Package ‘RTL’

February 23, 2020

Type Package
Title Risk Tool Library
Version 0.1.1
Date 2020-02-16
Description Collection of functions and metadata to complement core packages in Finance and Commodities, including futures expiry tables and <http://www.morningstarcommodity.com/> API functions. See <https://github.com/risktoollib/RTL>.
Depends R (>= 3.6)
License GPL (>= 3)
Encoding UTF-8
LazyData true

URL https://github.com/risktoollib/RTL
Suggests testthat (>= 2.1.0)
RoxygenNote 7.0.2

Imports zoo, xts, stats, magrittr, tibble, dplyr, tidyr, ggplot2, httr, stringr, purrr, lubridate, EIAdata, timetk, PerformanceAnalytics, tibbletime, quantmod, forecast, tidyquant, readr, Quandl, fGarch, fitdistrplus, tsibble, feasts, plotly, fabletools

NeedsCompilation no
Author Philippe Cote [aut, cre], Nima Safaian [aut]
Maintainer Philippe Cote <pcote@ualberta.ca>
Repository CRAN
Date/Publication 2020-02-23 18:50:02 UTC
R topics documented:

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>bond</td>
<td>3</td>
</tr>
<tr>
<td>cancrudeassays</td>
<td>3</td>
</tr>
<tr>
<td>cancrudeprices</td>
<td>4</td>
</tr>
<tr>
<td>chart_fwd_curves</td>
<td>4</td>
</tr>
<tr>
<td>chart_PerfSummary</td>
<td>5</td>
</tr>
<tr>
<td>chart_zscore</td>
<td>6</td>
</tr>
<tr>
<td>CRRReuro</td>
<td>7</td>
</tr>
<tr>
<td>dflong</td>
<td>8</td>
</tr>
<tr>
<td>dfwide</td>
<td>8</td>
</tr>
<tr>
<td>df_fut</td>
<td>8</td>
</tr>
<tr>
<td>distdescplot</td>
<td>9</td>
</tr>
<tr>
<td>eia2tidy</td>
<td>9</td>
</tr>
<tr>
<td>expiry_table</td>
<td>10</td>
</tr>
<tr>
<td>fitOU</td>
<td>10</td>
</tr>
<tr>
<td>garch</td>
<td>11</td>
</tr>
<tr>
<td>getIRswapCurve</td>
<td>12</td>
</tr>
<tr>
<td>getPrice</td>
<td>13</td>
</tr>
<tr>
<td>getPrices</td>
<td>14</td>
</tr>
<tr>
<td>holidaysOil</td>
<td>15</td>
</tr>
<tr>
<td>ir_df_us</td>
<td>16</td>
</tr>
<tr>
<td>ng_storage</td>
<td>16</td>
</tr>
<tr>
<td>npv</td>
<td>17</td>
</tr>
<tr>
<td>promptBeta</td>
<td>18</td>
</tr>
<tr>
<td>returns</td>
<td>19</td>
</tr>
<tr>
<td>rolladjust</td>
<td>19</td>
</tr>
<tr>
<td>simGBM</td>
<td>20</td>
</tr>
<tr>
<td>simOU</td>
<td>21</td>
</tr>
<tr>
<td>simOUJ</td>
<td>22</td>
</tr>
<tr>
<td>stl_decomp</td>
<td>23</td>
</tr>
<tr>
<td>swapCOM</td>
<td>24</td>
</tr>
<tr>
<td>swapIRS</td>
<td>25</td>
</tr>
<tr>
<td>tickers_eia</td>
<td>26</td>
</tr>
<tr>
<td>tradeCycle</td>
<td>26</td>
</tr>
<tr>
<td>tradeStats</td>
<td>27</td>
</tr>
<tr>
<td>twoott</td>
<td>27</td>
</tr>
<tr>
<td>twtrump</td>
<td>28</td>
</tr>
<tr>
<td>usSwapCurves</td>
<td>28</td>
</tr>
<tr>
<td>usSwapCurvesPar</td>
<td>28</td>
</tr>
<tr>
<td>usSwapIR</td>
<td>29</td>
</tr>
<tr>
<td>usSwapIRdef</td>
<td>29</td>
</tr>
</tbody>
</table>

Index            30
bond

Description

Compute bond price, cash flow table and duration

Usage

bond(ytm = 0.05, C = 0.05, T2M = 1, m = 2, output = "price")

Arguments

ytm Yield to Maturity
C Coupon rate per annum
T2M Time to maturity in years
m Periods per year for coupon payments e.g semi-annual = 2.
output "price", "df" or "duration"

Value

Price, cash flows data frame and/or duration

Author(s)

Philippe Cote

Examples

bond(ytm = 0.05, C = 0.05, T2M = 1, m = 2, output = "price")
bond(ytm = 0.05, C = 0.05, T2M = 1, m = 2, output = "df")
bond(ytm = 0.05, C = 0.05, T2M = 1, m = 2, output = "duration")

cancrudeassays
cancrudeassays
cancrudeassays

description

Data set with historical Canadian Crude Assays.

usage
cancrudeassays
**chart_fwd_curves**

**Format**

data frame

**Source**

https://crudemonitor.ca/

---

**cancrudeprices**  
cancrudeprices

---

**Description**

Randomized dataset of Canadian Crude monthly prices versus WTi Calendar Month Average.

**Usage**

cancrudeprices

**Format**

data frame

---

**chart_fwd_curves**  
chart_fwd_curves

---

**Description**

Returns a plot of forward curves through time

**Usage**

chart_fwd_curves(df = dfwide, cmdty = "cmewti", weekly = FALSE, ...)

**Arguments**

- df  
  Wide dataframe with date column and multiple series columns (multivariate)
- cmdty  
  Futures contract code in expiry_table object: unique(expiry_table$cmdty)
- weekly
  TRUE if you want weekly forward curves
- ...
  other graphical parameters

**Value**

plot of forward curves through time
Author(s)

Philippe Cote

Examples

```r
## Not run:
chart_fwd_curves(df=dfwide,cmdty="cmewti",weekly=TRUE,
main="WTI Forward Curves",ylab="\$ per bbl",xlab="",cex=2)
## End(Not run)
```

Description

Multi Asset Display of Cumulative Performance and Drawdowns

Usage

```r
chart_PerfSummary(
  ret = ret,
  geometric = TRUE,
  main = "Cumulative Returns and Drawdowns",
  linesize = 1.25
)
```

Arguments

- `ret` Wide dataframe univariate or multivariate of percentage returns.
- `geometric` Use geometric returns `TRUE` or `FALSE`.
- `main` Chart title.
- `linesize` Size of lines in chart and legend.

Value

Cumulative performance and drawdown charts.

Author(s)

Philippe Cote

Examples

```r
df <- dflong %>% dplyr::filter(series %in% c("CL01","CL12","CL36"))
ret <- returns(df=df,retType="rel",period.return=1,spread=TRUE)
ret <- data.frame(rolladjust(x=ret,commodityname=c("cmewti"),rolltype=c("Last.Trade")))
chart_PerfSummary(ret=ret, geometric=TRUE, main="Cumulative Returns and Drawdowns",linesize=1.25)
```
Description

Supports analytics and display of seasonal data. Z-Score is computed on residuals conditional on their seasonal period. Beware that most seasonal charts in industry e.g. (NG Storage) is not detrended so results once you apply an STL decomposition will vary from the unadjusted seasonal plot.

Usage

chart_zscore(
  df = df,
  title = "NG Storage Z Score",
  per = "yearweek",
  output = "zscore",
  chart = "seasons"
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>df</td>
<td>Long data frame with columns series, date and value</td>
</tr>
<tr>
<td>title</td>
<td>Your chart title</td>
</tr>
<tr>
<td>per</td>
<td>Frequency of seasonality &quot;yearweek&quot; (DEFAULT). &quot;yearmonth&quot;, &quot;yearquarter&quot;</td>
</tr>
<tr>
<td>output</td>
<td>&quot;stl&quot; for STL decomposition chart, &quot;stats&quot; for STL statistical test results. &quot;zs-core&quot; for residuals Z-score, &quot;seasonal&quot; for standard seasonal chart.</td>
</tr>
<tr>
<td>chart</td>
<td>&quot;seasons&quot; for feasts::gg_season() (DEFAULT) &quot;series&quot; for feasts::gg_subseries()</td>
</tr>
</tbody>
</table>

Value

Time series of STL decomposition residuals Z-Scores, or standard seasonal chart with feast package.

Author(s)

Philippe Cote

Examples

chart_zscore(df = ng_storage, title = "NG Storage Z Score",
  per = "yearweek", output = "stl", chart = "seasons")
chart_zscore(df = ng_storage, title = "NG Storage Z Score",
  per = "yearweek", output = "stats", chart = "seasons")
chart_zscore(df = ng_storage, title = "NG Storage Z Score",
  per = "yearweek", output = "zscore", chart = "seasons")
chart_zscore(df = ng_storage, title = "NG Storage Z Score",
  per = "yearweek", output = "seasonal", chart = "seasons")
Description

European option binomial model on a stock without dividends. For academic purpose only. Use fOptions::CRRBinomialTreeOptions for real-life usage.

Usage

CRReuro(S, X, sigma, r, T2M, N, type)

Arguments

S    Stock price.
X    Strike price.
sigma    Implied volatility e.g. 0.20
r    Risk-free rate.
T2M    Time to maturity in years
N    Number of time steps. Internally dt = T2M/N.
type    "call" or "put"

Value

List of asset price tree, option value tree and option price.

Author(s)

Philippe Cote

Examples

CRReuro(S=100, X=100, sigma=0.2, r=0.1, T2M=1, N=5, type="call")
df_long

Description
Futures settlement data set.

Usage
dflong

Format
data frame #' @source http://www.morningstarcommodity.com

---

df_wide

Description
Futures settlement data set.

Usage
dfwide

Format
data frame #' @source http://www.morningstarcommodity.com

---

df_fut

Description
Futures settlement data set.

Usage
df_fut

Format
data frame #' @source http://www.morningstarcommodity.com
distdescplot

Description

Provides a summary of returns distribution

Usage

distdescplot(x = x)

Arguments

x Wide dataframe with date column and single series (univariate).

Value

Multiple plots describing the distribution.

Author(s)

Philippe Cote

Examples

x <- dflong %>% dplyr::filter(series=="CL01")
x <- returns(df=x,retType="rel",period.return=1,spread=TRUE)
x <- rolladjust(x=x,commodityname=c("cmewti"),rolltype=c("Last.Trade"))
distdescplot(x=x)

eia2tidy

Description

Converts output of getEAI() in a tidy tibble with names("date","value"). Makes a clean wrapper for use with purrr.

Usage

eia2tidy(ticker, key)

Arguments

ticker EIA series name.
key EIA API token.
**Value**

A tibble object

**Author(s)**

Philippe Cote

**Examples**

```r
## Not run:
eia_df <- tibble::tribble(~ticker, ~series,
"PET.W_EPC0_SAX_YCUOK_MBBL.W", "Cushing Crude Stocks",
"NG.NW2_EPG0_SWO_R48_BCF.W","NG Storage - Lower 48") %>%
dplyr::mutate(key = EIAkey) %>%
dplyr::mutate(df = purrr::pmap(list(ticker,key),.f=eia2tidy)) %>%
dplyr::select(series, df) %>% tidyr::unnest()
## End(Not run)
```

**Description**

This dataframe provides detailed information on major futures contracts specifications pertaining to last settlement, notices and delivery dates. It also provides tickers in some data service.

**Usage**

expiry_table

**Format**

data frame

---

**fitOU**

**Description**

Parameter estimation for Ornstein–Uhlenbeck process

**Usage**

fitOU(spread)
Arguments
spread Spread time series.

Value
List of alpha, mu and sigma estimates

Author(s)
Philippe Cote

Examples
spread <- simOU(mu=5,theta=.5,sigma=0.2,T=5,dt=1/250)
fitOU(spread)

Description
Computes annualised Garch(1,1) volatilities using fGarch package.

Usage
garch(x = x, out = TRUE)

Arguments
x Wide dataframe with date column and single series (univariate).
out "chart" to return chart, "data" to return data or "fit" for garch fit output

Value
plot.xts object or xts series

Author(s)
Philippe Cote

Examples
x <- dflong %>% dplyr::filter(series=="CL01")
x <- returns(df=x,retType="rel",period.return=1,spread=TRUE)
x <- rolladjust(x=x,commodityname=c("cmewti"),rolltype=c("Last.Trade"))
summary(garch(x=x,out="fit"))
garch(x=x,out="chart")
garch(x=x,out="data")
getIRswapCurve

getIRswapCurve getIRswapCurve

description

Extract historical data for tsQuotes in RQuantlib to bootstrap swap curve using Morningstar and FRED as data source.

Usage

getIRswapCurve(
  currency = "USD",
  from = "2019-01-01",
  iuser = "x@xyz.com",
  ipassword = "pass"
)

Arguments

currency Currently only USD LIBOR implemented.
from From date as character string
iuser Morningstar user name as character - sourced locally in examples.
ipassword Morningstar user password as character - sourced locally in examples.

Value

wide data frame

Author(s)

Philippe Cote

Examples

## Not run:
getIRswapCurve(currency="USD", from="2019-08-26",iuser = username, ipassword = password)

## End(Not run)
getPrice

Description

Returns data from Morningstar API. See below for current feeds supported. You need your own credentials with Morningstar. In examples sourced locally.

Usage

```r
getPrice(
  feed = "CME_NymexFutures_EOD",
  contract = "CL9Z",
  from = "2019-01-01",
  iuser = "x@xyz.com",
  ipassword = "pass"
)
```

Arguments

- **feed**: Morningstar Feed Table.
- **contract**: Morningstar key.
- **from**: From date as character string
- **iuser**: Morningstar user name as character - sourced locally in examples.
- **ipassword**: Morningstar user password as character - sourced locally in examples.

Value

wide data frame

Current Feeds Supported

- CME_CbotFuturesEOD and CME_CbotFuturesEOD_continuous
- CME_NymexFutures_EOD and CME_NymexFutures_EOD_continuous
- CME_CmeFutures_EOD and CME_CmeFutures_EOD_continuous
- ICE_EuroFutures and ICE_EuroFutures_continuous
- ICE_NybotCoffeeSugarCocoaFutures and ICE_NybotCoffeeSugarCocoaFutures_continuous
- CME_STLCPC_Futures
- CFTC_CommitmentsOfTradersCombined. Requires multiple keys. Separate them by a space e.g. "N10 06765A NYME 01".
- Morningstar_FX_Forwards. Requires multiple keys. Separate them by a space e.g. "USD-CAD 2M".
getPrices

Author(s)
Philippe Cote

Examples

```r
## Not run:
getPrice(feed="CME_NymexFutures_EOD",contract="CL9Z",
from="2019-08-26",iuser = username, ipassword = password)
getPrice(feed="CME_NymexFutures_EOD_continuous",contract="CL_006_Month",
from="2019-08-26",iuser = username, ipassword = password)
getPrice(feed="CME_CbotFuturesEOD",contract="C9Z",
from="2019-08-26",iuser = username, ipassword = password)
getPrice(feed="CME_CbotFuturesEOD_continuous",contract="ZB_001_Month",
from="2019-08-26",iuser = username, ipassword = password)
getPrice(feed="CME_CmeFutures_EOD_continuous",contract="HE_006_Month",
from="2019-08-26",iuser = username, ipassword = password)
getPrice(feed="Morningstar_FX_Forwards",contract="USDCAD 2M",
from="2019-08-26",iuser = username, ipassword = password)
getPrice(feed="CME_CmeFutures_EOD",contract="LH0N",
from="2019-08-26",iuser = username, ipassword = password)
getPrice(feed="CME_CmeFutures_EOD_continuous",contract="HE_006_Month",
from="2019-08-26",iuser = username, ipassword = password)
getPrice(feed="ICE_EuroFutures",contract="BRN0Z",
from="2019-08-26",iuser = username, ipassword = password)
getPrice(feed="ICE_EuroFutures_continuous",contract="BRN_001_Month",
from="2019-08-26",iuser = username, ipassword = password)
getPrice(feed="ICE_NybotCoffeeSugarCocoaFutures",contract="SB0H",
from="2019-08-26",iuser = username, ipassword = password)
getPrice(feed="ICE_NybotCoffeeSugarCocoaFutures_continuous",contract="SF_001_Month",
from="2019-08-26",iuser = username, ipassword = password)

## End(Not run)
```

getPrices

Description

Multiple Morningstar API calls using getPrice functions. Refer to `getPrices()` for list of currently supported data feeds.

Usage

```r
getPrices(
  feed = "CME_NymexFutures_EOD",
  contracts = c("CL9Z", "CL0F", "CL0M"),
  from = "2019-01-01",
  iuser = "x@xyz.com",
  ipassword = "pass"
)
```
Arguments

- **feed**: Morningstar Feed Table
- **contracts**: Symbols vector
- **from**: From date as character string
- **iuser**: Morningstar user name as character - sourced locally in examples.
- **ipassword**: Morningstar user password as character - sourced locally in examples.

Value

- Wide data frame

Author(s)

Philippe Cote

Examples

```r
## Not run:
getPrices(feed="CME_NymexFutures_EOD",contracts=c("CL9Z","CL0F","CL0M"),
from="2019-08-26",iuser = username, ipassword = password)
## End(Not run)
```

Description

THoliday calendars for NYMEX and ICE Brent

Usage

`holidaysOil`

Format

- Data frame
Description

Extracts US Treasury Zero Rates

Usage

ir_df_us(quandlkey = quandlkey, ir.sens = 0.01)

Arguments

quandlkey Your Quandl key "quandlkey"
ir.sens Creates plus and minus IR sensitivity scenarios with specified shock value.

Value

Data frame of zero rates

Author(s)

Philippe Cote

Examples

## Not run:
us.df <- ir_df_us(quandlkey = quandlkey, ir.sens=0.01)

## End(Not run)

Description

EIA NG Storage Data

Usage

ng_storage

Format

data frame
Description
Compute NPV

Usage

npv(
  init.cost = -375,
  C = 50,
  cf.freq = 0.25,
  TV = 250,
  T2M = 2,
  disc.factors = us.df,
  BreakEven = FALSE,
  BE.yield = 0.01
)

Arguments
init.cost Initial investment cost
C Periodic cash flow
cf.freq Cash flow frequency in year fraction e.g. quarterly = 0.25
TV Terminal Value
T2M Time to Maturity in years
disc.factors Data frame of discount factors using ir.df.us() function.
BreakEven TRUE when using a flat discount rate assumption.
BE.yield Set the flat IR rate when BeakEven = TRUE.

Value
List of NPV and NPV Data frame

Author(s)
Philippe Cote

Examples
## Not run:
us.df <- ir_df_us(quandlkey = quandlkey,ir.sens=0.01)
npv(init.cost=-375,C=50,cf.freq=.5,TV=250,T2M=2,
disc.factors=us.df,BreakEven=TRUE,BE.yield=.0399)$npv
npv(init.cost=-375,C=50,cf.freq=.5,TV=250,T2M=2,
Description
Returns betas of multiple xts prices (by using relative returns).

Usage
promptBeta(x = x, period = "all", betatype = "all", output = "chart")

Arguments
x
Wide dataframe with date column and multiple series columns (multivariate).
period
"all" or numeric period of time in last n periods.
betatype
"all" "bull" "bear".
output
"betas", "chart", "stats"

Value
ggplot chart, df of betas or stats

Author(s)
Philippe Cote

Examples
## Not run:
x <- dflong %>% dplyr::filter(grepl("CL",series))
x <- x %>% dplyr::mutate(series=readr::parse_number(series)) %>% dplyr::group_by(series)
x <- returns(df=x,retType="abs",period.return=1,spread=TRUE)
x <- rolladjust(x=x,commodityname=c("cmewti"),rolltype=c("Last.Trade"))
promptBeta(x=x,period="all",betatype="all",output="chart")
promptBeta(x=x,period="all",betatype="all",output="betas")
promptBeta(x=x,period="all",betatype="all",output="stats")

## End(Not run)
**returns**

**Description**
Computes periodic returns from a dataframe ordered by date

**Usage**
`returns(df = dflong, retType = "abs", period.return = 1, spread = FALSE)`

**Arguments**
- **df**: Long dataframe with colnames = c("date","value","series")
- **retType**: "abs" for absolute, "rel" for relative, or "log" for log returns.
- **period.return**: Number of rows over which to compute returns.
- **spread**: TRUE if you want to spread into a long dataframe.

**Value**
A dataframe object of returns.

**Author(s)**
Philippe Cote

**Examples**
`returns(df=dflong,retType="rel",period.return=1,spread=TRUE)`
`returns(df=dflong,retType="rel",period.return=1,spread=FALSE)`

---

**rolladjust**

**Description**
Returns a xts price or return object adjusted for contract roll. The methodology used to adjust returns is to remove the daily returns on the day after expiry and for prices to adjust historical rolling front month contracts by the size of the roll at each expiry. This is conducive to quantitative trading strategies as it reflects the PL of a financial trader.

**Usage**
`rolladjust(x, commodityname = c("cmewti"), rolltype = c("Last.Trade"), ...)"
Arguments

x An xts object of prices or returns.
commodityname Name of commodity in expiry_table. See example below for values.
rolltype Type of contract roll: "Last.Trade" or "First.Notice".
... Other parms

Value

Roll-adjusted xts object of returns

Author(s)

Philippe Cote

Examples

unique(expiry_table$cmdty) # for list of commodity names
ret <- returns(df=dflong,retType="abs",period.return=1,spread=TRUE)[,1:2]
rolladjust(x=ret,commodityname=c("cmewti"),rolltype=c("Last.Trade"))

Description

Simulates a Geometric Brownian Motion process

Usage

simGBM(S0 = 10, drift = 0, sigma = 0.2, T2M = 1, dt = 1/12)

Arguments

S0 Spot price at t=0
drift Drift term in percentage
sigma Standard deviation
T2M Maturity in years
dt Time step in period e.g. 1/250 = 1 business day.

Value

A numeric vector of simulated values

Author(s)

Philippe Cote
simOU

Examples

simOU(S0=10,drift=0,sigma=0.2,T2M=1,dt=1/12)

Description

Simulates a Ornstein–Uhlenbeck process

Usage

simOU(S0 = 5, mu = 5, theta = 0.5, sigma = 0.2, T2M = 1, dt = 1/12)

Arguments

S0
S0 at t=0

mu
Mean reversion level

theta
Mean reversion speed

sigma
Standard deviation

T2M
Maturity in years

dt
Time step size e.g. 1/250 = 1 business day.

Value

A numeric vector of simulated values

Author(s)

Philippe Cote

Examples

simOU(S0=5,mu=5,theta=.5,sigma=0.2,T2M=1,dt=1/12)
Description

Simulates a Ornstein–Uhlenbeck process with Jumps

Usage

```r
simOUJ(
  S0 = 5,
  mu = 5,
  theta = 10,
  sigma = 0.2,
  jump_prob = 0.05,
  jump_avesize = 2,
  jump_stdv = 0.05,
  T2M = 1,
  dt = 1/250
)
```

Arguments

- **S0**: S at t=0
- **mu**: Mean reversion level
- **theta**: Mean reversion speed
- **sigma**: Standard deviation
- **jump_prob**: Probability of jumps
- **jump_avesize**: Average size of jumps
- **jump_stdv**: Standard deviation of jump average size
- **T2M**: Maturity in years
- **dt**: Time step size e.g. 1/250 = 1 business day.

Value

A numeric vector of simulated values

Author(s)

Philippe Cote

Examples

```r
simOUJ(S0=5,mu=5,theta=.5,sigma=0.2,jump_prob=0.05,jump_avesize = 3,jump_stdv = 0.05,T2M=1,dt=1/12)
```
描述

提供了一个返回分布的摘要。

使用

stl_decomp(x = x, output = "chart", s.window = 13, s.degree = 1, ...)  

参数

- **x**: 宽数据框，带有日期列和单个系列（一元）。  
- **output**: "chart" 用作图表。"data" 用于结果列表。  
- **s.window**: 季节性抽样的字符字符串"periodic"或loess 窗口的跨度（在滞后期），用于季节性提取，必须为奇数。没有默认值。  
- **s.degree**: 季节性抽样的局部拟合多项式。应该为零或一。  
- **...**: 其他参数

值

一个包含结果的图表或列表对象。

作者

Philippe Cote

示例

```r  
x <- dflong %>% dplyr::filter(series=="CL01")  
stl_decomp(x, output="chart", s.window=13, s.degree=1)  
stl_decomp(x, output="data", s.window=13, s.degree=1)  ```
swapCOM

Description

Commodity swap pricing from exchange settlement

Usage

\[
\text{swapCOM}(\text{futures} = \text{futs}, \\
\text{futuresNames} = c(\text{"CL0M"}, \text{"CL0N"}), \\
\text{pricingDates} = c(\text{"2020-05-01"}, \text{"2020-05-30"}), \\
\text{contract} = \text{"cmewti"}, \\
\text{exchange} = \text{"nymex"})
\]

Arguments

futures Wide data frame of futures prices for the given swap pricing dates
futuresNames Tickers of relevant futures contracts
pricingDates Vector of start and end pricing dates as character. See example.
contract Contract code in data(expiry_table). sort(unique(expiry_table$cmdty)) for options.
exchange Exchange code in data(holidaysOil). Currently only "nymex" and "ice" supported.

Value

Data frame of historic swap prices.

Author(s)

Philippe Cote

Examples

## Not run:
c <- paste0("CL0",c("M","N","Q"))
futs <- getPrices(feed="CME_NymexFutures_EOD",contracts = c,from="2019-08-26", 
iuser = username, ipassword = password)
swapCOM(futures = futs, futuresNames=c("CL0M","CL0N"), 
pricingDates = c("2020-05-01","2020-05-30"), contract = "cmewti", exchange = "nymex")

## End(Not run)
Description

Commodity swap pricing from exchange settlement

Usage

```r
swapIRS(
  trade.date = lubridate::today(),
  eff.date = lubridate::today() + 2,
  mat.date = lubridate::today() + 2 + lubridate::years(2),
  notional = 1e+06,
  PayRec = "Rec",
  fixed.rate = 0.05,
  float.curve = usSwapCurves,
  reset.freq = 3,
  disc.curve = usSwapCurves,
  convention = c("act", 360),
  bus.calendar = "NY",
  output = "price"
)
```

Arguments

- `trade.date`: Date object. Defaults to today().
- `eff.date`: Date object. Defaults to today() + 2 days.
- `mat.date`: Date object. Defaults to today() + 2 years.
- `notional`: Numeric value of notional. Defaults to 1,000,000.
- `PayRec`: "Pay" or "Rec" fixed.
- `fixed.rate`: Numeric fixed interest rate. Defaults to 0.05.
- `float.curve`: List of interest rate curves. Defaults to data("usSwapCurves").
- `reset.freq`: Numeric where 1 = "monthly", 3 = quarterly, 6 = Semi annual 12 = yearly.
- `disc.curve`: List of interest rate curves. Defaults to data("usSwapCurves").
- `convention`: Vector of convention e.g. c("act",360) c(30,360),...
- `bus.calendar`: Banking day calendar. Not implemented.
- `output`: "price" for swap price or "all" for price, cash flow data frame, duration.

Value

List of swap price, cash flow data frame, duration.
Author(s)

Philippe Cote

Examples

data("usSwapCurves")
swapIRS(trade.date = as.Date("2020-01-04"), eff.date = as.Date("2020-01-06"),
mat.date = as.Date("2022-01-06"), notional = 1000000,
PayRec = "Rec", fixed.rate=0.05, float.curve = usSwapCurves, reset.freq=3,
disc.curve = usSwapCurves, convention = c("act",360),
bus.calendar = "NY", output = "all")

tickers_eia
tickers_eia

data frame

Source

https://www.eia.gov/

tradeCycle

tradeCycle

data frame

Description

Supports automated upload of EIA data through its API by categories. Data frame organized by Supply Demand categories and products.

Usage

tickers_eia

Format

data frame

Source

https://www.eia.gov/

Description

Crude Trading Trade Cycles

Usage

tradeCycle

Format

data frame
tradeStats

Description
Compute list of risk reward metrics

Usage
tradeStats(x, Rf = 0)

Arguments
x xts object of returns
Rf Risk-free rate

Value
List of risk/reward metrics.

Author(s)
Philippe Cote

Examples
library(quantmod)
getSymbols("SPY", return.class = "zoo")
SPY$retClCl <- na.omit(quantmod::Delt(Cl(SPY),k=1,type='arithmetic'))
tradeStats(x=SPY$retClCl,Rf=0)

twoott
twoott

Description
NLP toy data set of Trump tweet.

Usage
twoott

Format
data frame
### Description
NLP toy data set of OOTT tweet.

### Usage
twtrump

### Format
data frame

---

### Description
USD IR Discount, Forward and Zero curves from RQuantlib::DiscountCurve

### Usage
usSwapCurves

### Format
List #’ @source Morningstar and FRED

---

### Description
USD IR Discount, Forward and Zero curves from RQuantlib::DiscountCurve - Parallel toy data set

### Usage
usSwapCurvesPar

### Format
data frame
**Description**

USD Interest Rate Swap Curve for RQuantlib bootstrapping. See `usSwapIRdef` for sources and tickers.

**Usage**

`usSwapIR`

**Format**

Data frame *source* Morningstar and FRED

**Description**

USD Interest Rate Swap Curve definitions with sources and tickers

**Usage**

`usSwapIRdef`

**Format**

Data frame *source* Morningstar and FRED
Index

*Topic datasets
  cancruceassays, 3
cancrudeprices, 4
df_fut, 8
dflong, 8
dfwide, 8
expiry_table, 10
holidaysOil, 15
ng_storage, 16
tickers_eia, 26
tradeCycle, 26
twoott, 27
twtrump, 28
usSwapCurves, 28
usSwapCurvesPar, 28
usSwapIR, 29
usSwapIRdef, 29

bond, 3
cancrudeassays, 3
cancrudeprices, 4
chart_fwd_curves, 4
chart_PerfSummary, 5
chart_zscore, 6
CRReuro, 7
df_fut, 8
dflong, 8
dfwide, 8
distdescplot, 9
eia2tidy, 9
expiry_table, 10

fitOU, 10

garch, 11
getIRswapCurve, 12
getPrice, 13
getPrices, 14

holidaysOil, 15
ir_df_us, 16
ng_storage, 16
npv, 17
promptBeta, 18
returns, 19
rolladjust, 19

simGBM, 20
simOU, 21
simOUJ, 22
stl_decomp, 23
swapCOM, 24
swapIRS, 25

tickers_eia, 26
tradeCycle, 26
tradeStats, 27
twoott, 27
twtrump, 28

usSwapCurves, 28
usSwapCurvesPar, 28
usSwapIR, 29
usSwapIRdef, 29