

# Package ‘timeperiodsR’

May 8, 2026

**Type** Package

**Title** Simple Definition Of Time Intervals

**Version** 0.7.5

**License** GPL-2

**Author** Alexey Seleznev [aut, cre]

**Maintainer** Alexey Seleznev <selesnow@gmail.com>

**Description** Simple definition of time intervals for the current, previous, and next week, month, quarter and year.

**BugReports** <https://github.com/selesnow/timeperiodsR/issues>

**URL** <https://selesnow.github.io/timeperiodsR/>,  
<https://t.me/R4marketing>,  
<https://www.youtube.com/playlist?list=PLD2LDq8edf4qed2KVKfXmKdh00Qcdj9gw>

**Imports** lubridate

**VignetteBuilder** knitr

**Suggests** knitr, rmarkdown, httr

**Encoding** UTF-8

**Language** en, ru

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2026-04-16 23:10:02 UTC

## Contents

timeperiodsR-package . . . . .	2
as_timeperiod . . . . .	4
check_dayoffs . . . . .	5
custom_period . . . . .	6
first_weekend . . . . .	8

first_workday . . . . .	8
last_n_days . . . . .	9
last_n_months . . . . .	10
last_n_quarters . . . . .	12
last_n_weeks . . . . .	13
last_n_years . . . . .	15
last_weekend . . . . .	16
last_workday . . . . .	17
next_month . . . . .	17
next_n_days . . . . .	18
next_n_months . . . . .	20
next_n_quarters . . . . .	21
next_n_weeks . . . . .	23
next_n_years . . . . .	24
next_quarter . . . . .	26
next_week . . . . .	27
next_year . . . . .	28
previous_month . . . . .	29
previous_quarter . . . . .	31
previous_week . . . . .	32
previous_year . . . . .	33
this_month . . . . .	34
this_quarter . . . . .	36
this_week . . . . .	37
this_year . . . . .	38
tpr_operators . . . . .	39
weekends . . . . .	40
weekends_length . . . . .	41
workdays . . . . .	42
workdays_length . . . . .	42
<b>Index</b>	<b>44</b>

---

timeperiodsR-package    *Simple Definition Of Time Intervals*

---

## Description

Simple definition of time intervals for the current, previous, and next week, month, quarter and year.

## Details

The DESCRIPTION file:

```
Package:      timeperiodsR
Type:        Package
Title:       Simple Definition Of Time Intervals
Version:     0.7.5
```

License: GPL-2  
 Authors@R: c(person(given = "Alexey", family = "Seleznev", role = c("aut", "cre"), email = "selesnow@gmail.com"))  
 Author: Alexey Seleznev [aut, cre]  
 Maintainer: Alexey Seleznev <selesnow@gmail.com>  
 Description: Simple definition of time intervals for the current, previous, and next week, month, quarter and year.  
 BugReports: <https://github.com/selesnow/timeperiodsR/issues>  
 URL: <https://selesnow.github.io/timeperiodsR/>, <https://t.me/R4marketing>, <https://www.youtube.com/playlist?list=...>  
 Imports: lubridate  
 VignetteBuilder: knitr  
 Suggests: knitr, rmarkdown, httr  
 Encoding: UTF-8  
 Language: en, ru  
 RoxygenNote: 7.3.2

## Index of help topics:

as_timeperiod	Convert date or string vector to timeperiod.
check_dayoffs	Check if the day is an official day off
custom_period	Custom period
first_weekend	Get first weekend day in period.
first_workday	Get first workday day in period.
last_n_days	Start and end of last n days
last_n_months	Start and end of last n months
last_n_quarters	Start and end of last n quarters
last_n_weeks	Start and end of last n weeks
last_n_years	Start and end of last n years
last_weekend	Get last weekend day in period.
last_workday	Get last workday day in period.
next_month	Start and end of next month
next_n_days	Start and end of next n days
next_n_months	Start and end of next n months
next_n_quarters	Start and end of next n quarters
next_n_weeks	Start and end of next n weeks
next_n_years	Start and end of next n years
next_quarter	Start and end of next quarter
next_week	Start and end of next week
next_year	Start and end of next year
previous_month	Start and end of previous month
previous_quarter	Start and end of previous quarter
previous_week	Start and end of previous week
previous_year	Start and end of previous year
this_month	Start and end of month
this_quarter	Start and end of quarter
this_week	Start and end of week
this_year	Start and end of year
timeperiodsR-package	Simple Definition Of Time Intervals

weekends	Get vector with weekends.
weekends_length	Number of weekdends days in the period.
workdays	Get vector with workdays.
workdays_length	Number of workdays in the period.

**Author(s)**

Alexey Seleznev [aut, cre]

Maintainer: Alexey Seleznev <selesnow@gmail.com>

---

as_timeperiod	<i>Convert date or string vector to timeperiod.</i>
---------------	---

---

**Description**

Convert any date or string vector to tpr class.

**Usage**

```
as_timeperiod(x)
```

**Arguments**

x	Date or string vector
---	-----------------------

**Value**

Object of tpr class

**Author(s)**

Alexey Seleznev

**See Also**

See [custom\\_period\(\)](#)

**Examples**

```
dates <- as.Date(c("2019-09-11",
                  "2019-09-02",
                  "2019-10-11",
                  "2019-08-30"))

dates_tpr <- as_timeperiod(dates)
```

---

check_dayoffs	<i>Check if the day is an official day off</i>
---------------	--

---

### Description

Check any date or date vector for is an official day off or not.

### Usage

```
check_dayoffs(date = NULL,
              year  = NULL,
              month = NULL,
              day   = NULL,
              cc    = getOption("timeperiodsR.official_day_offs_country"),
              pre   = getOption("timeperiodsR.official_day_offs_pre"),
              include_custom_day_offs = TRUE)
```

### Arguments

year	Year for check
month	Month for check
day	Month for check.
date	Date, or date vector for checking
cc	Country, one of ru, ua, kz, by
pre	Including shorter working days, 0 or 1
include_custom_day_offs	Including custom dayoffs from options or global variables

### Details

Function use 'isDayOff() API'.

For get official day offs for your country you must install httr package and switch options `timeperiodsR.official_day_offs` to TRUE or set system variable `TPR_DAY_OFFS=TRUE`.

Now allow next country:

**ru** Russia  
**ua** Ukraine  
**kz** Kazakhstan  
**by** Belarus

Also you can set default country by options or system envariable:

**option** `timeperiodsR.official_day_offs_country`  
**system variable** `TPR_COUNTRY`

And you can include or exclude shorter working days. Using option `timeperiodsR.official_day_offs_pre`:

**0** Exclude shorter work days

**1** Include shorter work days

Day marks:

**0** Workday

**1** Day off

**2** Shorten day off

**3** Custom day off

You can set your custom day offs, for example it can be your vacation. Using option `timeperiodsR.custom_day_offs` or system variable `TPR_CUSTOM_DAY_OFFS`. In `TPR_CUSTOM_DAY_OFFS` you can set custom day offs like comma or semicolon list of dates in format `YYYY-MM-DD`.

### Value

Named vector with date and marks

### Author(s)

Alexey Seleznev

### References

[Official documentation for isDayOff API](#)

### Examples

```
ld <- next_n_weeks(n = 2)
check_dayoffs(date = ld)
```

---

custom\_period

*Custom period*

---

### Description

Create tpr object between start and end dates

### Usage

```
custom_period( start,
               end,
               part = getOption("timeperiodsR.parts"))
```

**Arguments**

start	Start date in YYYY-MM-DD format
end	End date in YYYY-MM-DD format
part	Part of period you need to receive, one of "all", "start", "end", "sequence", "length", "workdays", "weekends", "first_workday", "last_workday", "first_weekend", "last_weekend", "length". See details.

**Details**

You can get object of tpr class with all components or specify which component you need, use part for manage this option:

- all - get all components
- start - get only first date of period
- end - get only last date of period
- start - get vector of all dates in period
- length - get number of dates in period

**Value**

Object of tpr class

**Author(s)**

Alexey Seleznev

**Examples**

```
## To create tpr object between two dates
customper <- custom_period(start = "2019-09-01",
                           end   = "2019-09-05")

## To get vector of date sequences
seq(customper)

## Get number of days of previous months
length(customper)

## To get start of end dates
start(customper)
end(customper)
```

---

first\_weekend            *Get first weekend day in period.*

---

**Description**

Method for get first weekend in timeperiod.

**Usage**

```
first_weekend(x)
```

**Arguments**

x                    tpr object or date vector

**Value**

Vector of dates

**Author(s)**

Alexey Seleznev

**Examples**

```
pm <- previous_month()
# get first weekend
pm_first_weekend <- first_weekend(pm)
```

---

first\_workday            *Get first workday day in period.*

---

**Description**

Method for get first workday in timeperiod.

**Usage**

```
first_workday(x)
```

**Arguments**

x                    tpr object or date vector

**Value**

Vector of dates

**Author(s)**

Alexey Seleznev

**Examples**

```
pm <- previous_month()
# get first workday
pm_first_workday <- first_workday(pm)
```

---

last_n_days	<i>Start and end of last n days</i>
-------------	-------------------------------------

---

**Description**

Defines first and last date in previous period

**Usage**

```
last_n_days(x = Sys.Date(),
            n = 1,
            part = getOption("timeperiodsR.parts"),
            include_current = F)
```

**Arguments**

x	Date object
n	Number of periods for offset
part	Part of period you need to receive, one of "all", "start", "end", "sequence", "length". See details.
include_current	If TRUE including current period in sequence

**Details**

You can get object of tpr class with all components or specify which component you need, use part for manage this option:

- all - get all components
- start - get only first date of period
- end - get only last date of period
- start - get vector of all dates in period
- length - get number of dates in period

**Value**

Object of tpr class

**Author(s)**

Alexey Seleznev

**See Also**

For get next other periods see [last\\_n\\_quarters\(\)](#), [last\\_n\\_months\(\)](#), [last\\_n\\_years\(\)](#), [last\\_n\\_weeks\(\)](#)

**Examples**

```
## To get start, end and sequence of last 10 days,
## exclude today
last10days <- last_n_days(n = 10)

## include today
last10days_2 <- last_n_days(n = 10, include_current = TRUE)

## To get vector of date sequences
last_n_days(n = 10, part = "sequence")
last_n_days(n = 10)$sequence
seq(last10days)
```

---

last_n_months	<i>Start and end of last n months</i>
---------------	---------------------------------------

---

**Description**

Defines first and last date in previous period

**Usage**

```
last_n_months(x = Sys.Date(),
              n = 1,
              part = getOption("timeperiodsR.parts"),
              include_current = F)
```

**Arguments**

x	Date object
n	Number of periods for offset
part	Part of period you need to receive, one of "all", "start", "end", "sequence", "length". See details.
include_current	If TRUE including current period in sequence

## Details

You can get object of tpr class with all components or specify which component you need, use part for manage this option:

- all - get all components
- start - get only first date of period
- end - get only last date of period
- start - get vector of all dates in period
- length - get number of dates in period

## Value

Object of tpr class

## Author(s)

Alexey Seleznev

## See Also

For get next other periods see [last\\_n\\_quarters\(\)](#), [last\\_n\\_days\(\)](#), [last\\_n\\_years\(\)](#), [last\\_n\\_weeks\(\)](#)

## Examples

```
## To get start, end and sequence of last 2 months,  
## exclude current month  
last2month <- last_n_months(n = 2)  
  
## include current month  
last2month_2 <- last_n_months(n = 2, include_current = TRUE)  
  
## To get vector of date sequences  
last_n_months(n = 2, part = "sequence")  
last_n_months(n = 2)$sequence  
seq(last2month)  
  
## Get number of days of last 2 months  
day_nums <- last_n_months(n = 2, part = "length")  
last_n_months()$length  
length(last2month)
```

---

last_n_quarters	<i>Start and end of last n quarters</i>
-----------------	---

---

### Description

Defines first and last date in previous period

### Usage

```
last_n_quarters(x = Sys.Date(),
               n = 1,
               part = getOption("timeperiodsR.parts"),
               include_current = F)
```

### Arguments

x	Date object
n	Number of periods for offset
part	Part of period you need to receive, one of "all", "start", "end", "sequence", "length". See details.
include_current	If TRUE including current period in sequence

### Details

You can get object of tpr class with all components or specify which component you need, use part for manage this option:

- all - get all components
- start - get only first date of period
- end - get only last date of period
- start - get vector of all dates in period
- length - get number of dates in period

### Value

Object of tpr class

### Author(s)

Alexey Seleznev

### See Also

For get next other periods see [last\\_n\\_months\(\)](#), [last\\_n\\_days\(\)](#), [last\\_n\\_years\(\)](#), [last\\_n\\_weeks\(\)](#)

**Examples**

```
## To get start, end and sequence of last 2 quarters,
## exclude current quarter
last2quarters <- last_n_quarters(n = 2)

## include current quarter
last2quarters_2 <- last_n_quarters(n = 2, include_current = TRUE)

## To get vector of date sequences
last_n_quarters(n = 2, part = "sequence")
last_n_quarters(n = 2)$sequence
seq(last2quarters)

## Get number of days of last 2 quarters
day_nums <- last_n_quarters(n = 2, part = "length")
last_n_quarters()$length
length(last2quarters)
```

---

last_n_weeks	<i>Start and end of last n weeks</i>
--------------	--------------------------------------

---

**Description**

Defines first and last date in previous period

**Usage**

```
last_n_weeks(x = Sys.Date(),
             n = 1,
             part = getOption("timeperiodsR.parts"),
             week_start = 1,
             include_current = F)
```

**Arguments**

x	Date object
n	Number of periods for offset
part	Part of period you need to receive, one of "all", "start", "end", "sequence", "length". See details.
week_start	Start of week, default = 1 is Monday, 7 is Sunday
include_current	If TRUE including current period in sequence

## Details

You can get object of tpr class with all components or specify which component you need, use part for manage this option:

- all - get all components
- start - get only first date of period
- end - get only last date of period
- start - get vector of all dates in period
- length - get number of dates in period

## Value

Object of tpr class

## Author(s)

Alexey Seleznev

## See Also

For get next other periods see [last\\_n\\_months\(\)](#), [last\\_n\\_days\(\)](#), [last\\_n\\_years\(\)](#), [last\\_n\\_quarters\(\)](#)

## Examples

```
## To get start, end and sequence of last 2 weeks,  
## exclude current week  
last2weeks <- last_n_weeks(n = 2)  
  
## include current week  
last2weeks_2 <- last_n_weeks(n = 2, include_current = TRUE)  
  
## Get last 2 week with start on Sunday  
lastWeek <- last_n_weeks(n = 2, week_start = 7)  
  
## To get vector of date sequences  
last_n_weeks(n = 2, part = "sequence")  
last_n_weeks(n = 2)$sequence  
seq(last2weeks)  
  
## Get number of days of last 2 weeks  
day_nums <- last_n_weeks(n = 2, part = "length")  
last_n_weeks(n = 2)$length  
length(last2weeks)
```

---

last_n_years	<i>Start and end of last n years</i>
--------------	--------------------------------------

---

### Description

Defines first and last date in previous period

### Usage

```
last_n_years(x = Sys.Date(),
             n = 1,
             part = getOption("timeperiodsR.parts"),
             include_current = F)
```

### Arguments

x	Date object
n	Number of periods for offset
part	Part of period you need to receive, one of "all", "start", "end", "sequence", "length". See details.
include_current	If TRUE including current period in sequence

### Details

You can get object of tpr class with all components or specify which component you need, use part for manage this option:

- all - get all components
- start - get only first date of period
- end - get only last date of period
- start - get vector of all dates in period
- length - get number of dates in period

### Value

Object of tpr class

### Author(s)

Alexey Seleznev

### See Also

For get next other periods see [last\\_n\\_months\(\)](#), [last\\_n\\_days\(\)](#), [last\\_n\\_weeks\(\)](#), [last\\_n\\_quarters\(\)](#)

**Examples**

```
## To get start, end and sequence of last 2 years,  
## exclude current year  
last2years <- last_n_years(n = 2)  
  
## include current year  
last2years_2 <- last_n_years(n = 2, include_current = TRUE)  
  
## To get vector of date sequences  
last_n_years(n = 2, part = "sequence")  
last_n_years(n = 2)$sequence  
seq(last2years)  
  
## Get number of days of last 2 years  
day_nums <- last_n_years(n = 2, part = "length")  
last_n_years()$length  
length(last2years)
```

---

last_weekend	<i>Get last weekend day in period.</i>
--------------	--

---

**Description**

Method for get last weekend in timeperiod.

**Usage**

```
last_weekend(x)
```

**Arguments**

x                    tpr object or date vector

**Value**

Vector of dates

**Author(s)**

Alexey Seleznev

**Examples**

```
pm <- previous_month()  
# get last weekend  
pm_last_weekend <- last_weekend(pm)
```

---

last_workday	<i>Get last workday day in period.</i>
--------------	--

---

**Description**

Method for get last workday in timeperiod.

**Usage**

```
last_workday(x)
```

**Arguments**

x                    tpr object or date vector

**Value**

Vector of dates

**Author(s)**

Alexey Seleznev

**Examples**

```
pm <- previous_month()
# get last workday
pm_last_workday <- last_workday(pm)
```

---

next_month	<i>Start and end of next month</i>
------------	------------------------------------

---

**Description**

Defines first and last date in next month and all dates in month

**Usage**

```
next_month(x = Sys.Date(),
           n = 1,
           part = getOption("timeperiodsR.parts"))
```

**Arguments**

x                    Date object  
n                    Number of periods for offset  
part                 Part of period you need to receive, one of "all", "start", "end", "sequence", "length".  
See details.

**Details**

You can get object of tpr class with all components or specify which component you need, use part for manage this option:

- all - get all components
- start - get only first date of period
- end - get only last date of period
- start - get vector of all dates in period
- length - get number of dates in period

**Value**

Object of tpr class

**Author(s)**

Alexey Seleznev

**See Also**

For get next other periods see [next\\_week\(\)](#), [next\\_quarter\(\)](#), [next\\_year\(\)](#)

**Examples**

```
## To get start, end and sequence of next month
nextmonth <- next_month()

## To get vector of date sequences
next_month(part = "sequence")
next_month()$sequence
seq(nextmonth)

## Get number of days of next months
day_nums <- next_month(part = "length")
next_month()$length
length(nextmonth)
```

---

next\_n\_days

*Start and end of next n days*

---

**Description**

Defines first and last date in next period

**Usage**

```
next_n_days(x = Sys.Date(),
            n = 1,
            part = getOption("timeperiodsR.parts"),
            include_current = F)
```

**Arguments**

x	Date object
n	Number of periods for offset
part	Part of period you need to receive, one of "all", "start", "end", "sequence", "length". See details.
include_current	If TRUE including current period in sequence

**Details**

You can get object of tpr class with all components or specify which component you need, use part for manage this option:

- all - get all components
- start - get only first date of period
- end - get only last date of period
- start - get vector of all dates in period
- length - get number of dates in period

**Value**

Object of tpr class

**Author(s)**

Alexey Seleznev

**See Also**

For get next other periods see [next\\_n\\_quarters\(\)](#), [next\\_n\\_months\(\)](#), [next\\_n\\_years\(\)](#), [next\\_n\\_weeks\(\)](#)

**Examples**

```
## To get start, end and sequence of next 10 days,
## exclude today
next10days <- next_n_days(n = 10)

## include today
next10days_2 <- next_n_days(n = 10, include_current = TRUE)

## To get vector of date sequences
```

```
next_n_days(n = 10, part = "sequence")
next_n_days(n = 10)$sequence
seq(next10days)
```

---

next_n_months	<i>Start and end of next n months</i>
---------------	---------------------------------------

---

### Description

Defines first and last date in next period

### Usage

```
next_n_months(x = Sys.Date(),
              n = 1,
              part = getOption("timeperiodsR.parts"),
              include_current = F)
```

### Arguments

x	Date object
n	Number of periods for offset
part	Part of period you need to receive, one of "all", "start", "end", "sequence", "length". See details.
include_current	If TRUE including current period in sequence

### Details

You can get object of tpr class with all components or specify which component you need, use part for manage this option:

- all - get all components
- start - get only first date of period
- end - get only last date of period
- start - get vector of all dates in period
- length - get number of dates in period

### Value

Object of tpr class

### Author(s)

Alexey Seleznev

**See Also**

For get next other periods see [next\\_n\\_quarters\(\)](#), [next\\_n\\_days\(\)](#), [next\\_n\\_years\(\)](#), [next\\_n\\_weeks\(\)](#)

**Examples**

```
## To get start, end and sequence of next 2 months,
## exclude current month
next2month <- next_n_months(n = 2)

## include current month
next2month_2 <- next_n_months(n = 2, include_current = TRUE)

## To get vector of date sequences
next_n_months(n = 2, part = "sequence")
next_n_months(n = 2)$sequence
seq(next2month)

## Get number of days of next 2 months
day_nums <- next_n_months(part = "length")
next_n_months()$length
length(next2month)
```

---

next_n_quarters	<i>Start and end of next n quarters</i>
-----------------	---

---

**Description**

Defines first and last date in next period

**Usage**

```
next_n_quarters(x = Sys.Date(),
               n = 1,
               part = getOption("timeperiodsR.parts"),
               include_current = F)
```

**Arguments**

x	Date object
n	Number of periods for offset
part	Part of period you need to receive, one of "all", "start", "end", "sequence", "length". See details.
include_current	If TRUE including current period in sequence

## Details

You can get object of tpr class with all components or specify which component you need, use part for manage this option:

- all - get all components
- start - get only first date of period
- end - get only last date of period
- start - get vector of all dates in period
- length - get number of dates in period

## Value

Object of tpr class

## Author(s)

Alexey Seleznev

## See Also

For get next other periods see [next\\_n\\_months\(\)](#), [next\\_n\\_days\(\)](#), [next\\_n\\_years\(\)](#), [next\\_n\\_weeks\(\)](#)

## Examples

```
## To get start, end and sequence of next 2 quarters,  
## exclude current quarter  
next2quarters <- next_n_quarters(n = 2)  
  
## include current quarter  
next2quarters_2 <- next_n_quarters(n = 2, include_current = TRUE)  
  
## To get vector of date sequences  
next_n_quarters(n = 2, part = "sequence")  
next_n_quarters(n = 2)$sequence  
seq(next2quarters)  
  
## Get number of days of next 2 quarters  
day_nums <- next_n_quarters(part = "length")  
next_n_quarters()$length  
length(next2quarters)
```

---

next_n_weeks	<i>Start and end of next n weeks</i>
--------------	--------------------------------------

---

**Description**

Defines first and last date in next period

**Usage**

```
next_n_weeks(x = Sys.Date(),
             n = 1,
             part = getOption("timeperiodsR.parts"),
             week_start = 1,
             include_current = F)
```

**Arguments**

x	Date object
n	Number of periods for offset
part	Part of period you need to receive, one of "all", "start", "end", "sequence", "length". See details.
week_start	Start of week, default = 1 is Monday, 7 is Sunday
include_current	If TRUE including current period in sequence

**Details**

You can get object of tpr class with all components or specify which component you need, use part for manage this option:

- all - get all components
- start - get only first date of period
- end - get only last date of period
- start - get vector of all dates in period
- length - get number of dates in period

**Value**

Object of tpr class

**Author(s)**

Alexey Seleznev

**See Also**

For get next other periods see [last\\_n\\_months\(\)](#), [last\\_n\\_days\(\)](#), [last\\_n\\_years\(\)](#), [last\\_n\\_quarters\(\)](#)

**Examples**

```
## To get start, end and sequence of next 2 weeks,
## exclude current week
next2weeks <- next_n_weeks(n = 2)

## include current week
next2weeks_2 <- next_n_weeks(n = 2, include_current = TRUE)

## Get next 2 week with start on Sunday
nextWeek <- next_n_weeks(n = 2, week_start = 7)

## To get vector of date sequences
next_n_weeks(n = 2, part = "sequence")
next_n_weeks(n = 2)$sequence
seq(next2weeks)

## Get number of days of next 2 weeks
day_nums <- next_n_weeks(part = "length")
next_n_weeks()$length
length(next2weeks)
```

---

next_n_years	<i>Start and end of next n weeks</i>
--------------	--------------------------------------

---

**Description**

Defines first and last date in next period

**Usage**

```
next_n_years(x = Sys.Date(),
             n = 1,
             part = getOption("timeperiodsR.parts"),
             include_current = F)
```

**Arguments**

x	Date object
n	Number of periods for offset
part	Part of period you need to receive, one of "all", "start", "end", "sequence", "length". See details.
include_current	If TRUE including current period in sequence

## Details

You can get object of tpr class with all components or specify which component you need, use part for manage this option:

- all - get all components
- start - get only first date of period
- end - get only last date of period
- start - get vector of all dates in period
- length - get number of dates in period

## Value

Object of tpr class

## Author(s)

Alexey Seleznev

## See Also

For get next other periods see [next\\_n\\_months\(\)](#), [next\\_n\\_days\(\)](#), [next\\_n\\_quarters\(\)](#), [next\\_n\\_weeks\(\)](#)

## Examples

```
## To get start, end and sequence of next 2 years,  
## exclude current year  
next2years <- next_n_years(n = 2)  
  
## include current year  
next2years_2 <- next_n_years(n = 2, include_current = TRUE)  
  
## To get vector of date sequences  
next_n_years(n = 2, part = "sequence")  
next_n_years(n = 2)$sequence  
seq(next2years)  
  
## Get number of days of next 2 years  
day_nums <- next_n_years(part = "length")  
next_n_years()$length  
length(next2years)
```

---

next_quarter	<i>Start and end of next quarter</i>
--------------	--------------------------------------

---

### Description

Defines first and last date in n next quarter

### Usage

```
next_quarter(x = Sys.Date(),  
            n = 1,  
            part = getOption("timeperiodsR.parts"))
```

### Arguments

x	Date object
n	Number of periods for offset
part	Part of period you need to receive, one of "all", "start", "end", "sequence", "length". See details.

### Details

You can get object of tpr class with all components or specify which component you need, use part for manage this option:

- all - get all components
- start - get only first date of period
- end - get only last date of period
- start - get vector of all dates in period
- length - get number of dates in period

### Value

Object of tpr class

### Author(s)

Alexey Seleznev

### See Also

For get next other periods see [next\\_week\(\)](#), [next\\_month\(\)](#), [next\\_year\(\)](#)

**Examples**

```
## To get start, end and sequence of next quarter
nextquarter <- next_quarter()

## To get vector of date sequences
next_quarter(part = "sequence")
next_quarter()$sequence
seq(nextquarter)

## Get number of days of next quarter
day_nums <- next_quarter(part = "length")
next_quarter()$length
length(nextquarter)
```

---

next_week	<i>Start and end of next week</i>
-----------	-----------------------------------

---

**Description**

Defines first and next date in n next week

**Usage**

```
next_week(x = Sys.Date(),
          n = 1,
          part = getOption("timeperiodsR.parts"),
          week_start = 1)
```

**Arguments**

x	Date object
n	Number of periods for offset
part	Part of period you need to receive, one of "all", "start", "end", "sequence", "length". See details.
week_start	Start of week, default = 1 is Monday, 7 is Sunday

**Details**

You can get object of tpr class with all components or specify which component you need, use part for manage this option:

- all - get all components
- start - get only first date of period
- end - get only next date of period
- start - get vector of all dates in period
- length - get number of dates in period

**Value**

Object of tpr class

**Author(s)**

Alexey Seleznev

**See Also**

For get next other periods see [next\\_month\(\)](#), [next\\_quarter\(\)](#), [next\\_year\(\)](#)

**Examples**

```
## To get start, end and sequence of next weeks
nextweek <- next_week()

## Get next week with start on Sunday
nextweeksun <- next_week(week_start = 7)

## To get vector of date sequences
next_week(part = "sequence")
next_week()$sequence
seq(nextweek)

## Get number of days of next 2 weeks
day_nums <- next_week(part = "length")
next_week()$length
length(nextweek)
```

---

next_year	<i>Start and end of next year</i>
-----------	-----------------------------------

---

**Description**

Defines first and last date in n next year

**Usage**

```
next_year(x = Sys.Date(),
          n = 1,
          part = getOption("timeperiodsR.parts"))
```

**Arguments**

x	Date object
n	Number of periods for offset
part	Part of period you need to receive, one of "all", "start", "end", "sequence", "length". See details.

**Details**

You can get object of tpr class with all components or specify which component you need, use part for manage this option:

- all - get all components
- start - get only first date of period
- end - get only last date of period
- start - get vector of all dates in period
- length - get number of dates in period

**Value**

Object of tpr class

**Author(s)**

Alexey Seleznev

**See Also**

For get next other periods see [next\\_month\(\)](#), [next\\_quarter\(\)](#), [next\\_week\(\)](#)

**Examples**

```
## To get start, end and sequence of next year
nextyear <- next_year()

## To get vector of date sequences
next_year(part = "sequence")
next_year()$sequence
seq(nextyear)

## Get number of days of next year
day_nums <- next_year(part = "length")
next_year()$length
length(nextyear)
```

---

previous_month	<i>Start and end of previous month</i>
----------------	--

---

**Description**

Defines first and last date in n previous month

**Usage**

```
previous_month(x = Sys.Date(),
               n = 1,
               part = getOption("timeperiodsR.parts"))
```

**Arguments**

x	Date object
n	Number of periods for offset
part	Part of period you need to receive, one of "all", "start", "end", "sequence", "length". See details.

**Details**

You can get object of tpr class with all components or specify which component you need, use part for manage this option:

- all - get all components
- start - get only first date of period
- end - get only last date of period
- start - get vector of all dates in period
- length - get number of dates in period

**Value**

Object of tpr class

**Author(s)**

Alexey Seleznev

**See Also**

For get next other periods see [previous\\_week\(\)](#), [previous\\_quarter\(\)](#), [previous\\_year\(\)](#)

**Examples**

```
## To get start, end and sequence of previous month
previousmonth <- previous_month()

## To get vector of date sequences
previous_month(part = "sequence")
previous_month()$sequence
seq(previousmonth)

## Get number of days of previous months
day_nums <- previous_month(part = "length")
previous_month()$length
length(previousmonth)
```

---

previous_quarter	<i>Start and end of previous quarter</i>
------------------	--

---

### Description

Defines first and last date in n previous quarter

### Usage

```
previous_quarter(x = Sys.Date(),  
                n = 1,  
                part = getOption("timeperiodsR.parts"))
```

### Arguments

x	Date object
n	Number of periods for offset
part	Part of period you need to receive, one of "all", "start", "end", "sequence", "length". See details.

### Details

You can get object of tpr class with all components or specify which component you need, use part for manage this option:

- all - get all components
- start - get only first date of period
- end - get only last date of period
- start - get vector of all dates in period
- length - get number of dates in period

### Value

Object of tpr class

### Author(s)

Alexey Seleznev

### See Also

For get next other periods see [previous\\_week\(\)](#), [previous\\_month\(\)](#), [previous\\_year\(\)](#)

**Examples**

```
## To get start, end and sequence of previous quarter
previousquarter <- previous_quarter()

## To get vector of date sequences
previous_quarter(part = "sequence")
previous_quarter()$sequence
seq(previousquarter)

## Get number of days of previous quarter
day_nums <- previous_quarter(part = "length")
previous_quarter()$length
length(previousquarter)
```

---

previous_week	<i>Start and end of previous week</i>
---------------	---------------------------------------

---

**Description**

Defines first and last date in n previous week

**Usage**

```
previous_week(x = Sys.Date(),
              n = 1,
              part = getOption("timeperiodsR.parts"),
              week_start = 1)
```

**Arguments**

x	Date object
n	Number of periods for offset
part	Part of period you need to receive, one of "all", "start", "end", "sequence", "length". See details.
week_start	Start of week, default = 1 is Monday, 7 is Sunday

**Details**

You can get object of tpr class with all components or specify which component you need, use part for manage this option:

- all - get all components
- start - get only first date of period
- end - get only last date of period
- start - get vector of all dates in period
- length - get number of dates in period

**Value**

Object of tpr class

**Author(s)**

Alexey Seleznev

**See Also**

For get next other periods see [previous\\_quarter\(\)](#), [previous\\_month\(\)](#), [previous\\_year\(\)](#)

**Examples**

```
## To get start, end and sequence of previous weeks
previousweek <- previous_week()

## Get previous week with start on Sunday
previousweeksun <- previous_week(week_start = 7)

## To get vector of date sequences
previous_week(part = "sequence")
previous_week()$sequence
seq(previousweek)

## Get number of days of previous 2 weeks
day_nums <- previous_week(part = "length")
previous_week()$length
length(previousweek)
```

---

previous_year	<i>Start and end of previous year</i>
---------------	---------------------------------------

---

**Description**

Defines first and last date in n previous year

**Usage**

```
previous_year(x = Sys.Date(),
              n = 1,
              part = getOption("timeperiodsR.parts"))
```

**Arguments**

x	Date object
n	Number of periods for offset
part	Part of period you need to receive, one of "all", "start", "end", "sequence", "length". See details.

## Details

You can get object of tpr class with all components or specify which component you need, use part for manage this option:

- all - get all components
- start - get only first date of period
- end - get only last date of period
- start - get vector of all dates in period
- length - get number of dates in period

## Value

Object of tpr class

## Author(s)

Alexey Seleznev

## See Also

For get next other periods see [previous\\_week\(\)](#), [previous\\_month\(\)](#), [previous\\_quarter\(\)](#)

## Examples

```
## To get start, end and sequence of previous year
previousyear <- previous_year()
```

```
## To get vector of date sequences
previous_year(part = "sequence")
previous_year()$sequence
seq(previousyear)
```

```
## Get number of days of previous year
day_nums <- previous_year(part = "length")
previous_year()$length
length(previousyear)
```

---

this_month	<i>Start and end of month</i>
------------	-------------------------------

---

## Description

Defines first and last date in month

## Usage

```
this_month(x = Sys.Date(),
           part = getOption("timeperiodsR.parts"))
```

**Arguments**

<code>x</code>	Date object
<code>part</code>	Part of period you need to receive, one of "all", "start", "end", "sequence", "length". See details.

**Details**

You can get object of tpr class with all components or specify which component you need, use `part` for manage this option:

- all - get all components
- start - get only first date of period
- end - get only last date of period
- start - get vector of all dates in period
- length - get number of dates in period

**Value**

Object of tpr class

**Author(s)**

Alexey Seleznev

**See Also**

For get next other periods see [this\\_week\(\)](#), [this\\_quarter\(\)](#), [this\\_year\(\)](#)

**Examples**

```
## To get start, end and sequence of this month
thismonth <- this_month()

## To get vector of date sequences
this_month(part = "sequence")
this_month()$sequence
seq(thismonth)

## Get number of days of this months
day_nums <- this_month(part = "length")
this_month()$length
length(thismonth)
```

---

this_quarter	<i>Start and end of quarter</i>
--------------	---------------------------------

---

### Description

Defines first and last date in quarter

### Usage

```
this_quarter(x = Sys.Date(),  
             part = getOption("timeperiodsR.parts"))
```

### Arguments

x	Date object
part	Part of period you need to receive, one of "all", "start", "end", "sequence", "length". See details.

### Details

You can get object of tpr class with all components or specify which component you need, use part for manage this option:

- all - get all components
- start - get only first date of period
- end - get only last date of period
- start - get vector of all dates in period
- length - get number of dates in period

### Value

Object of tpr class

### Author(s)

Alexey Seleznev

### See Also

For get next other periods see [this\\_week\(\)](#), [this\\_month\(\)](#), [this\\_year\(\)](#)

**Examples**

```
## To get start, end and sequence of this quarter
thisquarter <- this_quarter()

## To get vector of date sequences
this_quarter(part = "sequence")
this_quarter()$sequence
seq(thisquarter)

## Get number of days of this quarter
day_nums <- this_quarter(part = "length")
this_quarter()$length
length(thisquarter)
```

---

this_week	<i>Start and end of week</i>
-----------	------------------------------

---

**Description**

Defines first and last date in week

**Usage**

```
this_week(x = Sys.Date(),
          part = getOption("timeperiodsR.parts"),
          week_start = 1)
```

**Arguments**

x	Date object
part	Part of period you need to receive, one of "all", "start", "end", "sequence", "length". See details.
week_start	Start of week, default = 1 is Monday, 7 is Sunday

**Details**

You can get object of tpr class with all components or specify which component you need, use part for manage this option:

- all - get all components
- start - get only first date of period
- end - get only last date of period
- start - get vector of all dates in period
- length - get number of dates in period

**Value**

Object of tpr class

**Author(s)**

Alexey Seleznev

**See Also**

For get next other periods see [this\\_quarter\(\)](#), [this\\_month\(\)](#), [this\\_year\(\)](#)

**Examples**

```
## To get start, end and sequence of this weeks
thisweek <- this_week()

## Get this week with start on Sunday
thisweeksun <- this_week(week_start = 7)

## To get vector of date sequences
this_week(part = "sequence")
this_week()$sequence
seq(thisweek)

## Get number of days of this 2 weeks
day_nums <- this_week(part = "length")
this_week()$length
length(thisweek)
```

---

this_year	<i>Start and end of year</i>
-----------	------------------------------

---

**Description**

Defines first and last date in year

**Usage**

```
this_year(x = Sys.Date(),
          part = getOption("timeperiodsR.parts"))
```

**Arguments**

x	Date object
part	Part of period you need to receive, one of "all", "start", "end", "sequence", "length". See details.

## Details

You can get object of tpr class with all components or specify which component you need, use part for manage this option:

- all - get all components
- start - get only first date of period
- end - get only last date of period
- start - get vector of all dates in period
- length - get number of dates in period

## Value

Object of tpr class

## Author(s)

Alexey Seleznev

## See Also

For get next other periods see [this\\_week\(\)](#), [this\\_month\(\)](#), [this\\_year\(\)](#)

## Examples

```
## To get start, end and sequence of this year
thisyear <- this_year()
```

```
## To get vector of date sequences
this_year(part = "sequence")
this_year()$sequence
seq(thisyear)
```

```
## Get number of days of this year
day_nums <- this_year(part = "length")
this_year()$length
length(thisyear)
```

---

tpr\_operators

*Operators of tpr objects.*

---

## Description

Filtring timeperiods operators.

**Usage**

```
x %.in% y
x %left_in% y
x %left_out% y
x %right_in% y
x %right_out% y
```

**Arguments**

```
x          left Date or tpr object
y          Right tpr object.
```

**Value**

Date sequence or logical vector

**Author(s)**

Alexey Seleznev

**See Also**

For get more examples see vignette: `vignette("tpr_intro", package = "timeperiodsR")`

**Examples**

```
period1 <- this_month("2019-11-07")
period2 <- previous_week("2019-11-07")

period1
period1
period1
period1
```

---

weekends

*Get vector with weekends.*

---

**Description**

Method for get vector of weekends from timeperiod.

**Usage**

```
weekends(x)
```

**Arguments**

```
x          tpr object or date vector
```

**Value**

Vector of dates

**Author(s)**

Alexey Seleznev

**Examples**

```
pm <- previous_month()
# get weekends
pm_weekends <- weekends(pm)
```

---

weekends_length	<i>Number of weekends days in the period.</i>
-----------------	---

---

**Description**

Method for get number of weekdays days in period. Weekends is saturday and sunday.

**Usage**

```
weekends_length(x)
```

**Arguments**

x                    tpr object or date vector

**Value**

Integer

**Author(s)**

Alexey Seleznev

**Examples**

```
pm <- previous_month()
# weekends day number
pm_wd_len <- weekends_length(pm)
```

---

workdays	<i>Get vector with workdays.</i>
----------	----------------------------------

---

**Description**

Method for get vector of workdays from timeperiod.

**Usage**

```
workdays(x)
```

**Arguments**

x                    tpr object or date vector

**Value**

Vector of dates

**Author(s)**

Alexey Seleznev

**Examples**

```
pm <- previous_month()
# get workdays
pm_workdays <- workdays(pm)
```

---

workdays_length	<i>Number of workdays in the period.</i>
-----------------	--

---

**Description**

Method for get number of workdays in period. workdays is monday - friday.

**Usage**

```
workdays_length(x)
```

**Arguments**

x                    tpr object or date vector

**Value**

Integer

**Author(s)**

Alexey Seleznev

**Examples**

```
pm <- previous_month()
# workdays number
pm_wd_len <- workdays_length(pm)
```

# Index

`%.in%` (tpr\_operators), 39  
`%left_in%` (tpr\_operators), 39  
`%left_out%` (tpr\_operators), 39  
`%right_in%` (tpr\_operators), 39  
`%right_out%` (tpr\_operators), 39

`as_timeperiod`, 4

`check_dayoffs`, 5  
`custom_period`, 6  
`custom_period()`, 4

`first_weekend`, 8  
`first_workday`, 8

`last_n_days`, 9  
`last_n_days()`, 11, 12, 14, 15, 24  
`last_n_months`, 10  
`last_n_months()`, 10, 12, 14, 15, 24  
`last_n_quarters`, 12  
`last_n_quarters()`, 10, 11, 14, 15, 24  
`last_n_weeks`, 13  
`last_n_weeks()`, 10–12, 15  
`last_n_years`, 15  
`last_n_years()`, 10–12, 14, 24  
`last_weekend`, 16  
`last_workday`, 17

`next_month`, 17  
`next_month()`, 26, 28, 29  
`next_n_days`, 18  
`next_n_days()`, 21, 22, 25  
`next_n_months`, 20  
`next_n_months()`, 19, 22, 25  
`next_n_quarters`, 21  
`next_n_quarters()`, 19, 21, 25  
`next_n_weeks`, 23  
`next_n_weeks()`, 19, 21, 22, 25  
`next_n_years`, 24  
`next_n_years()`, 19, 21, 22  
`next_quarter`, 26  
`next_quarter()`, 18, 28, 29  
`next_week`, 27  
`next_week()`, 18, 26, 29  
`next_year`, 28  
`next_year()`, 18, 26, 28

`previous_month`, 29  
`previous_month()`, 31, 33, 34  
`previous_quarter`, 31  
`previous_quarter()`, 30, 33, 34  
`previous_week`, 32  
`previous_week()`, 30, 31, 34  
`previous_year`, 33  
`previous_year()`, 30, 31, 33

`this_month`, 34  
`this_month()`, 36, 38, 39  
`this_quarter`, 36  
`this_quarter()`, 35, 38  
`this_week`, 37  
`this_week()`, 35, 36, 39  
`this_year`, 38  
`this_year()`, 35, 36, 38, 39

`timeperiodsR` (timeperiodsR-package), 2  
`timeperiodsR-package`, 2  
`tpr_operators`, 39

`weekends`, 40  
`weekends_length`, 41  
`workdays`, 42  
`workdays_length`, 42