_to_calmonth_c(x = lubridate::today())

## S3 method for class 'Date'
date_to_calmonth_c(x = lubridate::today())
```

**Arguments**

x                      a Date object or string, in the form of 'yyyy-mm-dd'

**Value**

calmonth ("XX")

**Examples**

```
date_to_calmonth_c("2021-08-11")
date_to_calmonth_c(lubridate::today())
```

---

date\_to\_calmonth\_n      *Date -> calmonth (numeric)*

---

**Description**

Date -> calmonth (numeric)

**Usage**

```
date_to_calmonth_n(x = lubridate::today())

## Default S3 method:
date_to_calmonth_n(x)

## S3 method for class 'character'
date_to_calmonth_n(x = lubridate::today())

## S3 method for class 'Date'
date_to_calmonth_n(x = lubridate::today())
```

**Arguments**

x                      a Date object or string, in the form of 'yyyy-mm-dd'

**Value**

calmonth

**Examples**

```
date_to_calmonth_n("2021-08-11")
date_to_calmonth_n(lubridate::today())
```

---

```
date_to_cyearmonth_c  
Date -> cyearmonth (character)
```

---

**Description**

Date -> cyearmonth (character)

**Usage**

```
date_to_cyearmonth_c(x = lubridate::today())  
  
## Default S3 method:  
date_to_cyearmonth_c(x)  
  
## S3 method for class 'character'  
date_to_cyearmonth_c(x = lubridate::today())  
  
## S3 method for class 'Date'  
date_to_cyearmonth_c(x = lubridate::today())
```

**Arguments**

x                      a Date object or string, in the form of 'yyyy-mm-dd'

**Value**

cyearmonth ("YYYY-MXX")

**Examples**

```
date_to_cyearmonth_c("2021-08-11")  
date_to_cyearmonth_c(lubridate::today())
```

---

```
date_to_cyear_c        Date -> cyear (character)
```

---

**Description**

Date -> cyear (character)

**Usage**

```
date_to_calyear_c(x = lubridate::today())

## Default S3 method:
date_to_calyear_c(x)

## S3 method for class 'character'
date_to_calyear_c(x = lubridate::today())

## S3 method for class 'Date'
date_to_calyear_c(x = lubridate::today())
```

**Arguments**

x                      a Date object or string, in the form of 'yyyy-mm-dd'

**Value**

ISO year in character

**Examples**

```
date_to_calyear_c("2021-08-11")
date_to_calyear_c(lubridate::today())
```

---

date\_to\_calyear\_n      *Date -> calyear (numeric)*

---

**Description**

Date -> calyear (numeric)

**Usage**

```
date_to_calyear_n(x = lubridate::today())

## Default S3 method:
date_to_calyear_n(x)

## S3 method for class 'character'
date_to_calyear_n(x = lubridate::today())

## S3 method for class 'Date'
date_to_calyear_n(x = lubridate::today())
```

**Arguments**

x                      a Date object or string, in the form of 'yyyy-mm-dd'

**Value**

ISO year in character

**Examples**

```
date_to_calyear_n("2021-08-11")
date_to_calyear_n(lubridate::today())
```

---

date_to_isoweek_c	<i>ISO week (character) from Date object</i>
-------------------	--

---

**Description**

ISO week (character) from Date object

**Usage**

```
date_to_isoweek_c(x = lubridate::today())

## Default S3 method:
date_to_isoweek_c(x)

## S3 method for class 'character'
date_to_isoweek_c(x = lubridate::today())

## S3 method for class 'Date'
date_to_isoweek_c(x = lubridate::today())
```

**Arguments**

x                      a Date object or string, in the form of 'yyyy-mm-dd'

**Value**

ISO week in character

**Examples**

```
date_to_isoweek_c("2021-08-11")
date_to_isoweek_c(lubridate::today())
```

---

date\_to\_isoweek\_n      *ISO week (numeric) from Date object*

---

### Description

ISO week (numeric) from Date object

### Usage

```
date_to_isoweek_n(x = lubridate::today())

## Default S3 method:
date_to_isoweek_n(x)

## S3 method for class 'character'
date_to_isoweek_n(x = lubridate::today())

## S3 method for class 'Date'
date_to_isoweek_n(x = lubridate::today())
```

### Arguments

x                      a Date object or string, in the form of 'yyyy-mm-dd'

### Value

ISO week in numeric

### Examples

```
date_to_isoweek_n("2021-08-11")
date_to_isoweek_n(lubridate::today())
```

---

date\_to\_isoyearweek\_c    *ISO year and week (character) from Date object*

---

### Description

ISO year and week (character) from Date object



**Usage**

```
date_to_isoyearweek_c(x = lubridate::today())

## Default S3 method:
date_to_isoyearweek_c(x)

## S3 method for class 'character'
date_to_isoyearweek_c(x = lubridate::today())

## S3 method for class 'Date'
date_to_isoyearweek_c(x = lubridate::today())
```

**Arguments**

x                      a Date object or string, in the form of 'yyyy-mm-dd'

**Value**

ISO year and week in character

**Examples**

```
date_to_isoyearweek_c("2021-08-11")
date_to_isoyearweek_c(lubridate::today())
```

---

date_to_isoyear_c	<i>ISO year (character) from Date object</i>
-------------------	--

---

**Description**

ISO year (character) from Date object

**Usage**

```
date_to_isoyear_c(x = lubridate::today())

## Default S3 method:
date_to_isoyear_c(x)

## S3 method for class 'character'
date_to_isoyear_c(x = lubridate::today())

## S3 method for class 'Date'
date_to_isoyear_c(x = lubridate::today())
```

**Arguments**

x                      a Date object or string, in the form of 'yyyy-mm-dd'

**Value**

ISO year in character

**Examples**

```
date_to_isoyear_c("2021-08-11")
date_to_isoyear_c(lubridate::today())
```

---

date_to_isoyear_n	<i>ISO year (numeric) from Date object</i>
-------------------	--

---

**Description**

ISO year (numeric) from Date object

**Usage**

```
date_to_isoyear_n(x = lubridate::today())

## Default S3 method:
date_to_isoyear_n(x)

## S3 method for class 'character'
date_to_isoyear_n(x = lubridate::today())

## S3 method for class 'Date'
date_to_isoyear_n(x = lubridate::today())
```

**Arguments**

x                      a Date object or string, in the form of 'yyyy-mm-dd'

**Value**

ISO year in numeric

**Examples**

```
date_to_isoyear_n("2021-08-11")
date_to_isoyear_n(lubridate::today())
```

---

date\_to\_seasonweek\_n    *Date to season week.*

---

**Description**

Date to season week.

**Usage**

```
date_to_seasonweek_n(x)
```

**Arguments**

x	date
---	------

**Value**

Season week in numeric

**Examples**

```
date_to_seasonweek_n(c("2021-01-01", "2021-12-01"))
```

---

date\_to\_season\_c    *Date to season.*

---

**Description**

Date to season.

**Usage**

```
date_to_season_c(x)
```

**Arguments**

x	date
---	------

**Value**

Season, e.g. 2020/2021

**Examples**

```
date_to_season_c(c("2021-01-01", "2021-12-01"))
```

---

```
isoweek_to_seasonweek_n
```

*ISO week to season week (numeric). Season week 1 is natural week 30.*

---

### Description

ISO week to season week (numeric). Season week 1 is natural week 30.

### Usage

```
isoweek_to_seasonweek_n(x)
```

```
## Default S3 method:
```

```
isoweek_to_seasonweek_n(x)
```

```
## S3 method for class 'character'
```

```
isoweek_to_seasonweek_n(x)
```

```
## S3 method for class 'numeric'
```

```
isoweek_to_seasonweek_n(x)
```

### Arguments

x	ISO week in a year (numeric), between 1 and 53. ISO week 53 is season week 23.5
---	---

### Value

Season week in numeric

### Examples

```
isoweek_to_seasonweek_n(31)
```

---

```
isoyearweek_to_isoweek_c
```

*ISO yearweek to week (character)*

---

### Description

This function breaks the string connected with '-' into year/week

**Usage**

```
isoyearweek_to_isoweek_c(x)

## Default S3 method:
isoyearweek_to_isoweek_c(x)

## S3 method for class 'character'
isoyearweek_to_isoweek_c(x)
```

**Arguments**

x                      Year-week, e.g. "2020-19" for 19th week in 2020

**Value**

ISO week in character

**Examples**

```
isoyearweek_to_isoweek_c('2020-19')
```

---

```
isoyearweek_to_isoweek_n
      ISO yearweek to week (numeric)
```

---

**Description**

This function breaks the string connected with '-' into year/week

**Usage**

```
isoyearweek_to_isoweek_n(x)

## Default S3 method:
isoyearweek_to_isoweek_n(x)

## S3 method for class 'character'
isoyearweek_to_isoweek_n(x)
```

**Arguments**

x                      Year-week, e.g. "2020-19" for 19th week in 2020

**Value**

ISO week in numeric

**Examples**

```
isoyearweek_to_isoweek_n('2020-19')
```

---

```
isoyearweek_to_isoyear_c
```

*ISO yearweek to year (character)*

---

**Description**

This function breaks the string connected with '-' into year/week

**Usage**

```
isoyearweek_to_isoyear_c(x)
```

```
## Default S3 method:
```

```
isoyearweek_to_isoyear_c(x)
```

```
## S3 method for class 'character'
```

```
isoyearweek_to_isoyear_c(x)
```

**Arguments**

x                      Year-week, e.g. "2020-19" for 19th week in 2020

**Value**

ISO year in character

**Examples**

```
isoyearweek_to_isoyear_c('2020-10')
```

---

```
isoyearweek_to_isoyear_n
```

*ISO yearweek to year (numeric)*

---

**Description**

This function breaks the string connected with '-' into year/week

**Usage**

```
isoyearweek_to_isoyear_n(x)

## Default S3 method:
isoyearweek_to_isoyear_n(x)

## S3 method for class 'character'
isoyearweek_to_isoyear_n(x)
```

**Arguments**

x                      Year-week, e.g. "2020-19" for 19th week in 2020

**Value**

ISO year in numeric

**Examples**

```
isoyearweek_to_isoyear_n('2020-10')
```

---

isoyearweek\_to\_last\_date

*Last date in ISO yearweek*

---

**Description**

Returns the Sunday in the isoyearweek

**Usage**

```
isoyearweek_to_last_date(x)

## Default S3 method:
isoyearweek_to_last_date(x)

## S3 method for class 'character'
isoyearweek_to_last_date(x)
```

**Arguments**

x                      ISO yearweek, e.g. "2020-19" for 19th week in 2020

**Value**

Date of Sunday of that isoyearweek

**Examples**

```
isoyearweek_to_last_date(c("2019-19", "2020-01"))
```

---

```
isoyearweek_to_seasonweek_n
```

*ISO yearweek to season week (numeric). Season week 1 is natural week 30.*

---

**Description**

ISO yearweek to season week (numeric). Season week 1 is natural week 30.

**Usage**

```
isoyearweek_to_seasonweek_n(x)
```

**Arguments**

x                      ISO yearweek

**Value**

Season week in numeric

**Examples**

```
isoyearweek_to_seasonweek_n(c("2021-01"))
```

---

```
isoyearweek_to_season_c
```

*ISO yearweek to season.*

---

**Description**

ISO yearweek to season.

**Usage**

```
isoyearweek_to_season_c(x)
```

```
## Default S3 method:
```

```
isoyearweek_to_season_c(x)
```

```
## S3 method for class 'character'
```

```
isoyearweek_to_season_c(x)
```



**Arguments**

x isoyearweek, connected with '-'

**Value**

Season, e.g. 2020/2021

**Examples**

```
isoyearweek_to_season_c(c("2021-01", "2021-50"))
```

---

isoyear_to_last_date	<i>Last Sunday in ISO year</i>
----------------------	--------------------------------

---

**Description**

Returns the last Sunday in the isoyear

**Usage**

```
isoyear_to_last_date(x)

## Default S3 method:
isoyear_to_last_date(x)

## S3 method for class 'character'
isoyear_to_last_date(x)

## S3 method for class 'numeric'
isoyear_to_last_date(x)
```

**Arguments**

x ISO year, e.g. 2020

**Value**

Date of the Sunday, for the last week in the isoyear

**Examples**

```
isoyear_to_last_date(c(2019, 2019, 2020, 2021))
```

---

isoyear\_to\_last\_isoweek\_n

*Last ISO week (numeric) in ISO year*


---

### Description

Returns the last week in the isoyear

### Usage

```
isoyear_to_last_isoweek_n(x)
```

```
## Default S3 method:
```

```
isoyear_to_last_isoweek_n(x)
```

```
## S3 method for class 'character'
```

```
isoyear_to_last_isoweek_n(x)
```

```
## S3 method for class 'numeric'
```

```
isoyear_to_last_isoweek_n(x)
```

### Arguments

x                      ISO year, e.g. 2020

### Value

ISO week in numeric

### Examples

```
isoyear_to_last_isoweek_n(c(2019, 2019, 2020, 2021))
```

---

isoyear\_to\_last\_isoyearweek\_c

*Last ISO yearweek (character) in ISO year*


---

### Description

Returns the last isoyearweek in the isoyear

**Usage**

```
isoyear_to_last_isoyearweek_c(x)

## Default S3 method:
isoyear_to_last_isoyearweek_c(x)

## S3 method for class 'character'
isoyear_to_last_isoyearweek_c(x)

## S3 method for class 'numeric'
isoyear_to_last_isoyearweek_c(x)
```

**Arguments**

x                      ISO year, e.g. 2020

**Value**

ISO year-week in character, of the last ISO year

**Examples**

```
isoyear_to_last_isoyearweek_c(c(2019, 2019, 2020, 2021))
```

---

keep_sundays	<i>keep_sundays</i>
--------------	---------------------

---

**Description**

keep\_sundays

**Usage**

```
keep_sundays(dates, format = "Uke isoweek_c-1/isoweek_c", keep_delete = TRUE)
```

**Arguments**

dates	dates (later)
format	format (later)
keep_delete	keep_delete (later)

**Value**

Label: delete or not

---

```
keep_sundays_and_latest_date
    keep_sundays_and_latest_date
```

---

### Description

keep\_sundays\_and\_latest\_date

### Usage

```
keep_sundays_and_latest_date(
  dates,
  format = "Uke isoweek_c-1/isoweek_c",
  keep_delete = TRUE
)
```

### Arguments

dates	dates (later)
format	format (later)
keep_delete	keep_delete (later)

### Value

Date with level: delete or not

---

```
keep_sundays_and_latest_date_internal
    keep_sundays_and_latest_date_internal
```

---

### Description

keep\_sundays\_and\_latest\_date\_internal

### Usage

```
keep_sundays_and_latest_date_internal(
  dates,
  format = "Uke isoweek_c-1/isoweek_c",
  keep_delete = TRUE,
  keep_latest_date = TRUE
)
```

**Arguments**

dates	dates (later)
format	format (later)
keep_delete	keep_delet (later)
keep_latest_date	keep_latest_date (later)

---

nor\_workdays\_by\_date    *Norwegian workdays and holidays by date*

---

**Description**

Norwegian workdays and holidays by date

**Usage**

nor\_workdays\_by\_date

**Format**

**date** Date.

**day\_of\_week** Integer. 1 = Monday, 7 = Sunday

**mon\_to\_fri** Integer. 1 between Monday and Friday, 0 between Saturday and Sunday

**sat\_to\_sun** Integer. 1 between Saturday and Sunday, 0 between Monday and Friday

**public\_holiday** Integer. 1 if public holiday (helligdag), 0 if not public holiday

**freeday** Integer. 1 if public holiday (helligdag) or sat\_to\_sun==1, 0 otherwise

**workday** Integer. 1 if freeday==0, 0 if freeday==1

---

nor\_workdays\_by\_isoyearweek  
*Norwegian workdays and holidays by isoyearweek*

---

**Description**

Norwegian workdays and holidays by isoyearweek

**Usage**

nor\_workdays\_by\_isoyearweek

**Format**

- isoyearweek** YYYY-WW
- public\_holiday** The proportion of the days within the isoyearweek that are public holidays
- freeday** The proportion of the days within the isoyearweek that are either public holidays or Saturday/Sunday
- workday** 1 minus freeday

---

now_c	<i>Current time as character</i>
-------	----------------------------------

---

**Description**

Current time as character

**Usage**

```
now_c(format = "%Y-%m-%d %H:%M:%S")
```

**Arguments**

- format Defaults to %Y-%m-%d %H:%M:%S

---

seasonweek_to_isoweek_c	<i>Season week to ISO week (character). Season week 1 is ISO week 30.</i>
-------------------------	---

---

**Description**

Season week to ISO week (character). Season week 1 is ISO week 30.

**Usage**

```
seasonweek_to_isoweek_c(x)

## Default S3 method:
seasonweek_to_isoweek_c(x)

## S3 method for class 'numeric'
seasonweek_to_isoweek_c(x)
```

**Arguments**

- x Season week in a year (numeric), between 1 and 52

**Value**

ISO week in character

**Examples**

```
seasonweek_to_isoweek_c(31)
```

---

```
seasonweek_to_isoweek_n
```

*Season week to ISO week (numeric). Season week 1 is ISO week 30.*

---

**Description**

Season week to ISO week (numeric). Season week 1 is ISO week 30.

**Usage**

```
seasonweek_to_isoweek_n(x)
```

```
## Default S3 method:
```

```
seasonweek_to_isoweek_n(x)
```

```
## S3 method for class 'numeric'
```

```
seasonweek_to_isoweek_n(x)
```

**Arguments**

x                      Season week in a year, between 1 and 52

**Value**

ISO week in numeric

**Examples**

```
seasonweek_to_isoweek_n(31)
```

# Index

## \* datasets

dates\_by\_isoyearweek, [2](#)  
nor\_workdays\_by\_date, [21](#)  
nor\_workdays\_by\_isoyearweek, [21](#)

date\_to\_calmonth\_c, [3](#)  
date\_to\_calmonth\_n, [4](#)  
date\_to\_calyear\_c, [5](#)  
date\_to\_calyear\_n, [6](#)  
date\_to\_calyearmonth\_c, [5](#)  
date\_to\_isoweek\_c, [7](#)  
date\_to\_isoweek\_n, [8](#)  
date\_to\_isoyear\_c, [9](#)  
date\_to\_isoyear\_n, [10](#)  
date\_to\_isoyearweek\_c, [8](#)  
date\_to\_season\_c, [11](#)  
date\_to\_seasonweek\_n, [11](#)  
dates\_by\_isoyearweek, [2](#)

isoweek\_to\_seasonweek\_n, [12](#)  
isoyear\_to\_last\_date, [17](#)  
isoyear\_to\_last\_isoweek\_n, [18](#)  
isoyear\_to\_last\_isoyearweek\_c, [18](#)  
isoyearweek\_to\_isoweek\_c, [12](#)  
isoyearweek\_to\_isoweek\_n, [13](#)  
isoyearweek\_to\_isoyear\_c, [14](#)  
isoyearweek\_to\_isoyear\_n, [14](#)  
isoyearweek\_to\_last\_date, [15](#)  
isoyearweek\_to\_season\_c, [16](#)  
isoyearweek\_to\_seasonweek\_n, [16](#)

keep\_sundays, [19](#)  
keep\_sundays\_and\_latest\_date, [20](#)  
keep\_sundays\_and\_latest\_date\_internal,  
[20](#)

nor\_workdays\_by\_date, [21](#)  
nor\_workdays\_by\_isoyearweek, [21](#)  
now\_c, [22](#)

seasonweek\_to\_isoweek\_c, [22](#)

seasonweek\_to\_isoweek\_n, [23](#)