Package 'holideh'

May 8, 2026

Title Working with Canadian Dates

Version 0.1.0

Description Convenience date tools for identifying weekends, business days, and Canadian holidays, including R wrappers for the Canada Holidays API <https://canada-holidays.ca/>.

License MIT + file LICENSE

URL https://adamoshen.github.io/holideh/, https://github.com/adamoshen/holideh/

BugReports https://github.com/adamoshen/holideh/issues

Depends R (>= 4.1.0)

Imports cli, dplyr, httr2, lubridate, magrittr, purrr, rlang, stringr, tibble, tidyr, tidyselect, utf8

Suggests fansi, knitr, rmarkdown

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation no

Author Adam Shen [aut, cph, cre]

Maintainer Adam Shen <adamshen1@gmail.com>

Repository CRAN

Date/Publication 2026-04-09 09:00:31 UTC

Contents

`count_bizdays` *Count the number of business days between a range*

Description

Count the number of business days between a range of dates, inclusively.

Usage

```
count_bizdays(from, to, holidays, weekend = c("Sat", "Sun"))
```

Arguments

`from`	The beginning of the date range.
`to`	The end of the date range.
`holidays`	A vector of dates that are holidays.
`weekend`	A character vector of three-letter abbreviations of weekday names indicating days that should be considered a weekend. Acceptable values are: `"Sun"`, `"Mon"`, `"Tue"`, `"Wed"`, `"Thu"`, `"Fri"`, `"Sat"`, `"Sun"`.

Details

Holiday dates can be obtained using `get_holidays()`, `get_province()`, or by defining a custom vector of holidays.

Value

A single number.

See Also

`is_bizday()`, `is_holiday()`, `is_weekend()`

Examples

```
library(lubridate)

winter_holidays <- ymd(c("2025-12-25", "2025-12-26"))

count_bizdays(from = ymd("2025-12-20"), to = ymd("2025-12-31"), holidays = winter_holidays)
```

`get_holidays` *Get holidays from the Canada Holidays API*

Description

Get all holidays for a given year. Best used for obtaining a list of federal holidays.

Usage

```
get_holidays(year = NULL, federal = NULL, optional = NULL)
```

Arguments

`year`	The year for which holidays should be retrieved, between 2013 and 2038. The default, `NULL`, is equivalent to the current year.
`federal`	A boolean indicating whether only federal holidays should be retrieved. The default, `NULL`, is equivalent to `FALSE`.
`optional`	A boolean indicating whether optional (non-legislated) holidays should be retrieved. The default, `NULL`, is equivalent to `FALSE`.

Value

A tibble with columns:

date `<date>` The date when the holiday occurs.

observed_date `<date>` The date when the holiday is observed (celebrated). For example, if Christmas Day falls on a Sunday, it is observed (celebrated) on the proceeding Monday.

name_en `<chr>` The name of the holiday, in English.

name_fr `<chr>` The name of the holiday, in French.

federal `<lgl>` Whether the holiday is a federal holiday.

holiday_id `<int>` The id of the holiday.

provinces `<list>` A list of tibbles containing information on the provinces observing the holiday and source links.

See Also

`get_province()`, Canada Holidays API

Examples

```
if (interactive()) {
  get_holidays()
}
```

`get_province` *Get holidays for a province or territory from the Canada Holidays API*

Description

Get all holidays for a given year, for a province or territory. Best used to obtain holiday information for a specific province or territory.

Usage

```
get_province(province, year = NULL, optional = NULL)
```

Arguments

`province`	The two letter abbreviation for a province/territory (case-sensitive).
`year`	The year for which holidays should be retrieved, between 2013 and 2038. The default, `NULL`, is equivalent to the current year.
`optional`	A boolean indicating whether optional (non-legislated) holidays should be retrieved. The default, `NULL`, is equivalent to `FALSE`.

Details

Province and territory codes:

- Alberta: `"AB"`
- British Columbia: `"BC"`
- Manitoba: `"MB"`
- New Brunswick: `"NB"`
- Newfoundland and Labrador: `"NL"`
- Nova Scotia: `"NS"`
- Northwest Territories: `"NT"`
- Nunavut: `"NU"`
- Ontario: `"ON"`
- Prince Edward Island: `"PE"`
- Quebec: `"QC"`
- Saskatchewan: `"SK"`
- Yukon: `"YT"`

Value

A tibble with columns:

date <date> The date when the holiday occurs.

observed_date <date> The date when the holiday is observed (celebrated). For example, if Christmas Day falls on a Sunday, it is observed (celebrated) on the proceeding Monday.

name_en <chr> The name of the holiday, in English.

name_fr <chr> The name of the holiday, in French.

federal <lgl> Whether the holiday is a federal holiday.

holiday_id <int> The id of the holiday.

province_id <chr> The abbreviated province/territory code.

province_name_en <chr> The name of the province/territory, in English.

province_name_fr <chr> The name of the province/territory, in French.

source_info <list> A list containing a link to the information source.

See Also

get_holidays(), Canada Holidays API

Examples

```
if (interactive()) {
  get_province(province = "ON")
}
```

---

is_bizday *Detect business days*

Description

In a vector of dates, detect the business days (i.e. exclude holidays and weekends).

Usage

```
is_bizday(x, holidays, weekend = c("Sat", "Sun"))
```

Arguments

x	A vector of dates or date-times. If date-times are supplied, the date component will be extracted.
holidays	A vector of dates that are holidays.
weekend	A character vector of three-letter abbreviations of weekday names indicating days that should be considered a weekend. Acceptable values are: "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat".

Details

Holiday dates can be obtained using `get_holidays()`, `get_province()`, or by defining a custom vector of holidays.

Value

A logical vector of length equal to `x`.

See Also

`is_holiday()`, `is_weekend()`, `count_bizdays()`

Examples

```
library(lubridate)

dates <- seq.Date(from = ymd("2025-12-20"), to = ymd("2025-12-31"), by = "1 day")
winter_holidays <- ymd(c("2025-12-25", "2025-12-26"))

rlang::set_names(is_bizday(dates, holidays = winter_holidays), dates)
```

`off_day` *Detect non-business days*

Description

In a vector of dates, detect the non-business days (holiday or weekend).

Usage

```
is_holiday(x, holidays)

is_weekend(x, weekend = c("Sat", "Sun"))
```

Arguments

`x`	A vector of dates or date-times. If date-times are supplied, the date component will be extracted.
`holidays`	A vector of dates that are holidays.
`weekend`	A character vector of three-letter abbreviations of weekday names indicating days that should be considered a weekend. Acceptable values are: `"Sun"`, `"Mon"`, `"Tue"`, `"Wed"`, `"Thu"`, `"Fri"`, `"Sat"`.

Details

Holiday dates can be obtained using `get_holidays()`, `get_province()`, or by defining a custom vector of holidays.

Value

A logical vector of length equal to x.

See Also

is_bizday()

Examples

```
library(lubridate)

dates <- seq.Date(from = ymd("2025-12-20"), to = ymd("2025-12-31"), by = "1 day")
winter_holidays <- ymd(c("2025-12-25", "2025-12-26"))

rlang::set_names(is_holiday(dates, holidays = winter_holidays), dates)
rlang::set_names(is_weekend(dates), dates)
```

Index