Package ‘quantdates’

October 13, 2022

Type Package
Title Manipulate Dates for Finance
Version 1.0
Maintainer Julian Chitiva <julian.chitiva@quantil.com.co>
Encoding UTF-8
LazyData true
License GPL-3
BugReports https://github.com/quantilma/quantdates/issues
RoxygenNote 7.1.0
Depends R (>= 2.10)
Imports lubridate (>= 1.7.4)
Suggests knitr, rmarkdown
VignetteBuilder knitr
NeedsCompilation no
Author Julian Chitiva [aut, cre],
  Diego Jara [aut],
  Erick Transluateur [com],
  Quantil S.A.S [aut, cph]
Repository CRAN
Date/Publication 2020-06-09 14:30:02 UTC

R topics documented:

AddBusinessDays .................................................. 2
AddDate ........................................................... 3
AddBusinessDays

Description

Function to add a number of business days to a specific date. Currently the function works for returning values between 2000 and 2030.

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 added (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.
- LDN for London.
- NYLDN for the intersection of business days in New York and London.
- BOG for Bogota.
- BOGNY for the intersection of business days in Bogota and New York.

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.
AddDate

Author(s)

Diego Jara

Examples

# Date input as Date object
AddDate(date = Sys.Date(), addDays = 15, loc = 'BOG')

# Date input as character object
AddDate(date = as.character(Sys.Date()), addDays = 15, loc = 'BOG')

Description

Function to add a number of days, months and years to a specific date. The length of addDays, addMonths and addYears must be the same.

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 the final date after adding the number of days, months and years to the initial date.

Author(s)

Julian Chitiva and Diego Jara

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)
**Description**

Calculate business days for a given location. Data availability depends on the location.

**Usage**

```r
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:

- NY for New York.
- LDN for London.
- NYLDN for the intersection of business days in New York and London.
- BOG for Bogota.
- BOGNY for the intersection of business days in Bogota and New York.

**Value**

Vector of business days. Data availability depends on the location.

**Author(s)**

Diego Jara and Julian Chitiva

**Examples**

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

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

- # Returns business days within given range for the location and Dates as
  # Dates
  ```r
  BusinessDays(loc = 'BOG', from = as.Date('2020-10-10'), to = '2020-11-10')
  ```
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.
  \[
  \text{DayCount} = \frac{360 \times (Y_2 - Y_1) + 30 \times (M_2 - M_1) + (D_2 - D_1)}{360}
  \]
  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 = \text{min}(D_1, 30)
  - If D_1 = 30 then D_2 = \text{min}(D_2, 30)

- ACT/365 (Default).
  \[
  \text{DayCount} = \frac{\text{Days(tinitial, tfinal)}}{365}
  \]
  Also known as ACT/365 Fixed.

- ACT/360.
  \[
  \text{DayCount} = \frac{\text{Days(tinitial, tfinal)}}{365}
  \]
• ACT/365L.

\[
\text{DayCount} = \frac{\text{Days(tinitial}, t_{\text{final}})}{\text{DiY}}
\]

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.

\[
\text{DayCount} = \frac{\text{Days not in leap year}}{365} + \frac{\text{Days in leap year}}{366}
\]

• ACT/ACT-AFB.

\[
\text{DayCount} = \frac{\text{Days(tinitial}, t_{\text{final}})}{\text{DiY}}
\]

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


Examples

#Function accepts Dates as Dates or as characters.
\begin{align*}
day\_count(t\_{\text{final}}=\text{'2023-03-08'}, t\_{\text{initial}}=\text{'2019-02-28'}, \text{convention='ACT/365'}) \\
day\_count(t\_{\text{final}}=\text{as.Date('2023-03-08')}, t\_{\text{initial}}=\text{as.Date('2019-02-28')}, \text{convention='ACT/360'}) \\
day\_count(t\_{\text{final}}=\text{'2023-03-08'}, t\_{\text{initial}}=\text{as.Date('2019-02-28')}, \text{convention='30/360'}) \\
day\_count(t\_{\text{final}}=\text{'2023-03-08'}, t\_{\text{initial}}=\text{'2019-02-28'}, \text{convention='NL/365'}) \\
day\_count(t\_{\text{final}}=\text{'2023-03-08'}, t\_{\text{initial}}=\text{'2019-02-28'}, \text{convention='ACT/ACT-ISDA'}) \\
day\_count(t\_{\text{final}}=\text{'2023-03-08'}, t\_{\text{initial}}=\text{'2019-02-28'}, \text{convention='ACT/ACT-AFB'})
\end{align*}
**diffitime_business**

Description

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

Usage

diffitime_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.

Value

Number of days between the specified dates.

Author(s)

Diego Jara

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

Examples

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

**diffitime_leap_year**

Description

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

Usage

diffitime_leap_year(tfinal, tinitial, leapDatesIn = TRUE)
Arguments

- `tfinal` Final date.
- `tinitial` Initial date.
- `leapDatesIn` 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

Bogota holidays dates.

Description

Bogota (Colombia) holidays dates. The holidays were created using the package timeDate. Dates range between 2011-01-10 and 2050-12-08.

holiDaysBOG Vector of dates of Bogota holidays

Usage

holiDaysBOG

Format

Vector of dates.

Author(s)

Quantil S.A.S

Source

Author Calculations
holiDaysLDN  

London holidays dates.

Description

London(England) holidays dates. The holidays were created using the package timeDate. Dates range between 1900-04-13 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 holidays dates.

Description

New York-United States holidays dates. The holidays were created using the package timeDate. Dates range between 1900-01-01 and 2100-12-24.

holiDaysNY  Vector of dates of New York holidays

Usage

holiDaysNY

Format

Vector of dates.

Author(s)

Quantil S.A.S
LastDayOfMonth

Source

Author Calculations

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**

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**

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()))

Description
Bogota (Colombia) business dates. Dates range between 1998-01-02 and 2030-12-31.

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**  

**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 business dates.*

**Description**  
New York (United States) business dates. Dates range between 2000-01-03 and 2030-12-31.

**wdNY**  
Vector of dates of New York business days

**Usage**  
wdNY

**Format**  
Vector of dates.

**Author(s)**  
Quantil S.A.S

**Source**  
Author Calculations
Index

* Number to Date  
  NumExcel2DateR, 11  
  NumR2DateR, 11

* datasets  
  holiDaysBOG, 8  
  holiDaysLDN, 9  
  holiDaysNY, 9  
  wdBOG, 12  
  wdLDN, 13  
  wdNY, 13

AddBusinessDays, 2  
AddDate, 3  
BusinessDays, 4  
day_count, 5  
difftime_business, 7  
difftime_leap_year, 7

holiDaysBOG, 8  
holiDaysLDN, 9  
holiDaysNY, 9  

LastDayOfMonth, 10  
NumExcel2DateR, 11, 12  
NumR2DateR, 11, 11

wdBOG, 12  
wdLDN, 13  
wdNY, 13