

Package ‘jubilee’

July 22, 2025

Type Package

Title Forecasting Long-Term Growth of the U.S. Stock Market and Business Cycles

Version 0.3.3

Date 2020-01-24

Author Stephen H-T. Lihn [aut, cre]

Maintainer Stephen H-T. Lihn <stevelihn@gmail.com>

Description A long-term forecast model called “Jubilee-Tectonic model” is implemented to forecast future returns of the U.S. stock market, Treasury yield, and gold price. The five-factor model forecasts the 10-year and 20-year future equity returns with high R-squared above 80 percent. It is based on linear growth and mean reversion characteristics in the U.S. stock market. This model also enhances the CAPE model by introducing the hypothesis that there are fault lines in the historical CAPE, which can be calibrated and corrected through statistical learning. In addition, it contains a module for business cycles, optimal interest rate, and recession forecasts.

URL <https://ssrn.com/abstract=3156574>

<https://ssrn.com/abstract=3422278>

<https://ssrn.com/abstract=3435667>

Depends R (>= 3.5.0)

Imports stats, yaml, utils, xts, zoo, splines, parallel, graphics, methods, readxl (>= 1.3.1), data.table, dplyr

Suggests knitr, tinytex, R.rsp, testthat, roxygen2, scales, shape

VignetteBuilder R.rsp

License Artistic-2.0

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Collate 'daily2fraction-method.R' 'fraction2daily-method.R'
'jubilee-adj-fault-line-method.R' 'jubilee-calc-cape-method.R'
'jubilee-package.R' 'jubilee-class.R' 'jubilee-constructor.R'

'jubilee-qty-ols-method.R' 'jubilee-forward-rtn-method.R'
 'jubilee-fred-data-method.R' 'jubilee-locate-file.R'
 'jubilee-macro-cost-method.R' 'jubilee-macro-fit-method.R'
 'jubilee-macro-predict-method.R' 'jubilee-mcsapply-method.R'
 'jubilee-ols-method.R' 'jubilee-optimal-tb3ms-method.R'
 'jubilee-predict-method.R' 'jubilee-read-fred-file.R'
 'jubilee-repo-class.R' 'jubilee-repo-config.R'
 'jubilee-repo-constructor.R' 'jubilee-std-fault-line-method.R'
 'jubilee-yield-inversion-method.R' 'tri-wave-class.R'
 'tri-wave-constructor.R' 'tri-wave-model.R'

NeedsCompilation no

Repository CRAN

Date/Publication 2020-01-24 16:40:02 UTC

Contents

jubilee-package	3
daily2fraction	3
fraction2daily	4
jubilee	4
jubilee-class	5
jubilee.adj_fault_line	6
jubilee.calc_ape	7
jubilee.qty_ols	7
jubilee.forward_rtn	8
jubilee.fred_data	9
jubilee.locate_file	10
jubilee.macro_cost	11
jubilee.macro_fit	11
jubilee.macro_predict	12
jubilee.mcsapply	13
jubilee.ols	14
jubilee.optimal_tb3ms	15
jubilee.predict	16
jubilee.read_fred_file	16
jubilee.repo	17
jubilee.repo-class	18
jubilee.repo.config	19
jubilee.std_fault_line	19
jubilee.yield_inversion	20
tri.wave	21
tri.wave class	21
triangle	22

Index

24

jubilee-package	<i>jubilee: A package to forecast long-term growth of the US stock market and business cycles</i>
-----------------	---

Description

The jubilee package provides the core class and functions to forecast long-term growth of the U.S. stock market. It also contains a module for business cycles, optimal interest rate, and recession forecasts. A tutorial is provided to demonstrate how to use this package and explain the relation between the mathematical notations and the functions and data columns in this package.

Author(s)

Stephen H-T. Lihn

References

Stephen H.T. Lihn, "Jubilee Tectonic Model: Forecasting Long-Term Growth and Mean Reversion in the U.S. Stock Market." Available at SSRN: <https://ssrn.com/abstract=3156574> or via DOI: <http://dx.doi.org/10.2139/ssrn.3156574>

Stephen H.T. Lihn, "Business Cycles, Optimal Interest Rate, and Recession Forecast From Yield Curve, Unemployment, GDP, and Payrolls." Available at SSRN: <https://ssrn.com/abstract=3422278>

daily2fraction	<i>Converter from daily Date to fraction</i>
----------------	--

Description

Utility to convert from daily Date (R's Date object) to fraction.

Usage

```
daily2fraction(d)
```

Arguments

d	array of Date object, or string in ISO yyyy-mm-dd format
---	--

Value

numeric, year in fraction convention

Author(s)

Stephen H. Lihn

Examples

```
daily2fraction(as.Date("2017-01-15")) # 2017.038
daily2fraction(as.Date("2017-02-14")) # 2017.122
daily2fraction(as.Date("2017-07-15")) # 2017.538
```

<code>fraction2daily</code>	<i>Converter from fraction to daily Date</i>
-----------------------------	--

Description

Utility to convert from fraction to daily Date (R's Date object).

Usage

```
fraction2daily(fraction)
```

Arguments

`fraction` numeric, representing year in fraction convention.

Value

array of Date object

Author(s)

Stephen H. Lihn

Examples

```
fraction2daily(2017.038) # 2017-01-15
fraction2daily(2017.125) # 2017-02-15
```

<code>jubilee</code>	<i>Constructor of the jubilee class</i>
----------------------	---

Description

Construct an jubilee object which holds raw and derived data, channel regression results, and other derived analytical quantities. This object is the main object to perform various forecasts and analyses.

Usage

```
jubilee(dtb, lookback.channel = 45, fwd.rtn.duration = 20,
        force = TRUE)
```

Arguments

<code>dtb</code>	data.table from the <code>jubilee.repo</code> object, typically it is the <code>ie</code> slot. The user is allowed to provide custom data object to research different markets, as long as the column names are compliant.
<code>lookback.channel</code>	numeric, look-back channel in years to calculate mean-reversion. Default is 45.
<code>fwd.rtn.duration</code>	numeric, forward return duration in years. Default is 20.
<code>force</code>	logical, if FALSE, allowed to retrieve previous object stored in option. Default is TRUE.

Value

an object of the jubilee class

Author(s)

Stephen H. Lihn

Examples

```
## Not run:
repo <- jubilee.repo(online=FALSE)
ju <- jubilee(repo@ie, 45, 20)

## End(Not run)
```

jubilee-class

The jubilee class

Description

This S4 class stores raw and derived data, channel regression settings and results.

Slots

`call` the match.call slot.

`lookback.channel` numeric, the look-back channel in years.

`fwd.rtn.duration` numeric, the forward return duration in years.

`reg.dtb` data.table, contains the regression data.

dtb data.table, contains the consolidated market data.

rate.spread.mean numeric, the mean of the yield spread, used to calculate rate.spread.norm column.

create.time POSIXct, records the creation time of this object.

jubilee.adj_fault_line

Adjust the time series by fault lines

Description

This utility is used to adjust the time series by the provided fault lines.

Usage

```
jubilee.adj_fault_line(fraction, ts, fl, months = 1)
```

Arguments

<code>fraction</code>	numeric, representing year in fraction convention.
<code>ts</code>	numeric, time series to be adjusted, typically it is <code>log.cape10</code> or <code>log.cape20</code> .
<code>fl</code>	the fault line matrix. See <code>jubilee.std_fault_line()</code> for more detail. If it is provided as character string, it will be looked up as the name of data set in the standard fault line library. If it is provided as numeric array, it will be converted to a matrix.
<code>months</code>	interval in months to ramp up the fault line. Default is 1.

Value

numeric, ts adjusted by fault lines

Author(s)

Stephen H. Lihn

Examples

```
## Not run:
repo <- jubilee.repo(online=FALSE)
dj <- jubilee(repo@ie, 45, 10)@reg.dtb
dj$log.cape10.adj <- jubilee.adj_fault_line(dj$fraction, dj$log.cape10, "r_nom_f10_5ftr_4fl")

## End(Not run)
```

jubilee.calc_cape	<i>Internal utility to calculate n-year CAPE</i>
-------------------	--

Description

This CAPE calculator replicates the methodology of Shiller, so that one can calculate n-year CAPE, e.g. n=20. This utility has been calibrated by original 10-year CAPE data from Shiller.

Usage

```
jubilee.calc_cape(dtb, period, tol.frac = 1/6)
```

Arguments

dtb	data.table
period	numeric, the backward-looking regression period
tol.frac	numeric, tolerance of missing data in the beginning of the time series, expressed as fraction. Default is 1/6, that is, two months.

Value

numeric, the same length as dtb\$fraction.

Author(s)

Stephen H. Lihn

Examples

```
## Not run:
dtb <- jubilee.repo(online=FALSE)@ie
cape10 <- jubilee.calc_cape(dtb, 10)
cape20 <- jubilee.calc_cape(dtb, 20)

## End(Not run)
```

jubilee.eqty_ols	<i>Internal utility to calculate OLS regression for log total return index</i>
------------------	--

Description

Calculate the OLS regression for log total return index

Usage

```
jubilee.eqty_ols(dtb, end.frac, lookback.channel, tol.frac = 1/6)
```

Arguments

<code>dtb</code>	data.table that contains <code>fraction</code> and <code>log.tri</code> columns.
<code>end.frac</code>	numeric, the ending fraction of regression.
<code>lookback.channel</code>	numeric, the backward-looking regression period
<code>tol.frac</code>	numeric, tolerance of missing data in the beginning, expressed as fraction. Default is 1/6, that is, two months.

Value

two-element array `c(a,R)` if `end.frac` is length-one; data.table with `end.frac` as fraction column if `end.frac` is an array.

Author(s)

Stephen H. Lihn

Examples

```
## Not run:
dtb <- jubilee.repo(online=FALSE)@ie
jubilee.eqty_ols(dtb, 1970, 50) # c(11.8671626, 0.1008371)

## End(Not run)
```

<code>jubilee.forward_rtn</code>	<i>Internal utility to calculate annualized forward and backward (log) return</i>
----------------------------------	---

Description

These two internal utilities are intended to be used to calculate the annualized forward and backward log-return on the given time series. It is really calculating the speed of change, aka log-return, expecting the input to be in logarithmic scale. The forward return is typically the response variable in a forecast. The backward return is often used as explanatory variable in a regression.

Usage

```
jubilee.forward_rtn(fraction, ts, fwd.rtn.duration, tol.frac = 1/12)

jubilee.backward_rtn(fraction, ts, bwd.rtn.duration, tol.frac = 1/12)
```

Arguments

<code>fraction</code>	numeric, the ending fraction of regression
<code>ts</code>	numeric, the time series data, typically in log-scale
<code>fwd.rtn.duration</code>	numeric, the forward-looking regression period
<code>tol.frac</code>	numeric, tolerance of missing data in the beginning of backward return, or the ending of the forward return, expressed as fraction. Default is 1/12, that is, one month.
<code>bwd.rtn.duration</code>	numeric, the backward-looking regression period

Value

numeric, the same length as `fraction`

Author(s)

Stephen H. Lihn

Examples

```
## Not run:
dtb <- jubilee.repo(online=FALSE)@ie
dtb$fwd.logr.10 <- jubilee.forward_rtn(dtb$fraction, dtb$log.tri, 10)
dtb$bwd.logr.10 <- jubilee.backward_rtn(dtb$fraction, dtb$log.tri, 10)
head(subset(dtb, fraction >= 1990),1)$fwd.logr.10 # 1/1990+10y: 0.16745
tail(subset(dtb, fraction <= 2000+1/12),1)$bwd.logr.10 # the same as above

## End(Not run)
```

<code>jubilee.fred_data</code>	<i>Internal utility to download time series data from FRED</i>
--------------------------------	--

Description

This utility downloads time series from FRED. Many time series that this package uses are available on FRED. Therefore, this utility is used to provide daily or monthly updates by concatenating live data to the internal static data.

Usage

```
jubilee.fred_data(symbol, col_out = "Close", retry = 3)
```

Arguments

<code>symbol</code>	character, the name of the time series
<code>col_out</code>	character, the name of the output closing price column. Default is "Close"
<code>retry</code>	numeric, number of retries on the URL. Default is 3.

Value

The xts object for the time series

Examples

```
## Not run:  
  jubilee.fred_data("VIXCLS") # VIX  
  
## End(Not run)
```

`jubilee.locate_file` *Internal utility to locate static file*

Description

This utility returns the path to internal file

Usage

```
jubilee.locate_file(local_file, stop = TRUE)
```

Arguments

<code>local_file</code>	character, the file name of an internal file.
<code>stop</code>	logical, whether to stop if file can't be located. Default is TRUE.

Value

The path to the file, or else, an empty string

Author(s)

Stephen H. Lihn

Examples

```
jubilee.locate_file("UNRATE.csv")
```

jubilee.macro_cost	<i>Calculate the cost function of the macro model</i>
--------------------	---

Description

This utility calculates the cost function of the macro model according to the squared error sum with penalty parameter. This utility can be used to experiment more sophisticated optimization schemes.

Usage

```
jubilee.macro_cost(dtb, rs, penalty = c(1, 1, 1), new.tb3ms = NA,
  new.gs10 = NA)
```

Arguments

dtb	data table, usually this is the reg.dtb of the jubilee object
rs	the list returned from jubilee.macro_fit
penalty	numeric, the penalty vector for the 6 models. Default is c(1, 1, 1).
new.tb3ms	numeric, vector of new rate.tb3ms with length equal to NROW of dtb. Default is NA.
new.gs10	numeric, vector of new rate.gs10 with length equal to NROW of dtb. Default is NA.

Value

The data table containing the "macro.cost" column

Author(s)

Stephen H. Lihn

jubilee.macro_fit	<i>The GUPTY macro model</i>
-------------------	------------------------------

Description

This utility contains the macro regression models, covering GUPTY: three types of GDP, UNRATE (unemployment rate), Payroll, and Treasury yield curve. TCU (total capacity utilization) is also covered in the model but less recommended. Given the in-sample time periods, it will perform model regressions and return a list storing relevant information about the result. The purpose of this method is to automate the regression and facilitate programmatic cross validation.

Usage

```
jubilee.macro_fit(dtb, N, K, unrate.frac.start, gdp.frac.start, frac.end,
  cv.frac.end)
```

Arguments

dtb	data table, usually this is the reg.dtb of the jubilee object
N	numeric, number of years for GDP log-return calculation in GDP models
K	numeric, number of years for GDP log-return calculation in Payroll and TCU models
unrate.frac.start	numeric, starting fraction of unrate regression time period
gdp.frac.start	numeric, starting fraction of gdp regression time period
frac.end	numeric, ending fraction of regression time period. This is also the starting fraction of cross-validation.
cv.frac.end	numeric, ending fraction of cross-validation time period. Cross validation can be disabled by setting it to NA.

Value

The list of data elements and their attributes.

Author(s)

Stephen H. Lihn

References

Stephen H.T. Lihn, "Business Cycles, Optimal Interest Rate, and Recession Forecast From Yield Curve, Unemployment, GDP, and Payrolls." Available at SSRN: <https://ssrn.com/abstract=3422278>

Examples

```
## Not run:
repo <- jubilee.repo()
ju <- jubilee(repo@ie, 45, 20)
N <- 4
K <- 1.5
rs <- jubilee.macro_fit(ju@reg.dtb, N, K, 1950, 1960, 2010, 2019)

## End(Not run)
```

jubilee.macro_predict *Prediction from UNRATE and GDP models*

Description

This utility performs the prediction from the linear models of UNRATE and GDP. The purpose of this method is to automate the prediction and to allow users experimenting optimization on the natural rate of interest.

Usage

```
jubilee.macro_predict(dtb, rs, new.tb3ms = NA, new.gs10 = NA)
```

Arguments

<code>dtb</code>	data table, usually this is <code>lm.dtb</code> of the <code>rs</code> object, with GDP log-return percent (<code>logrp.N</code> , <code>logrp.K</code>) calculated.
<code>rs</code>	the list returned from <code>jubilee.macro_fit</code> , which provides regression parameters for the prediction (not the data).
<code>new.tb3ms</code>	numeric, vector of new rate.tb3ms with length equal to <code>NROW</code> of <code>dtb</code> . Default is <code>NA</code> .
<code>new.gs10</code>	numeric, vector of new rate.gs10 with length equal to <code>NROW</code> of <code>dtb</code> . Default is <code>NA</code> .

Value

The data table containing the predictions and all the required input columns

Author(s)

Stephen H. Lihn

<code>jubilee.mcsapply</code>	<i>Wrapper to calculate <code>sapply</code> using multi-core</i>
-------------------------------	--

Description

This utility calculates `sapply` using multi-core capability. It is a simple wrapper on `simplify2array` and `parallel::mclapply`. It is particularly convenient on Linux and Mac when parallelism saves significant amount of computing time.

Usage

```
jubilee.mcsapply(x, FUN, ...)
```

Arguments

<code>x</code>	numeric
<code>FUN</code>	the function to be applied to each element of <code>x</code>
<code>...</code>	optional arguments to <code>FUN</code>

Value

numeric

Author(s)

Stephen H. Lihn

Examples

```
a <- seq(1,100)
jubilee.mcsapply(a, function(x) x^2) # use multi-core!
```

jubilee.ols

Internal utility to calculate OLS regression

Description

Calculate the OLS regression for a given time series and fraction

Usage

```
jubilee.ols(fraction, ts, lookback.channel, tol.frac = 1/6)
```

Arguments

<code>fraction</code>	numeric, the ending fraction of regression
<code>ts</code>	numeric, the time series data
<code>lookback.channel</code>	numeric, the backward-looking regression period
<code>tol.frac</code>	numeric, tolerance of missing data in the beginning, expressed as fraction. Default is 1/6, that is, two months.

Value

data.table with columns of `fraction`, `lm.a`, `lm.y`, `lm.r`

Author(s)

Stephen H. Lihn

References

See Section 2.3 of Stephen H.T. Lihn, "Jubilee Tectonic Model: Forecasting Long-Term Growth and Mean Reversion in the U.S. Stock Market." Available at <http://dx.doi.org/10.2139/ssrn.3156574>

Examples

```
## Not run:
dtb <- jubilee.repo(online=FALSE)@ie
df <- jubilee.ols(dtb$fraction, dtb$log.tri, 50)
subset(df, fraction > 1970 & fraction < 1970.05)
# fraction    lm.a      lm.r      lm.y
# 1970.042 11.86401 0.1007617 0.02103105

## End(Not run)
```

jubilee.optimal_tb3ms *Calculate the optimal TB3MS*

Description

This utility calculates the optimal TB3MS using the analytic solution.

Usage

```
jubilee.optimal_tb3ms(dtb, rs, penalty = c(1, 1, 1))
```

Arguments

dtb	data table, usually this is <code>lm.dtb</code> of the <code>rs</code> object, with GDP log-return percent (<code>logrp.N</code> , <code>logrp.K</code>) calculated.
rs	the list returned from <code>jubilee.macro_fit</code> .
penalty	numeric, the penalty vector for the models. Default is <code>c(1, 1, 1)</code> .

Value

The data table containing the "optimal.tb3ms" column

Author(s)

Stephen H. Lihn

jubilee.predict	<i>Make prediction based on linear regression</i>
-----------------	---

Description

Make prediction based on the linear regression of the forward return. Refer to the tutorial for more detail.

Usage

```
jubilee.predict(object, lm, data)
```

```
jubilee.predict_real(object, lm, data)
```

Arguments

object	object of jubilee class
lm	the linear model
data	data used to predict (similar to newdata of stats::predict)

Value

data.table containing the prediction

Author(s)

Stephen H. Lihn

References

See Section 7 of Stephen H.T. Lihn, "Jubilee Tectonic Model: Forecasting Long-Term Growth and Mean Reversion in the U.S. Stock Market." Available at <http://dx.doi.org/10.2139/ssrn.3156574>

jubilee.read_fred_file	<i>Internal utility to read FRED file</i>
------------------------	---

Description

This utility reads the internal static file, optionally amends with FRED online data, and returns the values of a given symbol.

Usage

```
jubilee.read_fred_file(fraction, local_file, symbol, online = FALSE,
  daily_symbol = NULL, period = "M")
```

Arguments

<code>fraction</code>	numeric, the fraction to return the value. The utility will lookup within a month to find value. For debug purpose, set it to NULL, and the intermediate data table will be returned.
<code>local_file</code>	character, the file name of an internal file. For debug purpose, set it to NULL, and the process will initiate the source data from FRED via symbol, instead of a local file.
<code>symbol</code>	character, the FRED symbol.
<code>online</code>	logical, whether to fetch online data from FRED. Default is FALSE.
<code>daily_symbol</code>	character, the FRED symbol to read daily data that supplements the monthly data. Default is NULL.
<code>period</code>	character, length-1 string indicating the data period of the symbol. M is monthly, Q is quarterly. Default is M.

Value

The values of the symbol, numeric with the same length as fraction.

Author(s)

Stephen H. Lihn

Examples

```
## Not run:
repo <- jubilee.repo(online=FALSE)
a <- jubilee.read_fred_file(repo@ie$fraction, "BAA.csv", "BAA")
tail(a)

## End(Not run)
```

jubilee.repo	<i>Constructor of jubilee.repo class</i>
--------------	--

Description

Construct a `jubilee.repo` class by combining data from that of Robert Shiller since 1871, historical stock market data from 1802 to 1987 by William Schwert, 3-month Treasury bill rate, gold price, and several other economic time series from FRED. Optionally, this function can fetch more recent data from the website of Robert Shiller and Federal Reserve FRED website if the R session has connection to the internet.

Usage

```
jubilee.repo(online = TRUE, force = TRUE)
```

Arguments

online	logical, indicating whether to fetch data from online resource or not. Default is TRUE.
force	logical, if FALSE, allowed to retrieve previous object stored in option. The FALSE setting overrides the online=TRUE setting. Default is TRUE.

Value

An object of jubilee.repo class

Author(s)

Stephen H. Lihn

Examples

```
## Not run:
repo <- jubilee.repo(online=FALSE)
dtb <- repo@ie
tail(dtb,1)

## End(Not run)
```

jubilee.repo-class	<i>The jubilee repository class</i>
--------------------	-------------------------------------

Description

This S4 class stores the raw data for the jubilee package

Slots

call	The match.call slot
ie	data.table, contains the combined data from ie.raw, ws, and inflation.
yield.inversion	numeric, the fractions of yield curve inversion
raw.ie	data.table, contains the data from ie_data.xls of Robert Shiller
ws	data.table, contains the historical market return data from William Schwert
inflation	data.table, contains the historical inflation data from Minneapolis FED
comm.int	data.table, contains the historical commercial interest rate
tb3ms	data.table, contains the historical 3-month Treasury bill rate

gold data.table, contains the historical monthly gold prices
 gold2 data.table, contains the historical annual gold prices
 create.time POSIXct, records the creation time of this object.

jubilee.repo.config *Configuration of jubilee's data repository*

Description

This utility stores the data configuration for the jubilee's data repository. This is used internally to provide proper abstraction to the data sources, such as file name, URL, FRED symbol, column name, decimal format, etc.

Usage

```
jubilee.repo.config()
```

Value

The list of data elements and their attributes.

Author(s)

Stephen H. Lihn

Examples

```
c <- jubilee.repo.config()
c$ie$url
```

jubilee.std_fault_line *Standard fault line data sets*

Description

This method defines a collection of standard fault line data sets that have been analyzed and optimized in the research. It is intended for end users to produce standard regressions, forecasts, and charts quickly.

Usage

```
jubilee.std_fault_line(name)
```

Arguments

name character, the name of the collection. If "list" is supplied, the list of names will be returned. If a numeric array is supplied, it will be converted to a matrix format.

Value

numeric, pairs of fault lines, each is c(year, delta)

Author(s)

Stephen H. Lihn

Examples

```
jubilee.std_fault_line("r_nom_f10_5ftr_4f1")
jubilee.std_fault_line("r_nom_f20_5ftr_2f1")
jubilee.std_fault_line("r_nom_f20_5ftr_2f1_ramp5y")
```

jubilee.yield_inversion

List of dates for yield curve inversion

Description

List of dates for yield curve inversion, generally compliant to the dating of business cycles after WWII in the U.S.. This data is also stored in the `yield.inversion` slot in the `jubilee.repo` object.

Usage

```
jubilee.yield_inversion()
```

Value

numeric, in the unit of fraction.

Author(s)

Stephen H. Lihn

Examples

```
jubilee.yield_inversion()
```

tri.wave	<i>Constructor of tri.wave class</i>
----------	--------------------------------------

Description

Construct an `tri.wave` object to simulate the triangular wave model.

Usage

```
tri.wave()
```

Value

an object of `tri.wave` class

Author(s)

Stephen H. Lihn

Examples

```
w <- tri.wave()
```

tri.wave class	<i>The triangular wave model class</i>
----------------	--

Description

This S4 class defines the parameters in the triangular wave model.

Slots

`call` the `match.call` slot.

`a.t` numeric, the look-back channel in years

`a0` numeric, the look-back channel in years

`s1` numeric, the forward return duration in years

`s2` numeric, the start fraction of in-sample training period

`y.mean` numeric, the end fraction of in-sample training period

`y.amp` numeric, the end fraction of in-sample training period

`y.t` numeric, the end fraction of in-sample training period

`y.p` numeric, the end fraction of in-sample training period

References

See Section 4 of Stephen H.T. Lihn, "Jubilee Tectonic Model: Forecasting Long-Term Growth and Mean Reversion in the U.S. Stock Market." Available at <http://dx.doi.org/10.2139/ssrn.3156574>

triangle

Methods of triangular wave model

Description

Methods of triangular wave model

Usage

```
triangle(t, p)
tri.wave.s(object, t)
tri.wave.a(object, t)
tri.wave.y(object, t)
tri.wave.x(object, t)
tri.wave.logr.y(object, t, p)
tri.wave.logr(object, t, p)
tri.wave.logr.semi(object, t)
tri.wave.logr.quarter(object, t)
```

Arguments

t	the time vector in fraction
p	the period of the triangle wave
object	the object of tri.wave class

Value

numeric

Author(s)

Stephen H. Lihn

References

See Section 4 of Stephen H.T. Lihn, "Jubilee Tectonic Model: Forecasting Long-Term Growth and Mean Reversion in the U.S. Stock Market." Available at <http://dx.doi.org/10.2139/ssrn.3156574>

Examples

```
w <- tri.wave()
t <- seq(1900, 2000, by=1)
tri.wave.y(w, t)
```

Index

- * **class**
 - jubilee-class, 5
 - jubilee.repo-class, 18
 - tri.wave class, 21
- * **constructor**
 - jubilee, 4
 - jubilee.repo, 17
 - jubilee.yield_inversion, 20
 - tri.wave, 21
- * **data**
 - jubilee.fred_data, 9
 - jubilee.locate_file, 10
 - jubilee.macro_cost, 11
 - jubilee.macro_fit, 11
 - jubilee.macro_predict, 12
 - jubilee.optimal_tb3ms, 15
 - jubilee.read_fred_file, 16
 - jubilee.repo.config, 19
- * **model**
 - jubilee.adj_fault_line, 6
 - jubilee.predict, 16
 - jubilee.std_fault_line, 19
 - triangle, 22
- * **utility**
 - daily2fraction, 3
 - fraction2daily, 4
 - jubilee.calc_cape, 7
 - jubilee.eqty_ols, 7
 - jubilee.forward_rtn, 8
 - jubilee.mcsapply, 13
 - jubilee.ols, 14
 - jubilee.backward_rtn
 (jubilee.forward_rtn), 8
 - jubilee.calc_cape, 7
 - jubilee.eqty_ols, 7
 - jubilee.forward_rtn, 8
 - jubilee.fred_data, 9
 - jubilee.locate_file, 10
 - jubilee.macro_cost, 11
 - jubilee.macro_fit, 11
 - jubilee.macro_predict, 12
 - jubilee.mcsapply, 13
 - jubilee.ols, 14
 - jubilee.optimal_tb3ms, 15
 - jubilee.predict, 16
 - jubilee.predict_real (jubilee.predict),
 16
 - jubilee.read_fred_file, 16
 - jubilee.repo, 17
 - jubilee.repo-class, 18
 - jubilee.repo.config, 19
 - jubilee.std_fault_line, 19
 - jubilee.yield_inversion, 20
 - tri.wave, 21
 - tri.wave class, 21
 - tri.wave.a (triangle), 22
 - tri.wave.logr (triangle), 22
 - tri.wave.s (triangle), 22
 - tri.wave.x (triangle), 22
 - tri.wave.y (triangle), 22
 - triangle, 22

daily2fraction, 3

fraction2daily, 4

jubilee, 4

jubilee-class, 5

jubilee-package, 3

jubilee.adj_fault_line, 6