

# Package: quantdates (via r-universe)

July 5, 2024

**Type** Package

**Title** Manipulate Dates for Finance

**Version** 2.0.4

**Maintainer** Juan Pablo Bermudez <juan.bermudez@quantil.com.co>

**Description** Functions to manipulate dates and count days for quantitative finance analysis. The 'quantdates' package considers leap, holidays and business days for relevant calendars in a financial context to simplify quantitative finance calculations, consistent with International Swaps and Derivatives Association (ISDA) (2006) <<https://www.isda.org/book/2006-isda-definitions/>> regulations.

**Encoding** UTF-8

**LazyData** true

**License** GPL-3

**BugReports** <https://github.com/quantilma/quantdates/issues>

**RoxygenNote** 7.3.1

**Depends** R (>= 2.10)

**Imports** lubridate (>= 1.7.4)

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**URL** <https://github.com/quantilma/quantdates>

**NeedsCompilation** no

**Author** Julian Chitiva [aut], Diego Jara [aut], Erick Translateur [com], Juan Pablo Bermudez [aut, cre], Quantil S.A.S [aut, cph]

**Repository** CRAN

**Date/Publication** 2024-07-04 07:20:02 UTC

## Contents

AddBusinessDays . . . . .	2
AddDate . . . . .	3
BusinessDays . . . . .	4
day_count . . . . .	5
difftime_business . . . . .	7
difftime_leap_year . . . . .	8
holiDaysBOG . . . . .	9
holiDaysLDN . . . . .	9
holiDaysNY . . . . .	10
holiDaysNYGB . . . . .	11
LastDayOfMonth . . . . .	11
NumExcel2DateR . . . . .	12
NumR2DateR . . . . .	13
wdBOG . . . . .	13
wdLDN . . . . .	14
wdNY . . . . .	15
wdNYGB . . . . .	15
<b>Index</b>	<b>17</b>

---

AddBusinessDays	<i>AddBusinessDays</i>
-----------------	------------------------

---

### Description

Function to add a number of business days to a specific date. Currently the function work for returning values between 1990 and 2100.

### Usage

```
AddBusinessDays(date = Sys.Date(), numDate, loc = "BOG")
```

### Arguments

date	Initial date, the default is set to the date returned by Sys.Date().
numDate	Number of dates to be add (positive or negative).
loc	String that determines the location for business days. See details.

### Details

loc refers to the location for business days:

- NY for New York Stock Exchange Market.
- NYGB for New York Government Bonds Market.
- LDN for London.

- NYLDN for the intersection of business days in New York Stock Exchange and London.
- NYGBLDN for the intersection of business days in New York Government Bond and London.
- BOG for Bogota.
- BOGNY for the intersection of business days in Bogota and New York Stock Exchange.
- BOGNYGB for the intersection of business days in Bogota and New York Government Bond.

### Value

The output is the final date after adding the number of business dates to the initial date. If the initial date is a non-working date, the result of the function for numDate equal to 0 or 1 is the same.

### Author(s)

Diego Jara and Juan Pablo Bermudez

### Examples

```
# Date input as Date object
AddBusinessDays(date = Sys.Date(), numDate = 15, loc = 'BOG')

# Date input as character object
AddBusinessDays(date = as.character(Sys.Date()), numDate = 15, loc = 'BOG')
```

---

AddDate

*AddDate*

---

### Description

Function to add a number of days, months and years to a specific date.

### Usage

```
AddDate(date = Sys.Date(), addDays = 0, addMonths = 0, addYears = 0)
```

### Arguments

date	Initial date.
addDays	If specified, vector number of days to add to the initial date.
addMonths	If specified, vector number of months to add to the initial date.
addYears	If specified, vector number of years to add to the initial date.

### Value

The output is a vector of final dates after adding the number of days, months and years to the initial date.

**Author(s)**

Julian Chitiva, Diego Jara and Juan Pablo Bermudez

**Examples**

```
# Date input as Date object
AddDate(date = Sys.Date(), addDays=14, addMonths=2, addYears=3)

# Date input as character object
AddDate(date = '2019-10-04', addDays=14, addMonths=2, addYears=3)

# Date vectors to add to initial date
AddDate(Sys.Date(), addDays = c(1,2), addMonths = c(4,5), addYears = c(5,6))
```

---

BusinessDays

*BusinessDays*

---

**Description**

Calculate business days for a given location. Currently, data availability goes from 1990 to 2100.

**Usage**

```
BusinessDays(loc = "BOG", from = NULL, to = NULL)
```

**Arguments**

loc	String that determines the location for business days. See details.
from	If provided returns available business dates after this date (inclusive).
to	If provided returns available business dates until this date (inclusive).

**Details**

loc refers to the location for business days:

- NYGB for New York Government Bonds Market.
- NY for New York Stock Exchange Market.
- LDN for London.
- NYLDN for the intersection of business days in New York Stock Exchange and London.
- NYGBLDN for the intersection of business days in New York Government Bond and London.
- BOG for Bogota.
- BOGNYGB for the intersection of business days in Bogota and New York Government Bond.
- BOGNY for the intersection of business days in Bogota and New York Stock Exchange.

**Value**

Vector of business days. Data availability from 1990 up to 2100.

**Author(s)**

Diego Jara, Julian Chitiva and Juan Pablo Bermudez

**Examples**

```
# Returns all business days available for the location
BusinessDays(loc='BOG')

# Returns business days within given range for the location and Dates as
# character
BusinessDays(loc='BOG', from='2020-10-10', to='2020-11-10')

# Returns business days within given range for the location and Dates as
# Dates
BusinessDays(loc='BOG', from=as.Date('2020-10-10'), to='2020-11-10')

# Returns all available business days for the location after given
# 'from' date as character
BusinessDays(loc='BOG', from='2020-10-10')
```

---

day\_count

*day\_count*

---

**Description**

Function to count the number of years between two dates according to the given convention.

**Usage**

```
day_count(tfinal, tinitial, convention = "ACT/365")
```

**Arguments**

- tfinal            Final date.
- tinitial         Initial date.
- convention      Character that specifies the convention. See details.

**Details**

The convention accepts the following values:

- 30/360.

$$DayCount = \frac{360 \times (Y_2 - Y_1) + 30 \times (M_2 - M_1) + (D_2 - D_1)}{360}$$

Here the dates are in the following format

- tfinal =  $Y_2$ - $M_2$ - $D_2$  (YYYY-MM-DD).
- tinitial =  $Y_1$ - $M_1$ - $D_1$  (YYYY-MM-DD).

It is important to note that

- $D_1 = \min(D_1, 30)$
- If  $D_1 = 30$  then  $D_2 = \min(D_2, 30)$

- ACT/365 (Default).

$$DayCount = \frac{Days(tinitial, tfinal)}{365}$$

Also known as ACT/365 Fixed.

- ACT/360.

$$DayCount = \frac{Days(tinitial, tfinal)}{360}$$

- ACT/365L.

$$DayCount = \frac{Days(tinitial, tfinal)}{DiY}$$

If February 29 is in the range from Date1 (exclusive) to Date2 (inclusive), then DiY = 366, else DiY = 365.

- NL/365.

If February 29 is not in the period then actual number of days between dates is used. Else actual number of days minus 1 is used. Day count basis = 365.

- ACT/ACT-ISDA.

$$DayCount = \frac{Days\ not\ in\ leap\ year}{365} + \frac{Days\ in\ leap\ year}{366}$$

- ACT/ACT-AFB.

$$DayCount = \frac{Days(tinitial, tfinal)}{DiY}$$

The basic rule is that if February 29 is in the range from Date1 (inclusive) to Date2 (exclusive), then DiY = 366, else DiY = 365.

If the period from Date1 to Date2 is more than one year, the calculation is split into two parts:

- The number of complete years, counted back from the last day of the period.
- The remaining initial stub, calculated using the basic rule.

**Value**

Number of years between the specified dates according to the convention.

**Author(s)**

Julian Chitiva

**Source**

International Swaps and Derivatives Association - ISDA.

**References**

International Swaps and Derivatives Association. (2006). 2006 ISDA definitions. New York, N.Y: International Swaps and Derivatives Association.

**Examples**

```
#Function accepts Dates as Dates or as characters.
day_count(tfinal='2023-03-08',tinitial='2019-02-28',convention='ACT/365')
day_count(tfinal=as.Date('2023-03-08'),tinitial=as.Date('2019-02-28'),convention='ACT/360')
day_count(tfinal='2023-03-08',tinitial=as.Date('2019-02-28'),convention='30/360')
day_count(tfinal='2023-03-08',tinitial='2019-02-28',convention='NL/365')
day_count(tfinal='2023-03-08',tinitial='2019-02-28',convention='ACT/ACT-ISDA')
day_count(tfinal='2023-03-08',tinitial='2019-02-28',convention='ACT/ACT-AFB')
```

---

difftime\_business      *difftime\_business*

---

**Description**

difftime\_business

**Usage**

```
difftime_business(tfinal, tinitial, wd = wdBOG)
```

**Arguments**

tfinal	Final date, it must be a business day.
tinitial	Initial date, it must be a business day.
wd	Vector of dates with business days. The default are the business days of Bogota. See details

**Details**

wd refers to the business days of a specific location:

- wdNYGB for New York Government Bonds Market.
- wdNY for New York Stock Exchange Market.
- wdLDN for London.
- wdBOG for Bogota.

**Value**

Number of days between the specified dates.

**Author(s)**

Diego Jara and Juan Pablo Bermudez

Function to count the number of business days between two dates.

**Examples**

```
#Function accepts Dates as Dates or as characters.
difftime_business(tfinal='2023-03-08',tinitial='2019-02-28',wd=wdBOG)
difftime_business(tfinal=as.Date('2023-03-08'),tinitial=as.Date('2019-02-28'),wd=wdBOG)
difftime_business(tfinal='2023-03-08',tinitial=as.Date('2019-02-28'),wd=wdLDN)
difftime_business(tfinal='2023-03-08',tinitial='2019-02-28',wd=wdNY)
difftime_business(tfinal='2023-03-08',tinitial='2019-02-28',wd=wdNYGB)
```

---

*difftime\_leap\_year*      *difftime\_leap\_year*

---

**Description**

Function to count the number of days between two dates. Optional parameters to count without the leap-days.

**Usage**

```
difftime_leap_year(tfinal, tinitial, leapDatesIn = TRUE)
```

**Arguments**

<code>tfinal</code>	Final date.
<code>tinitial</code>	Initial date.
<code>leapDatesIn</code>	If TRUE count leap Dates, else exclude from counting.

**Value**

Number of days between the specified dates.

**Author(s)**

Julian Chitiva and Diego Jara



**Examples**

```
#Function accepts Dates as Dates or as characters.  
difftime_leap_year(tfinal='2023-03-05',tinitial='2019-02-28',leapDatesIn=TRUE)  
difftime_leap_year(tfinal=as.Date('2023-03-05'),tinitial=as.Date('2019-02-28'),leapDatesIn=TRUE)  
difftime_leap_year(tfinal='2023-03-05',tinitial='2019-02-28',leapDatesIn=FALSE)  
difftime_leap_year(tfinal='2023-03-05',tinitial=as.Date('2019-02-28'),leapDatesIn=FALSE)
```

---

holiDaysBOG	<i>Bogota holidays dates.</i>
-------------	-------------------------------

---

**Description**

Bogota (Colombia) holidays dates. The holidays were created using the package bizdays. Dates range between 1990-01-01 and 2100-12-08.

**holiDaysBOG** Vector of dates of Bogota holidays

**Usage**

```
holiDaysBOG
```

**Format**

Vector of dates.

**Author(s)**

Quantil S.A.S

**Source**

Author Calculations

---

holiDaysLDN	<i>London holidays dates.</i>
-------------	-------------------------------

---

**Description**

London (England) holidays dates. The holidays were created using the package bizdays. Dates range between 1990-01-01 and 2100-12-28.

**holiDaysLDN** Vector of dates of London holidays

**Usage**

```
holiDaysLDN
```

**Format**

Vector of dates.

**Author(s)**

Quantil S.A.S

**Source**

Author Calculations

---

holiDaysNY

*New York Stock Exchange holidays dates.*

---

**Description**

New York Stock Exchange -United States holidays dates. The holidays were created using the package bizdays. Dates range between 1990-01-01 and 2100-12-24.

**holiDaysNY** Vector of dates of New York Stock Exchange holidays

**Usage**

holiDaysNY

**Format**

Vector of dates.

**Author(s)**

Quantil S.A.S

**Source**

Author Calculations

---

holiDaysNYGB	<i>New York Government Bonds holidays dates.</i>
--------------	--

---

**Description**

New York Government Bonds -United States holidays dates. The holidays were created using the package bizdays. Dates range between 1990-01-01 and 2100-12-24.

**holiDaysNY** Vector of dates of New York Government Bonds holidays

**Usage**

```
holiDaysNYGB
```

**Format**

Vector of dates.

**Author(s)**

Quantil S.A.S

**Source**

Author Calculations

---

LastDayOfMonth	<i>LastDayOfMonth</i>
----------------	-----------------------

---

**Description**

Returns the last day of a month.

**Usage**

```
LastDayOfMonth(year, month, date = NULL)
```

**Arguments**

year	Year as a number.
month	Month as a number.
date	If provided, uses year and month from this date. It could be date or a string format date YYYY-MM-DD.

**Value**

Last day of the month in the current year.

**Author(s)**

Diego Jara

**Examples**

```
# Return last day of the month in year
LastDayOfMonth(year = 2020, month = 2)

# Return last day of the month for the date
LastDayOfMonth(date = '2020-02-03')
```

---

NumExcel2DateR

*NumExcel2DateR*

---

**Description**

Takes a date represented by a number in Excel format (origin="1899-12-30") and returns a date in R format.

**Usage**

```
NumExcel2DateR(date)
```

**Arguments**

date            numeric vector.

**Value**

date in R.

**Author(s)**

Diego Jara

**See Also**

For dates with R origin.  
Other Number to Date: [NumR2DateR\(\)](#)

**Examples**

```
NumExcel2DateR(as.numeric(Sys.Date()))
```

---

`NumR2DateR`*NumR2DateR*

---

**Description**

Takes a date represented by a number in R format (origin="1970-01-01") and returns a date.

**Usage**

```
NumR2DateR(date)
```

**Arguments**

date                    numeric vector.

**Value**

date in R.

**Author(s)**

Diego Jara

**See Also**

For dates with Excel origin.

Other Number to Date: [NumExcel2DateR\(\)](#)

**Examples**

```
NumR2DateR(as.numeric(Sys.Date()))
```

---

`wdBOG`*Bogota business dates.*

---

**Description**

Bogota (Colombia) business dates. Dates range between 1990-01-02 and 2100-12-31.

**wdBOG** Vector of dates of Bogota business days

**Usage**

```
wdBOG
```

**Format**

Vector of dates.

**Author(s)**

Quantil S.A.S

**Source**

Author Calculations

---

wdLDN

*London business dates.*

---

**Description**

London (England) business dates. Dates range between 1990-01-02 and 2100-12-31.

**wdLDN** Vector of dates of London business days

**Usage**

wdLDN

**Format**

Vector of dates.

**Author(s)**

Quantil S.A.S

**Source**

Author Calculations

---

wdNY *New York Stock Exchange business dates.*

---

**Description**

New York (United States) Stock Exchange business dates. Dates range between 1990-01-02 and 2100-12-31.

**wdNY** Vector of dates of New York Stock Exchange (NYSE) business days

**Usage**

wdNY

**Format**

Vector of dates.

**Author(s)**

Quantil S.A.S

**Source**

Author Calculations

---

wdNYGB *New York Government Bonds business dates.*

---

**Description**

New York (United States) Government Bonds business dates. Dates range between 1990-01-02 and 2100-12-31.

**wdNY** Vector of dates of New York Government Bonds (NYGB) business days

**Usage**

wdNYGB

**Format**

Vector of dates.

**Author(s)**

Quantil S.A.S

**Source**

Author Calculations



# Index

## \* **Number to Date**

NumExcel2DateR, [12](#)

NumR2DateR, [13](#)

## \* **datasets**

holiDaysBOG, [9](#)

holiDaysLDN, [9](#)

holiDaysNY, [10](#)

holiDaysNYGB, [11](#)

wdBOG, [13](#)

wdLDN, [14](#)

wdNY, [15](#)

wdNYGB, [15](#)

AddBusinessDays, [2](#)

AddDate, [3](#)

BusinessDays, [4](#)

day\_count, [5](#)

difftime\_business, [7](#)

difftime\_leap\_year, [8](#)

holiDaysBOG, [9](#)

holiDaysLDN, [9](#)

holiDaysNY, [10](#)

holiDaysNYGB, [11](#)

LastDayOfMonth, [11](#)

NumExcel2DateR, [12](#), [13](#)

NumR2DateR, [12](#), [13](#)

wdBOG, [13](#)

wdLDN, [14](#)

wdNY, [15](#)

wdNYGB, [15](#)