Package 'grates'

October 21, 2021

Title Grouped Date Classes

Version 0.3.0

Description Provides a coherent interface and implementation for creating grouped date classes. This package is part of the RECON (<https://www.repidemicsconsortium.org/>) toolkit for outbreak analysis.

URL https://www.reconverse.org/grates/, https://github.com/reconverse/grates

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.2

Imports clock, ellipsis, rlang, vctrs

Suggests knitr, ggplot2, scales, testthat (>= 3.0.0), rmarkdown, dplyr, outbreaks

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Tim Taylor [aut, cre] (<https://orcid.org/0000-0002-8587-7113>)

Maintainer Tim Taylor <tim.taylor@hiddenelephants.co.uk>

Repository CRAN

Date/Publication 2021-10-21 21:10:02 UTC

R topics documented:

`as_int_period` *Convert an object to grates_int_period object*

Description

- Date, POSIXct, and POSIXlt are converted, with the timezone respected, using `clock::as_date()`.
- Character input is parsed using `clock::date_parse()`.

Usage

```
as_int_period(x, ...)

## Default S3 method:
as_int_period(x, n = 1L, origin = 0L, ...)
```

Arguments

`x`	An object to convert.
`...`	Not currently used.
`n`	An integer indicating the (fixed) number of days used for grouping; defaults to 1.
`origin`	Day on which the grouping begins (default 0).

Value

A `grates_int_period` object.

Note

Internally `grates_int_period` objects are stored as the number of days to the earliest day in the specified grouping.

Examples

```
as_int_period(0:10, n = 2)
```

`as_month` *Convert an object to grates_month object*

Description

- Date, POSIXct, and POSIXlt are converted, with the timezone respected, using `clock::as_date()`.
- Character input is parsed using `clock::date_parse()`.

Usage

```
as_month(x, ...)

## Default S3 method:
as_month(x, n = 1L, origin = 0L, ...)

## S3 method for class 'character'
as_month(x, n = 1L, origin = 0L, format = NULL, locale = clock_locale(), ...)

## S3 method for class 'factor'
as_month(x, n = 1L, origin = 0L, format = NULL, locale = clock_locale(), ...)

## S3 method for class 'factor'
as_quarter(x, format = NULL, locale = clock_locale(), ...)
```

Arguments

`x`	An object to convert.
`...`	Not currently used.
`n`	Number of months that are being grouped by (default 1).
`origin`	Month since the Unix epoch where grouping begins (default 0).
`format`	[character / NULL]
	A format string. A combination of the following commands, or `NULL`, in which case a default format string is used.
	A vector of multiple format strings can be supplied. They will be tried in the order they are provided.
	Year

- %C: The century as a decimal number. The modified command %NC where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.
- %y: The last two decimal digits of the year. If the century is not otherwise specified (e.g. with %C), values in the range [69 - 99] are presumed to refer to the years [1969 - 1999], and values in the range [00 - 68] are presumed to refer to the years [2000 - 2068]. The modified command %Ny, where N is a positive decimal integer, specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.
- %Y: The year as a decimal number. The modified command %NY where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 4. Leading zeroes are permitted but not required.

Month

- %b, %B, %h: The locale's full or abbreviated case-insensitive month name.
- %m: The month as a decimal number. January is 1. The modified command %Nm where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.

Day

- %d, %e: The day of the month as a decimal number. The modified command %Nd where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.

Day of the week

- %a, %A: The locale's full or abbreviated case-insensitive weekday name.
- %w: The weekday as a decimal number (0-6), where Sunday is 0. The modified command %Nw where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 1. Leading zeroes are permitted but not required.

ISO 8601 week-based year

- %g: The last two decimal digits of the ISO week-based year. The modified command %Ng where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.
- %G: The ISO week-based year as a decimal number. The modified command %NG where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 4. Leading zeroes are permitted but not required.
- %V: The ISO week-based week number as a decimal number. The modified command %NV where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.

- %u: The ISO weekday as a decimal number (`1-7`), where Monday is `1`. The modified command %Nu where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `1`. Leading zeroes are permitted but not required.

Week of the year

- %U: The week number of the year as a decimal number. The first Sunday of the year is the first day of week `01`. Days of the same year prior to that are in week `00`. The modified command %NU where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required.
- %W: The week number of the year as a decimal number. The first Monday of the year is the first day of week `01`. Days of the same year prior to that are in week `00`. The modified command %NW where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required.

Day of the year

- %j: The day of the year as a decimal number. January 1 is `1`. The modified command %Nj where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `3`. Leading zeroes are permitted but not required.

Date

- %D, %x: Equivalent to %m/%d/%y.
- %F: Equivalent to %Y-%m-%d. If modified with a width (like %NF), the width is applied to only %Y.

Time of day

- %H: The hour (24-hour clock) as a decimal number. The modified command %NH where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required.
- %I: The hour (12-hour clock) as a decimal number. The modified command %NI where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required.
- %M: The minutes as a decimal number. The modified command %NM where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required.
- %S: The seconds as a decimal number. Leading zeroes are permitted but not required. If encountered, the `locale` determines the decimal point character. Generally, the maximum number of characters to read is determined by the precision that you are parsing at. For example, a precision of `"second"` would read a maximum of 2 characters, while a precision of `"millisecond"` would read a maximum of 6 (2 for the values before the decimal point, 1 for the decimal point, and 3 for the values after it). The modified command %NS, where `N` is a positive decimal integer, can be used

to exactly specify the maximum number of characters to read. This is only useful if you happen to have seconds with more than 1 leading zero.

- %p: The `locale`'s equivalent of the AM/PM designations associated with a 12-hour clock. The command %I must precede %p in the format string.
- %R: Equivalent to %H:%M.
- %T, %X: Equivalent to %H:%M:%S.
- %r: Equivalent to %I:%M:%S %p.

Time zone

- %z: The offset from UTC in the format [+|-]hh[mm]. For example `-0430` refers to 4 hours 30 minutes behind UTC. And `04` refers to 4 hours ahead of UTC. The modified command %Ez parses a `:` between the hours and minutes and leading zeroes on the hour field are optional: [+|-]h[h][:mm]. For example `-04:30` refers to 4 hours 30 minutes behind UTC. And 4 refers to 4 hours ahead of UTC.
- %Z: The full time zone name or the time zone abbreviation, depending on the function being used. A single word is parsed. This word can only contain characters that are alphanumeric, or one of `'_'`, `'/'`, `'-'` or `'+'`.

Miscellaneous

- %c: A date and time representation. Equivalent to %a %b %d %H:%M:%S %Y.
- `%%`: A % character.
- %n: Matches one white space character. %n, %t, and a space can be combined to match a wide range of white-space patterns. For example `"%n "` matches one or more white space characters, and `"%n%t%t"` matches one to three white space characters.
- %t: Matches zero or one white space characters.

`locale` [clock_locale]

A locale object created from `clock_locale()`.

Value

A `grates_month` object.

Note

Internally `grates_month` objects are stored as the number of months (starting at 0) since the Unix Epoch (1970-01-01) to the earliest month in the grouping. Precision is only to the month level (i.e. the day of the month is always dropped).

References

The algorithm to convert between dates and months relative to the UNIX Epoch comes from the work of Davis Vaughan in the unreleased datea package.

Examples

```
as_month(Sys.Date())
as_month(as.POSIXct("2019-03-04 01:01:01", tz = "America/New_York"), interval = 2)
as_month("2019-05-03")
```

`as_period` *Convert an object to grates_period object*

Description

- Date, POSIXct, and POSIXlt are converted, with the timezone respected, using `clock::as_date()`.
- Character input is parsed using `clock::date_parse()`.

Usage

```
as_period(x, ...)

## Default S3 method:
as_period(x, n = 1L, origin = 0L, ...)

## S3 method for class 'character'
as_period(x, n = 1L, origin = 0L, format = NULL, locale = clock_locale(), ...)

## S3 method for class 'factor'
as_period(x, n = 1L, origin = 0L, format = NULL, locale = clock_locale(), ...)
```

Arguments

`x`	An object to convert.
`...`	Not currently used.
`n`	An integer indicating the (fixed) number of days used for grouping; defaults to 1.
`origin`	Month since the Unix epoch where grouping begins (default 0).
`format`	[character / NULL] A format string. A combination of the following commands, or `NULL`, in which case a default format string is used. A vector of multiple format strings can be supplied. They will be tried in the order they are provided. **Year** • %C: The century as a decimal number. The modified command %NC where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required.

- %y: The last two decimal digits of the year. If the century is not otherwise specified (e.g. with %C), values in the range [69 - 99] are presumed to refer to the years [1969 - 1999], and values in the range [00 - 68] are presumed to refer to the years [2000 - 2068]. The modified command %Ny, where `N` is a positive decimal integer, specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required.
- %Y: The year as a decimal number. The modified command %NY where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `4`. Leading zeroes are permitted but not required.

Month

- %b, %B, %h: The `locale`'s full or abbreviated case-insensitive month name.
- %m: The month as a decimal number. January is `1`. The modified command %Nm where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required.

Day

- %d, %e: The day of the month as a decimal number. The modified command %Nd where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required.

Day of the week

- %a, %A: The `locale`'s full or abbreviated case-insensitive weekday name.
- %w: The weekday as a decimal number (`0-6`), where Sunday is `0`. The modified command %Nw where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `1`. Leading zeroes are permitted but not required.

ISO 8601 week-based year

- %g: The last two decimal digits of the ISO week-based year. The modified command %Ng where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required.
- %G: The ISO week-based year as a decimal number. The modified command %NG where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `4`. Leading zeroes are permitted but not required.
- %V: The ISO week-based week number as a decimal number. The modified command %NV where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required.
- %u: The ISO weekday as a decimal number (`1-7`), where Monday is `1`. The modified command %Nu where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `1`. Leading zeroes are permitted but not required.

Week of the year

- %U: The week number of the year as a decimal number. The first Sunday of the year is the first day of week `01`. Days of the same year prior to that are in week `00`. The modified command %NU where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required.
- %W: The week number of the year as a decimal number. The first Monday of the year is the first day of week `01`. Days of the same year prior to that are in week `00`. The modified command %NW where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required.

Day of the year

- %j: The day of the year as a decimal number. January 1 is `1`. The modified command %Nj where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `3`. Leading zeroes are permitted but not required.

Date

- %D, %x: Equivalent to %m/%d/%y.
- %F: Equivalent to %Y-%m-%d. If modified with a width (like %NF), the width is applied to only %Y.

Time of day

- %H: The hour (24-hour clock) as a decimal number. The modified command %NH where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required.
- %I: The hour (12-hour clock) as a decimal number. The modified command %NI where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required.
- %M: The minutes as a decimal number. The modified command %NM where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required.
- %S: The seconds as a decimal number. Leading zeroes are permitted but not required. If encountered, the `locale` determines the decimal point character. Generally, the maximum number of characters to read is determined by the precision that you are parsing at. For example, a precision of `"second"` would read a maximum of 2 characters, while a precision of `"millisecond"` would read a maximum of 6 (2 for the values before the decimal point, 1 for the decimal point, and 3 for the values after it). The modified command %NS, where `N` is a positive decimal integer, can be used to exactly specify the maximum number of characters to read. This is only useful if you happen to have seconds with more than 1 leading zero.
- %p: The `locale`'s equivalent of the AM/PM designations associated with a 12-hour clock. The command %I must precede %p in the format string.

- %R: Equivalent to %H:%M.
- %T, %X: Equivalent to %H:%M:%S.
- %r: Equivalent to %I:%M:%S %p.

Time zone

- %z: The offset from UTC in the format [+|-]hh[mm]. For example `-0430` refers to 4 hours 30 minutes behind UTC. And `04` refers to 4 hours ahead of UTC. The modified command %Ez parses a `:` between the hours and minutes and leading zeroes on the hour field are optional: [+|-]h[h][:mm]. For example `-04:30` refers to 4 hours 30 minutes behind UTC. And `4` refers to 4 hours ahead of UTC.
- %Z: The full time zone name or the time zone abbreviation, depending on the function being used. A single word is parsed. This word can only contain characters that are alphanumeric, or one of `'_'`, `'/'`, `'-'` or `'+'`.

Miscellaneous

- %c: A date and time representation. Equivalent to %a %b %d %H:%M:%S %Y.
- `%%`: A % character.
- %n: Matches one white space character. %n, %t, and a space can be combined to match a wide range of white-space patterns. For example `"%n "` matches one or more white space characters, and `"%n%t%t"` matches one to three white space characters.
- %t: Matches zero or one white space characters.

`locale` [clock_locale]

A locale object created from `clock_locale()`.

Value

A `grates_period` object.

Note

Internally `grates_period` objects are stored as the number of days (starting at 0) since the Unix Epoch (1970-01-01) to the earliest day in the specified grouping.

Examples

```
as_period(Sys.Date())
as_period(as.POSIXct("2019-03-04 01:01:01", tz = "America/New_York"), interval = 2)
as_period("2019-05-03")
```

as_quarter *Convert an object to grates_quarter object*

Description

- Date, POSIXct, and POSIXlt are converted, with the timezone respected, using `clock::as_date()`.
- Character input is parsed using `clock::date_parse()`.

Usage

```
as_quarter(x, ...)

## Default S3 method:
as_quarter(x, ...)

## S3 method for class 'character'
as_quarter(x, format = NULL, locale = clock_locale(), ...)
```

Arguments

`x`	An object to convert.
`...`	Not currently used.
`format`	[character / NULL] A format string. A combination of the following commands, or `NULL`, in which case a default format string is used. A vector of multiple format strings can be supplied. They will be tried in the order they are provided. **Year** • %C: The century as a decimal number. The modified command %NC where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required. • %y: The last two decimal digits of the year. If the century is not otherwise specified (e.g. with %C), values in the range [69 - 99] are presumed to refer to the years [1969 - 1999], and values in the range [00 - 68] are presumed to refer to the years [2000 - 2068]. The modified command %Ny, where `N` is a positive decimal integer, specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required. • %Y: The year as a decimal number. The modified command %NY where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `4`. Leading zeroes are permitted but not required. **Month**

- %b, %B, %h: The `locale`'s full or abbreviated case-insensitive month name.
- %m: The month as a decimal number. January is `1`. The modified command %Nm where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required.

Day

- %d, %e: The day of the month as a decimal number. The modified command %Nd where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required.

Day of the week

- %a, %A: The `locale`'s full or abbreviated case-insensitive weekday name.
- %w: The weekday as a decimal number (`0-6`), where Sunday is `0`. The modified command %Nw where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `1`. Leading zeroes are permitted but not required.

ISO 8601 week-based year

- %g: The last two decimal digits of the ISO week-based year. The modified command %Ng where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required.
- %G: The ISO week-based year as a decimal number. The modified command %NG where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `4`. Leading zeroes are permitted but not required.
- %V: The ISO week-based week number as a decimal number. The modified command %NV where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required.
- %u: The ISO weekday as a decimal number (`1-7`), where Monday is `1`. The modified command %Nu where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `1`. Leading zeroes are permitted but not required.

Week of the year

- %U: The week number of the year as a decimal number. The first Sunday of the year is the first day of week `01`. Days of the same year prior to that are in week `00`. The modified command %NU where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required.
- %W: The week number of the year as a decimal number. The first Monday of the year is the first day of week `01`. Days of the same year prior to that are in week `00`. The modified command %NW where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required.

Day of the year

- %j: The day of the year as a decimal number. January 1 is 1. The modified command %Nj where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 3. Leading zeroes are permitted but not required.

Date

- %D, %x: Equivalent to %m/%d/%y.
- %F: Equivalent to %Y-%m-%d. If modified with a width (like %NF), the width is applied to only %Y.

Time of day

- %H: The hour (24-hour clock) as a decimal number. The modified command %NH where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.
- %I: The hour (12-hour clock) as a decimal number. The modified command %NI where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.
- %M: The minutes as a decimal number. The modified command %NM where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.
- %S: The seconds as a decimal number. Leading zeroes are permitted but not required. If encountered, the locale determines the decimal point character. Generally, the maximum number of characters to read is determined by the precision that you are parsing at. For example, a precision of "second" would read a maximum of 2 characters, while a precision of "millisecond" would read a maximum of 6 (2 for the values before the decimal point, 1 for the decimal point, and 3 for the values after it). The modified command %NS, where N is a positive decimal integer, can be used to exactly specify the maximum number of characters to read. This is only useful if you happen to have seconds with more than 1 leading zero.
- %p: The locale's equivalent of the AM/PM designations associated with a 12-hour clock. The command %I must precede %p in the format string.
- %R: Equivalent to %H:%M.
- %T, %X: Equivalent to %H:%M:%S.
- %r: Equivalent to %I:%M:%S %p.

Time zone

- %z: The offset from UTC in the format [+|-]hh[mm]. For example -0430 refers to 4 hours 30 minutes behind UTC. And 04 refers to 4 hours ahead of UTC. The modified command %Ez parses a : between the hours and minutes and leading zeroes on the hour field are optional: [+|-]h[h][:mm]. For example -04:30 refers to 4 hours 30 minutes behind UTC. And 4 refers to 4 hours ahead of UTC.

- %Z: The full time zone name or the time zone abbreviation, depending on the function being used. A single word is parsed. This word can only contain characters that are alphanumeric, or one of '_', '/', '-' or '+'.

Miscellaneous

- %c: A date and time representation. Equivalent to %a %b %d %H:%M:%S %Y.
- %%: A % character.
- %n: Matches one white space character. %n, %t, and a space can be combined to match a wide range of white-space patterns. For example "%n " matches one or more white space characters, and "%n%t%t" matches one to three white space characters.
- %t: Matches zero or one white space characters.

`locale` [clock_locale]

A locale object created from `clock_locale()`.

Value

A `grates_quarter` object.

Note

Internally `grates_quarter` objects are stored as the number of quarters (starting at 0) since the Unix Epoch (1970-01-01)

Examples

```
as_quarter(Sys.Date())
as_quarter(as.POSIXct("2019-03-04 01:01:01", tz = "America/New_York"))
as_quarter("2019-05-03")
```

---

as_year *Convert an object to grates_year object*

Description

- Date, POSIXct, and POSIXlt are converted, with the timezone respected, using `clock::as_date()`.
- Character input is parsed using `clock::date_parse()`.

Usage

```
as_year(x, ...)

## Default S3 method:
as_year(x, ...)

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

```
as_year(x, format = NULL, locale = clock_locale(), ...)

## S3 method for class 'factor'
as_year(x, format = NULL, locale = clock_locale(), ...)
```

Arguments

x	An object to convert.
...	Not currently used.
format	[character / NULL]

A format string. A combination of the following commands, or NULL, in which case a default format string is used.

A vector of multiple format strings can be supplied. They will be tried in the order they are provided.

Year

- %C: The century as a decimal number. The modified command %NC where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.
- %y: The last two decimal digits of the year. If the century is not otherwise specified (e.g. with %C), values in the range [69 - 99] are presumed to refer to the years [1969 - 1999], and values in the range [00 - 68] are presumed to refer to the years [2000 - 2068]. The modified command %Ny, where N is a positive decimal integer, specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.
- %Y: The year as a decimal number. The modified command %NY where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 4. Leading zeroes are permitted but not required.

Month

- %b, %B, %h: The locale's full or abbreviated case-insensitive month name.
- %m: The month as a decimal number. January is 1. The modified command %Nm where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.

Day

- %d, %e: The day of the month as a decimal number. The modified command %Nd where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.

Day of the week

- %a, %A: The locale's full or abbreviated case-insensitive weekday name.

- %w: The weekday as a decimal number (0-6), where Sunday is 0. The modified command %Nw where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 1. Leading zeroes are permitted but not required.

ISO 8601 week-based year

- %g: The last two decimal digits of the ISO week-based year. The modified command %Ng where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.
- %G: The ISO week-based year as a decimal number. The modified command %NG where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 4. Leading zeroes are permitted but not required.
- %V: The ISO week-based week number as a decimal number. The modified command %NV where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.
- %u: The ISO weekday as a decimal number (1-7), where Monday is 1. The modified command %Nu where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 1. Leading zeroes are permitted but not required.

Week of the year

- %U: The week number of the year as a decimal number. The first Sunday of the year is the first day of week 01. Days of the same year prior to that are in week 00. The modified command %NU where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.
- %W: The week number of the year as a decimal number. The first Monday of the year is the first day of week 01. Days of the same year prior to that are in week 00. The modified command %NW where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.

Day of the year

- %j: The day of the year as a decimal number. January 1 is 1. The modified command %Nj where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 3. Leading zeroes are permitted but not required.

Date

- %D, %x: Equivalent to %m/%d/%y.
- %F: Equivalent to %Y-%m-%d. If modified with a width (like %NF), the width is applied to only %Y.

Time of day

- %H: The hour (24-hour clock) as a decimal number. The modified command %NH where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.

- %I: The hour (12-hour clock) as a decimal number. The modified command %NI where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required.
- %M: The minutes as a decimal number. The modified command %NM where `N` is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is `2`. Leading zeroes are permitted but not required.
- %S: The seconds as a decimal number. Leading zeroes are permitted but not required. If encountered, the `locale` determines the decimal point character. Generally, the maximum number of characters to read is determined by the precision that you are parsing at. For example, a precision of `"second"` would read a maximum of 2 characters, while a precision of `"millisecond"` would read a maximum of 6 (2 for the values before the decimal point, 1 for the decimal point, and 3 for the values after it). The modified command %NS, where `N` is a positive decimal integer, can be used to exactly specify the maximum number of characters to read. This is only useful if you happen to have seconds with more than 1 leading zero.
- %p: The `locale`'s equivalent of the AM/PM designations associated with a 12-hour clock. The command %I must precede %p in the format string.
- %R: Equivalent to %H:%M.
- %T, %X: Equivalent to %H:%M:%S.
- %r: Equivalent to %I:%M:%S %p.

Time zone

- %z: The offset from UTC in the format [+|-]hh[mm]. For example `-0430` refers to 4 hours 30 minutes behind UTC. And `04` refers to 4 hours ahead of UTC. The modified command %Ez parses a `:` between the hours and minutes and leading zeroes on the hour field are optional: [+|-]h[h][:mm]. For example `-04:30` refers to 4 hours 30 minutes behind UTC. And `4` refers to 4 hours ahead of UTC.
- %Z: The full time zone name or the time zone abbreviation, depending on the function being used. A single word is parsed. This word can only contain characters that are alphanumeric, or one of `'_'`, `'/'`, `'-'` or `'+'`.

Miscellaneous

- %c: A date and time representation. Equivalent to %a %b %d %H:%M:%S %Y.
- `%%`: A % character.
- %n: Matches one white space character. %n, %t, and a space can be combined to match a wide range of white-space patterns. For example `"%n "` matches one or more white space characters, and `"%n%t%t"` matches one to three white space characters.
- %t: Matches zero or one white space characters.

`locale` [clock_locale]

A locale object created from `clock_locale()`.

Value

A `grates_year` object.

Examples

```
as_year(Sys.Date())
as_year(as.POSIXct("2019-03-04 01:01:01", tz = "America/New_York"), interval = 2)
as_year("2019-05-03")
```

---

as_yearweek *Convert an object to grates_yearweek*

Description

- Date, POSIXct, and POSIXlt are converted, with the timezone respected, using `clock::as_date()`.
- Character input is parsed by two methods. Firstly, if `format = NULL` then character input is first checked to see if it is in the format of "YYYY-Www" (e.g. "2021-W03") and parsed accordingly. If it is in a different format, or `format` is non-NULL then it is parsed using `clock::date_parse()`.

Usage

```
as_yearweek(x, firstday = 1L, ...)

## Default S3 method:
as_yearweek(x, firstday = 1L, ...)

## S3 method for class 'character'
as_yearweek(
  x,
  firstday = 1L,
  format = "%Y-%m-%d",
  locale = clock_locale(),
  ...
)

## S3 method for class 'factor'
as_yearweek(x, firstday = 1L, format = NULL, locale = clock_locale(), ...)
```

Arguments

`x`	An object to coerce to yearweekly.
`firstday`	An integer representing the day the week starts on from 1 (Monday) to 7 (Sunday).
`...`	Not currently used.
`format`	[character / NULL] A format string. A combination of the following commands, or `NULL`, in which case a default format string is used.

A vector of multiple format strings can be supplied. They will be tried in the order they are provided.

Year

- %C: The century as a decimal number. The modified command %NC where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.
- %y: The last two decimal digits of the year. If the century is not otherwise specified (e.g. with %C), values in the range [69 - 99] are presumed to refer to the years [1969 - 1999], and values in the range [00 - 68] are presumed to refer to the years [2000 - 2068]. The modified command %Ny, where N is a positive decimal integer, specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.
- %Y: The year as a decimal number. The modified command %NY where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 4. Leading zeroes are permitted but not required.

Month

- %b, %B, %h: The locale's full or abbreviated case-insensitive month name.
- %m: The month as a decimal number. January is 1. The modified command %Nm where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.

Day

- %d, %e: The day of the month as a decimal number. The modified command %Nd where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.

Day of the week

- %a, %A: The locale's full or abbreviated case-insensitive weekday name.
- %w: The weekday as a decimal number (0-6), where Sunday is 0. The modified command %Nw where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 1. Leading zeroes are permitted but not required.

ISO 8601 week-based year

- %g: The last two decimal digits of the ISO week-based year. The modified command %Ng where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.
- %G: The ISO week-based year as a decimal number. The modified command %NG where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 4. Leading zeroes are permitted but not required.

- %V: The ISO week-based week number as a decimal number. The modified command %NV where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.
- %u: The ISO weekday as a decimal number (1-7), where Monday is 1. The modified command %Nu where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 1. Leading zeroes are permitted but not required.

Week of the year

- %U: The week number of the year as a decimal number. The first Sunday of the year is the first day of week 01. Days of the same year prior to that are in week 00. The modified command %NU where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.
- %W: The week number of the year as a decimal number. The first Monday of the year is the first day of week 01. Days of the same year prior to that are in week 00. The modified command %NW where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.

Day of the year

- %j: The day of the year as a decimal number. January 1 is 1. The modified command %Nj where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 3. Leading zeroes are permitted but not required.

Date

- %D, %x: Equivalent to %m/%d/%y.
- %F: Equivalent to %Y-%m-%d. If modified with a width (like %NF), the width is applied to only %Y.

Time of day

- %H: The hour (24-hour clock) as a decimal number. The modified command %NH where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.
- %I: The hour (12-hour clock) as a decimal number. The modified command %NI where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.
- %M: The minutes as a decimal number. The modified command %NM where N is a positive decimal integer specifies the maximum number of characters to read. If not specified, the default is 2. Leading zeroes are permitted but not required.
- %S: The seconds as a decimal number. Leading zeroes are permitted but not required. If encountered, the locale determines the decimal point character. Generally, the maximum number of characters to read is determined by the precision that you are parsing at. For example, a precision

of "second" would read a maximum of 2 characters, while a precision of "millisecond" would read a maximum of 6 (2 for the values before the decimal point, 1 for the decimal point, and 3 for the values after it). The modified command %NS, where N is a positive decimal integer, can be used to exactly specify the maximum number of characters to read. This is only useful if you happen to have seconds with more than 1 leading zero.

- %p: The locale's equivalent of the AM/PM designations associated with a 12-hour clock. The command %I must precede %p in the format string.
- %R: Equivalent to %H:%M.
- %T, %X: Equivalent to %H:%M:%S.
- %r: Equivalent to %I:%M:%S %p.

Time zone

- %z: The offset from UTC in the format [+|-]hh[mm]. For example -0430 refers to 4 hours 30 minutes behind UTC. And 04 refers to 4 hours ahead of UTC. The modified command %Ez parses a : between the hours and minutes and leading zeroes on the hour field are optional: [+|-]h[h][:mm]. For example -04:30 refers to 4 hours 30 minutes behind UTC. And 4 refers to 4 hours ahead of UTC.
- %Z: The full time zone name or the time zone abbreviation, depending on the function being used. A single word is parsed. This word can only contain characters that are alphanumeric, or one of '_', '/', '-' or '+'.

Miscellaneous

- %c: A date and time representation. Equivalent to %a %b %d %H:%M:%S %Y.
- %%: A % character.
- %n: Matches one white space character. %n, %t, and a space can be combined to match a wide range of white-space patterns. For example "%n " matches one or more white space characters, and "%n%t%t" matches one to three white space characters.
- %t: Matches zero or one white space characters.

`locale` [clock_locale]

A locale object created from `clock_locale()`.

Value

A `grates_yearweek` object.

Note

Internally `grates_yearweek` objects are stored as the number of weeks from the date of the `firstday` nearest the Unix Epoch (1970-01-01). That is:

```
- 1969-12-29 for `firstday` as Monday
- 1969-12-30 for `firstday` as Tuesday
- 1969-12-31 for `firstday` as Wednesday
- 1970-01-01 for `firstday` as Thursday
- 1970-01-02 for `firstday` as Friday
- 1970-01-03 for `firstday` as Saturday
- 1970-01-04 for `firstday` as Sunday
```

See Also

clock::date_parse()

Examples

```
as_yearweek(Sys.Date())
as_yearweek(as.POSIXct("2019-03-04 01:01:01", tz = "America/New_York"))
as_yearweek("2019-05-03", firstday = 5L)
as_yearweek("2021-W03", format = NULL)
```

---

format.grates_int_period *Format a grates_int_period object*

Description

Format a grates_int_period object

Usage

```
## S3 method for class 'grates_int_period'
format(x, sep = "to", ...)
```

Arguments

x	A grates_int_period object.
sep	Where more than one day is grouped with others, sep is placed between the upper and lower bounds when printing.
...	Not currently used.
format	The format to use for the bounds of each grates_int_period entry.

---

format.grates_month *Format a grates_month object*

Description

Format a grates_month object

Usage

```
## S3 method for class 'grates_month'
format(x, format = "%Y-%b", sep = "to", ...)
```

Arguments

x	A grates_month object.
format	The format to use for the bounds of each grates_month entry.
sep	Where more than one month is grouped with others, sep is placed between the upper and lower bounds when printing.
...	Not currently used.

---

format.grates_period *Format a grates_period object*

Description

Format a grates_period object

Usage

```
## S3 method for class 'grates_period'
format(x, format = "%Y-%m-%d", sep = "to", ...)
```

Arguments

x	A grates_period object.
format	The format to use for the bounds of each grates_period entry.
sep	Where more than one day is grouped with others, sep is placed between the upper and lower bounds when printing.
...	Not currently used.

---

format.grates_quarter *Format a grates_quarter object*

Description

Format a grates_quarter object

Usage

```
## S3 method for class 'grates_quarter'
format(x, ...)
```

Arguments

x	A grates_quarter object.
...	Not currently used.

format.grates_year *Format a grates_year object*

Description

Format a grates_year object

Usage

```
## S3 method for class 'grates_year'
format(x, ...)
```

Arguments

x	A grates_year object.
...	Not currently used.

format.grates_yearweek *Format a grates_yearweek object*

Description

Format a grates_yearweek object

Usage

```
## S3 method for class 'grates_yearweek'
format(x, ...)
```

Arguments

x	A grates_yearweek object.
...	Not currently used.

grates_accessors *Grouped date accessors*

Description

Generics and methods to work with grouped date objects.

Usage

```
get_n(x, ...)

## Default S3 method:
get_n(x, ...)

## S3 method for class 'grates_month'
get_n(x, ...)

## S3 method for class 'grates_period'
get_n(x, ...)

## S3 method for class 'grates_int_period'
get_n(x, ...)

get_firstday(x, ...)

## Default S3 method:
get_firstday(x, ...)

## S3 method for class 'grates_yearweek'
get_firstday(x, ...)

get_week(x, ...)

## Default S3 method:
get_week(x, ...)

## S3 method for class 'grates_yearweek'
get_week(x, ...)

get_quarter(x, ...)

## Default S3 method:
get_quarter(x, ...)

## S3 method for class 'grates_quarter'
get_quarter(x, ...)
```

```
get_year(x, ...)

## Default S3 method:
get_year(x, ...)

## S3 method for class 'grates_yearweek'
get_year(x, ...)

## S3 method for class 'grates_quarter'
get_year(x, ...)

## S3 method for class 'grates_year'
get_year(x, ...)

get_date_range(x, ...)

## Default S3 method:
get_date_range(x, ...)

## S3 method for class 'grates_yearweek'
get_date_range(x, ...)

## S3 method for class 'grates_month'
get_date_range(x, ...)

## S3 method for class 'grates_quarter'
get_date_range(x, ...)

## S3 method for class 'grates_period'
get_date_range(x, ...)

## S3 method for class 'grates_year'
get_date_range(x, ...)

## S3 method for class 'grates_int_period'
get_date_range(x, ...)
```

Arguments

x	A grates object.
...	Not currently used.

Value

- `get_year()`: The corresponding year values as integer for <grates_year> and <greats_quarter> objects.
- `get_quarter()`: The corresponding quarter values as integer for <greats_quarter> objects.

int_period *Construct a grates_int_period object*

Description

`period()` is a constructor for a <grates_int_period> object

Usage

```
int_period(x = integer(), n = 1L, origin = 0L)

is_int_period(x)
```

Arguments

`x`	Integer vector representing the number of days at the beginning of the specified grouping.
`n`	An integer indicating the (fixed) number of days used for grouping; defaults to 1.
`origin`	Day on which the grouping begins (default 0).

Value

a <grates_int_period> vector.

Note

To allow easy comparison between <grates_int_period> object, the `origin` will be stored as it's value modulo `n` (i.e `origin <-origin %% n`).

Examples

```
int_period(1, n = 2, origin = 1)
```

is_grate *Is object a grouped date*

Description

Is object a grouped date

Usage

```
is_grate(x)
```

Arguments

x	Grouped date object.

Value

Logical.

Examples

```
is_grate(as_yearweek(Sys.Date()))
```

---

month *Construct a grates_month object*

Description

`month()` is a constructor for a <grates_month> object

Usage

```
month(x = integer(), n = 1L, origin = 0L)

is_month(x)
```

Arguments

x	Integer vector representing the number of months since the Unix epoch (1970-01-01).
n	Number of months that are being grouped by (default 1).
origin	Month since the Unix epoch where grouping begins (default 0).

References

The algorithm to convert between dates and months relative to the UNIX Epoch comes from the work of Davis Vaughan in the unreleased datea package.

Examples

```
month(1)
month(c(4, 7), n = 3, origin = 1)
```

period *Construct a grates_period object*

Description

`period()` is a constructor for a <grates_period> object

Usage

```
period(x = integer(), n = 1L, origin = 0L)

is_period(x)
```

Arguments

`x`	Integer vector representing the number of days since the Unix epoch (1970-01-01) at the beginning of the specified grouping.
`n`	An integer indicating the (fixed) number of days used for grouping; defaults to 1.
`origin`	Day since the Unix epoch where grouping begins (default 0).

Value

a <grates_period> vector.

Note

To allow easy comparison between <grates_period> object, the `origin` will be stored as it's value modulo `n` (i.e `origin <- origin %% n`).

Examples

```
period(1, n = 2, origin = 1)
```

quarter *Construct a grates_quarter object*

Description

`quarter()` is a constructor for a <grates_quarter> object

Usage

```
quarter(x = integer())

is_quarter(x)
```

Arguments

x	Integer vector representing the number of quarters (starting at 0), since the Unix epoch (1970-01-01).

References

The algorithm to convert between dates and months relative to the UNIX Epoch comes from the work of Davis Vaughan in the unreleased datea package.

Examples

```
quarter(0:3)
```

---

scale_x_grates_int_period *<grates_int_period> scale*

Description

ggplot2 scale for <grates_int_period> vector.

Usage

```
scale_x_grates_int_period(n.breaks = 6, n, origin)
```

Arguments

n.breaks	Approximate number of breaks calculated using `scales::breaks_pretty` (default 6).
n	Number of days used for the original grouping.
origin	Original day on which the grouping began.

Value

A scale for use with ggplot2.

scale_x_grates_month *<grates_month> scale*

Description

ggplot2 scale for <grates_month> vector.

Usage

```
scale_x_grates_month(
  n.breaks = 6,
  format = "%Y-%m-%d",
  bounds_format = "%Y-%b",
  sep = "to",
  n,
  origin
)
```

Arguments

n.breaks	Approximate number of breaks calculated using scales::breaks_pretty (default 6).
format	Format to use if "Date" scales are required. If NULL then labels are centralised and of the form "lower category bound to upper category bound". If not NULL then the value is used by format.Date() and can be any input acceptable by that function (defaults to "%Y-%m-%d).
bounds_format	Format to use for grouped date labels. Only used if format is NULL.
sep	Separator to use for grouped date labels.
n	Number of months used for the original grouping.
origin	Month since the Unix epoch used in the original grouping.

Value

A scale for use with ggplot2.

scale_x_grates_period *<grates_period> scale*

Description

ggplot2 scale for <grates_period> vector.

Usage

```
scale_x_grates_period(n.breaks = 6, format = "%Y-%m-%d", n, origin)
```

Arguments

n.breaks	Approximate number of breaks calculated using scales::breaks_pretty (default 6).
format	Format to use for x scale. Passed to format.Date().
n	Number of months used for the original grouping.
origin	Month since the Unix epoch used in the original grouping.

Value

A scale for use with ggplot2.

---

scale_x_grates_quarter *<grates_quarter> scale*

Description

ggplot2 scale for <grates_quarter> vector.

Usage

```
scale_x_grates_quarter(n.breaks = 6, format = NULL)
```

Arguments

n.breaks	Approximate number of breaks calculated using scales::breaks_pretty (default 6).
format	Format to use if "Date" scales are required. If NULL then labels are centralised and of the form "YYYY-Qq. If not NULL then the value is used by format.Date() and can be any input acceptable by that function.

Value

A scale for use with ggplot2.

scale_x_grates_year <grates_year> scale

Description

ggplot2 scale for <grates_year> vector.

Usage

```
scale_x_grates_year(..., n.breaks = 6)
```

Arguments

...	Not currently used.
n.breaks	Approximate number of breaks calculated using scales::breaks_pretty (default 6).

Value

A scale for use with ggplot2.

scale_x_grates_yearweek <grates_yearweek> scale

Description

ggplot2 scale for <grates_yearweek> vector.

Usage

```
scale_x_grates_yearweek(..., n.breaks = 6, firstday, format = NULL)
```

Arguments

...	Not currently used.
n.breaks	Approximate number of breaks calculated using scales::breaks_pretty (default 6).
firstday	Integer value of the first weekday: 1 (Monday) to 7 (Sunday).
format	Format to use if "Date" scales are required. If NULL (default) then labels are centralised and of the form "lower category bound to upper category bound". If not NULL then the value is used by format.Date() and can be any input acceptable by that function.

Value

A scale for use with ggplot2.

year *Construct a grates_year object*

Description

`year()` is a constructor for a <grates_year> object

Usage

```
year(x = integer())

is_year(x)
```

Arguments

`x`	Integer vector representing the year.

Examples

```
year(2021)
```

yearweek *Construct a grates_yearweek object*

Description

`yearweek()` is a constructor for a <grates_yearweek> object.

Usage

```
yearweek(x = integer(), firstday = 1L)

is_yearweek(x)
```

Arguments

`x`	Integer vector representing the number of weeks.
`firstday`	An integer representing the day the week starts on from 1 (Monday) to 7 (Sunday). <grates_yearweek> objects are stored as the number of weeks (startint at 0) from the date of the specified `firstday` nearest the Unix Epoch (1970-01-01). That is, the number of seven day periods from:

- 1969-12-29 for `firstday` equal to 1 (Monday)
- 1969-12-30 for `firstday` equal to 2 Tuesday
- 1969-12-31 for `firstday` equal to 3 Wednesday
- 1970-01-01 for `firstday` equal to 4 Thursday
- 1970-01-02 for `firstday` equal to 5 Friday
- 1970-01-03 for `firstday` equal to 6 Saturday
- 1970-01-04 for `firstday` equal to 7 Sunday

Examples

```
yearweek(0:5)
```

Index