Package ‘Trading’

April 19, 2020

Type Package
Title CCR, Entropy-Based Correlation Estimates & Dynamic Beta
Version 2.0
Date 2020-04-19
Author Tasos Grivas
Maintainer Tasos Grivas <info@openriskcalculator.com>
Description Contains performance analysis metrics of track records including entropy-based correlation and dynamic beta based on the Kalman filter. The normalized sample entropy method has been implemented which produces accurate entropy estimation even on smaller datasets while for the dynamic beta calculation the Kalman filter methodology has been utilized. On a separate stream, trades from the five major asset classes and also functionality to use pricing curves, rating tables, CSAs and add-on tables. The implementation follows an object oriented logic whereby each trade inherits from more abstract classes while also the curves/tables are objects. There is a lot of functionality focusing on the counterparty credit risk calculations however the package can be used for trading applications in general.
Imports methods, reticulate, PerformanceAnalytics
URL www.openriskcalculator.com
License GPL-3
LazyData TRUE
Collate 'AngularDistance.R' 'Future.R' 'Swap.R' 'Vol.R' 'Option.R'
  'Trade.R' 'IRD.R' 'Bond.R' 'CSA.R' 'Chebyshev_distance.R'
  'Collateral.R' 'Commodity.R' 'Credit.R' 'CrossSampleEntropy.R'
  'Curve.R' 'DynamicBeta.R' 'Equity.R' 'FX.R' 'GetTradeDetails.R'
  'HashTable.R' 'InformationAdjustedBeta.R'
  'InformationAdjustedCorr.R' 'NormXASampEn.R' 'ParseTrades.R'
  'SampleEntropy.R' 'VariationOfInformation.R'
RoxygenNote 7.0.2
NeedsCompilation no
Repository CRAN
Repository/R-Forge/Project ccr
R topics documented:

AngularDistance ....................................................... 3
Bond-class ................................................................. 4
BondFuture-class ......................................................... 5
CDOTranche-class ......................................................... 6
Chebyshev_distance ...................................................... 6
Collateral-class ........................................................... 7
Commodity-class .......................................................... 8
CommodityForward-class ................................................ 9
CommSwap-class .......................................................... 10
CreditIndex-class ......................................................... 10
CreditSingle-class ......................................................... 11
CrossSampleEntropy ..................................................... 12
CSA-class ................................................................. 13
Curve-class ............................................................... 14
DynamicBeta .............................................................. 15
Equity-class .............................................................. 16
EquityIndexFuture-class .................................................. 16
EquityOption-class ....................................................... 17
FXSwap-class .............................................................. 18
GetTradeDetails .......................................................... 19
HashTable-class ........................................................... 19
InformationAdjustedBeta ............................................... 20
InformationAdjustedCorr ............................................... 21
IRDFuture-class ......................................................... 22
IRDSwap-class ........................................................... 22
IRDSwaption-class ....................................................... 23
IRDSwapVol-class ........................................................ 24
NormXASampEn .......................................................... 24
ParseTrades ............................................................... 25
SampleEntropy ........................................................... 26
VariationOfInformation .................................................. 27

Index 28
AngularDistance

AngularDistance

Angular distance metrics

Description

Calculates the angular distance between a matrix of the track records of various assets/strategies. The sign of the correlation can be ignored for long/short portfolios.

Usage

AngularDistance(returns_matrix, long_short = FALSE)

Arguments

returns_matrix  a matrix containing the track records of the underlying assets/strategies.
long_short       a boolean value which results in the sign of the correlation being ignored, default value is FALSE

Value

A matrix containing the angular distance values.

Author(s)

Tasos Grivas <tasos@openriskcalculator.com>

References


Examples

```r
## calling AngularDistance() without an argument loads the historical edhec data
## for the "Short Selling" and "Convertible Arbitrage" strategies
returns_matrix = PerformanceAnalytics::edhec[,c("Short Selling","Convertible Arbitrage")]
angular_distance = AngularDistance(returns_matrix, long_short=FALSE)
```
Bond-class

**Description**

Creates a Bond object with the relevant info needed to calculate the Exposure-at-Default (EAD)

**Arguments**

- **Notional**: The notional amount of the trade
- **MTM**: The mark-to-market valuation of the trade
- **Currency**: The currency set that the trade belongs to
- **Si**: The number of years that the trade will take to start (zero if already started)
- **BuySell**: Takes the values of either ‘Buy’ or ‘Sell’
- **yield**: The yield of the Bond
- **ISIN**: The ISIN of the Bond
- **payment_frequency**: The frequency that the bond pays coupon (Quarter, SA etc)
- **maturity_date**: The maturity date of the bond
- **coupon_type**: The coupon type of the bond (fixed, floating, flipper etc)
- **credit_risk_weight**: The percentage weight of the exposure of the bond that should be attributed to the 'Credit' asset class
- **Issuer**: The issuer of the bond

**Value**

An object of type Bond

**Author(s)**

Tasos Grivas <tasos@openriskcalculator.com>

**Examples**

```
tr1 = Bond(Notional=10000,MTM=30,Currency="EUR",Si=0,maturity_date="2026-04-04",BuySell='Buy',payment_Frequency="SA",credit_risk_weight=0.2,coupon_type="Fixed",Issuer="FirmA",ISIN = "XS0943423")
```
**BondFuture-class**

**Bond Future Class**

---

**Description**

Creates a Bond Future object with the relevant info needed to calculate the Exposure-at-Default (EAD)

**Arguments**

- **Notional**: The notional amount of the trade
- **MTM**: The mark-to-market valuation of the trade
- **Currency**: The currency set that the trade belongs to
- **Si**: The number of years that the trade will take to start (zero if already started)
- **Ei**: The number of years that the trade will expire
- **BuySell**: Takes the values of either 'Buy' or 'Sell'
- **yield**: The yield of the Underlying Bond
- **isin**: The ISIN of the Underlying Bond
- **payment_frequency**: The frequency that the bond pays coupon (Quarter, SA etc)
- **maturity_date**: The maturity date of the bond
- **coupon_type**: The coupon type of the bond (fixed, floating, flipper etc)
- **Issuer**: The issuer of the bond

**Value**

An object of type Bond

**Author(s)**

Tasos Grivas <tasos@openriskcalculator.com>

**Examples**

```r
example_trades = ParseTrades()
bondFuture_trade = example_trades[[17]]
tr1 = BondFuture(Notional=10000, MtM=30, Currency="EUR", Si=0, Ei=10, BuySell="Buy", payment_frequency="SA", coupon_type="Fixed", Issuer="CountryA", ISIN = "XS0943423")
```
**CDOTranche-class**

**CDO tranche Class**

**Description**

Creates a CDO tranche Object with the relevant info needed to calculate the Exposure-at-Default (EAD)

**Arguments**

- **Notional**: The notional amount of the trade
- **MTM**: The mark-to-market valuation of the trade
- **Currency**: The currency set that the belongs
- **Si**: The number of years after which the trade will start (zero if already started)
- **Ei**: The number of years that the trade will expire
- **BuySell**: Takes the values of either 'Buy' or 'Sell'
- **attach_point**: The attachment point of the tranche
- **detach_point**: The detachment point of the tranche

**Value**

An object of type CDOtrance

**Examples**

```r
## a CDO trance object
tr3 = CDOTranche(Notional=10000,MtM=0,Currency="USD",Si=0,Ei=5,
                  BuySell='Buy',SubClass='IG',RefEntity='CDX.IG',attach_point=0.3,detach_point=0.5)
```

---

**Chebyshev_distance**

**Chebyshev distance**

**Description**

Calculates the Chebyshev distance

**Usage**

```r
Chebyshev_distance(x, y)
```

**Arguments**

- **x**: a vector containing the track record of the underlying asset/strategy
- **y**: a vector containing the track record of the underlying asset/strategy
**Value**

The Chebyshev distance of the two vectors

**Author(s)**

Tasos Grivas <tasos@openriskcalculator.com>

**References**


**Examples**

```r
x = rnorm(1000)
y = rnorm(1000)

chebyshev_dist = Chebyshev_distance(x, y)
```

---

**Collateral-class**  
**Collateral Class**

**Description**

Creates a Collateral amount object which needs to be linked with a CSA ID

**Arguments**

<table>
<thead>
<tr>
<th>ID</th>
<th>The ID of each object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amount</td>
<td>The collateral amount</td>
</tr>
<tr>
<td>csa_id</td>
<td>The csa_id that this object is linked with</td>
</tr>
<tr>
<td>type</td>
<td>Describes the type of the collateral: can be &quot;ICA&quot;, &quot;VariationMargin&quot; etc</td>
</tr>
</tbody>
</table>

**Value**

An object of type Collateral

**Author(s)**

Tasos Grivas <tasos@openriskcalculator.com>

**References**

Basel Committee: The standardised approach for measuring counterparty credit risk exposures  
http://www.bis.org/publ/bcbs279.htm
Examples

colls = list()
coll_raw = read.csv(system.file("extdata", "coll.csv", package = "Trading"), header=TRUE, stringsAsFactors = FALSE)

for(i in 1:nrow(coll_raw))
{
  colls[[i]] = Collateral()
  colls[[i]]$PopulateViaCSV(coll_raw[i,])
}

Commodity-class  Commodity Class

Description

Creates a Commodity Object with the relevant info needed to calculate the Exposure-at-Default (EAD)

Arguments

Notional  The notional amount of the trade
MTM      The mark-to-market valuation of the trade
Currency The currency set that the trade belongs to
Si       The number of years that the trade will take to start (zero if already started)
BuySell  Takes the values of either 'Buy' or 'Sell'
commodity_type Takes the values of 'Oil/Gas','Silver','Electricity' etc.

Value

An object of type Commodity

Author(s)

Tasos Grivas <tasos@openriskcalculator.com>

References

Basel Committee: The standardised approach for measuring counterparty credit risk exposures http://www.bis.org/publ/bcbs279.htm

Examples

tr1 = Commodity(Notional=10000, MtM=-50, Si=0, BuySell='Buy', SubClass='Energy', commodity_type='Oil/Gas')
CommodityForward-class

Description

Creates a Commodity Forward Object with the relevant info needed to calculate the Exposure-at-Default (EAD)

Arguments

- **Notional**: The notional amount of the trade
- **MTM**: The mark-to-market valuation of the trade
- **Currency**: The currency set that the trade belongs to
- **Si**: The number of years that the trade will take to start (zero if already started)
- **Ei**: The number of years that the trade will expire
- **BuySell**: Takes the values of either 'Buy' or 'Sell'
- **commodity_type**: Takes the values of 'Oil/Gas','Silver','Electricity' etc.

Value

An object of type Commodity Forward

Author(s)

Tasos Grivas <tasos@openriskcalculator.com>

References

Basel Committee: The standardised approach for measuring counterparty credit risk exposures
http://www.bis.org/publ/bcbs279.htm

Examples

```r
## the Commodity Forward trade given in the Basel regulation Commodity example
tr1 = CommodityForward(Notional=10000,MTM=-50,si=0,Ei=0.75,
BuySell='Buy',SubClass='Energy',commodity_type='Oil/Gas')
```
**CreditIndex-class**

**Description**

Creates a Credit Index Object with the relevant info needed to calculate the Exposure-at-Default (EAD)

**Value**

An object of type CreditIndex

**Author(s)**

Tasos Grivas <tasos@openriskcalculator.com>

**References**

Basel Committee: The standardised approach for measuring counterparty credit risk exposures
http://www.bis.org/publ/bcbs279.htm

---

**CommSwap-class**

**Commodity Swap Class**

**Description**

Creates a Commodity Swap Object with the relevant info needed to calculate the Exposure-at-Default (EAD)

**Value**

An object of type CommSwap

**Author(s)**

Tasos Grivas <tasos@openriskcalculator.com>

**References**

Basel Committee: The standardised approach for measuring counterparty credit risk exposures
http://www.bis.org/publ/bcbs279.htm

---

**Arguments**

- **Notional**: The notional amount of the trade
- **MTM**: The mark-to-market valuation of the trade
- **Currency**: The currency set that the belongs
- **Si**: The number of years after which the trade will start (zero if already started)
- **Ei**: The number of years that the trade will expire
- **BuySell**: Takes the values of either 'Buy' or 'Sell'

**Value**

An object of type CreditIndex
Examples

```r
## the CreditIndex trade given in the Basel regulation Credit example
tr3 = CreditIndex(Notional=10000, MtM=0, Currency="USD", Si=0, Ei=5,
    BuySell='Buy', SubClass='IG', RefEntity='CDX.IG')
```

---

CreditSingle-class  
Credit Single Class

Description

Creates a Credit Single Object with the relevant info needed to calculate the Exposure-at-Default (EAD)

Arguments

- **Notional**: The notional amount of the trade
- **MtM**: The mark-to-market valuation of the trade
- **Currency**: The currency set that the trade belongs to
- **Si**: The number of years that the trade will take to start (zero if already started)
- **Ei**: The number of years that the trade will expire
- **BuySell**: Takes the values of either 'Buy' or 'Sell'

Value

An object of type CreditSingle

Author(s)

Tasos Grivas <tasos@openriskcalculator.com>

References

Basel Committee: The standardised approach for measuring counterparty credit risk exposures
http://www.bis.org/publ/bcbs279.htm

Examples

```r
## the CreditSingle trade given in the Basel regulation Credit example
tr1 = CreditSingle(Notional=10000, MtM=20, Currency="USD", Si=0, Ei=3,
    BuySell='Buy', SubClass='AA', RefEntity='FirmA')
```
CrossSampleEntropy

Angular distance metrics

Description

Calculates the cross sample entropy between two track records of various assets/strategies.

Usage

CrossSampleEntropy(returns_matrix, m = 2, r = 0.2)

Arguments

returns_matrix  a matrix containing the track records of the underlying assets/strategies. These will be normalized during the algorithm
m                  an integer value defining the embedding dimension, default value is 2
r                  a double value defining the tolerance, default value is 0.2

Value

The value of cross sample entropy

Author(s)

Tasos Grivas <tasos@openriskcalculator.com>

References


Examples

```r
## calling CrossSampleEntropy() without an argument loads the historical edhec data
## for the "Short Selling" and "Convertible Arbitrage" strategies
returns_matrix = PerformanceAnalytics::edhec[,c("Short Selling","Convertible Arbitrage")]
Cross_Sample_Entropy = CrossSampleEntropy(returns_matrix,m=2,r=0.2)
```
CSA-class

<table>
<thead>
<tr>
<th>CSA-class</th>
<th>CSA Class</th>
</tr>
</thead>
</table>

**Description**

Creates a collateral agreement Object containing all the relevant data and methods regarding the maturity factor and the calculation of the exposures after applying the relevant threshold.

**Arguments**

- ID: The ID of the CSA ID
- Counterparty: The counterparty the CSA is linked to
- Currency: The currency that the CSA applies to (can be a list of different currencies)
- TradeGroups: The trade groups that the CSA applies to
- Values_type: The type of the numerical values (can be "Actual" or "Perc" whereby the values are percentages of the MtM)
- thres_cpty: The maximum exposure that the counterparty can generate before collateral will need to be posted
- thres_PO: The maximum exposure that the processing organization can generate before collateral will need to be posted
- MTA_cpty: The minimum transfer amount for the counterparty
- MTA_PO: The minimum transfer amount for the processing organization
- IM_cpty: The initial margin that is posted by the counterparty
- IM_PO: The initial margin that is posted by the processing organization
- mpor_days: The margin period of risk in days
- remargin_freq: The frequency of re-margining the exposure in days
- rounding: The rounding amount of the transfers

**Value**

An object of type CSA

**Author(s)**

Tasos Grivas <tasos@openriskcalculator.com>

**References**

Basel Committee: The standardised approach for measuring counterparty credit risk exposures
http://www.bis.org/publ/bcbs279.htm
Examples

csa_raw = read.csv(system.file("extdata", "CSA.csv", package = "Trading"),
header=TRUE,stringsAsFactors = FALSE)

csas = list()
for(i in 1:nrow(csa_raw))
{
    csas[[i]] = CSA()
    csas[[i]]$PopulateViaCSV(csa_raw[i,])
}

Curve-class

Curve Class

Description

Creates a Curve Object containing pairs of Tenors with relevant rates and the interpolation function. Also, methods for populating the object via a .csv file and the generation of the interpolation function via cubic splines are included.

Arguments

Tenors The Tenors of the curve
Rates The rates on the corresponding tenors
interp_function (Optional) The interpolation function of the curve. Can be populated via the 'CalcInterpPoints' method

Value

An object of type Curve

Author(s)

Tasos Grivas <tasos@openriskcalculator.com>

Examples

```r
## generating a curve either directly or through a csv -
## the spot_rates.csv file can be found on the extdata folder in the installation library path
funding_curve = Curve(Tenors=c(1,2,3,4,5,6,10),Rates=c(4,17,43,47,76,90,110))
spot_rates = Curve()
spot_rates$PopulateViaCSV('spot_rates.csv')
time_points = seq(0,5,0.01)
spot_curve = spot_rates$CalcInterpPoints(time_points)
```
DynamicBeta  

Time Varying Beta via Kalman filter & smoother

Description

Calculates the beta of an investment strategy or stock by applying the Kalman filter & smoother. Apart from the beta timeseries, the state covariances are also returned so as to provide an estimate of the uncertainty of the results. The python package "Pykalman" is used for the calculations given its proven stability.

Usage

DynamicBeta(csvfilename, do_not_set_to_true=FALSE)

Arguments

csvfilename      the name of csv file containing the track record of the fund & the benchmark
do_not_set_to_true
                the name of csv file containing the track record of the fund & the benchmark

Value

A list of beta values based on Kalman Filter & smoother and the respective covariance matrices

Author(s)

Tasos Grivas <tasos@openriskcalculator.com>

Examples

## calling DynamicBeta() without an argument loads a test file containing
## a sample track record and a benchmark index
## ATTENTION!!: set do_not_set_to_true to FALSE when running the
## example -- this is only used to pass CRAN tests whereby pykalman was not installable!

dyn_beta_values = DynamicBeta(do_not_set_to_true = TRUE)
Equity-class  

Equity Class

Description

Creates an Equity object

Arguments

Notional  The notional amount of the trade
MTM  The mark-to-market valuation of the trade
Currency  The currency set that the trade belongs to
BuySell  Takes the values of either 'Buy' or 'Sell'
ISIN  the ISIN of the Equity
traded_price  the price that trade was done
Issuer  the issuer of the stock

Value

An object of type Equity

Author(s)

Tasos Grivas <tasos@openriskcalculator.com>

Examples

```r
tr1 = Equity(external_id="ext1",Notional=10000,MTM=30,Currency="EUR",BuySell='Buy',
traded_price = 10,ISIN = "XS04340432",Issuer='FirmA')
```

EquityIndexFuture-class  

Equity Index Future Class

Description

Creates an Equity Index Future object with the relevant info needed to calculate the Exposure-at-Default (EAD)
EquityOption-class

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notional</td>
<td>The notional amount of the trade</td>
</tr>
<tr>
<td>MTM</td>
<td>The mark-to-market valuation of the trade</td>
</tr>
<tr>
<td>Currency</td>
<td>The currency set that the trade belongs to</td>
</tr>
<tr>
<td>Si</td>
<td>The number of years that the trade will take to start</td>
</tr>
<tr>
<td>Ei</td>
<td>The number of years that the trade will expire</td>
</tr>
<tr>
<td>BuySell</td>
<td>Takes the values of either 'Buy' or 'Sell'</td>
</tr>
<tr>
<td>traded_price</td>
<td>the price that trade was done</td>
</tr>
</tbody>
</table>

Value

An object of type EquityIndexFuture

Author(s)

Tasos Grivas <tasos@openriskcalculator.com>

Examples

```r
equity_trades = ParseTrades()
equity_index_future_trade = equity_trades[[18]]
```

Declaration of an Equity Option class

**Description**

 Creates an Equity Option object with the relevant info needed to calculate the Exposure-at-Default (EAD)

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notional</td>
<td>The notional amount of the trade</td>
</tr>
<tr>
<td>MTM</td>
<td>The mark-to-market valuation of the trade</td>
</tr>
<tr>
<td>Currency</td>
<td>The currency set that the trade belongs to</td>
</tr>
<tr>
<td>Si</td>
<td>The number of years that the trade will take to start</td>
</tr>
<tr>
<td>Ei</td>
<td>The number of years that the trade will expire</td>
</tr>
<tr>
<td>BuySell</td>
<td>Takes the values of either 'Buy' or 'Sell'</td>
</tr>
<tr>
<td>traded_price</td>
<td>the price that trade was done</td>
</tr>
</tbody>
</table>

**Value**

An object of type EquityOption
FXSwap-class

Author(s)

Tasos Grivas <tasos@openriskcalculator.com>

---

**FX Swap Class**

**Description**

Creates an FX Swap object with the relevant info needed to calculate the Exposure-at-Default (EAD)

**Arguments**

- **Notional**: The notional amount of the trade
- **MTM**: The mark-to-market valuation of the trade
- **Currency**: The currency set that the trade belongs to
- **Si**: The number of years that the trade will take to start (zero if already started)
- **Ei**: The number of years that the trade will expire
- **BuySell**: Takes the values of either 'Buy' or 'Sell'
- **traded_price**: the price that trade was done

**Value**

An object of type FXSwap

Author(s)

Tasos Grivas <tasos@openriskcalculator.com>

**References**

Basel Committee: The standardised approach for measuring counterparty credit risk exposures
http://www.bis.org/publ/bcbs279.htm

**Examples**

```r
tr1 = FXSwap(Notional=10000,MTM=30,ccyPair="EUR/USD",Si=0,Ei=10,BuySell='Buy')
```
GetTradeDetails

Returns a list with the populated fields of a Trade Object

Description

Returns a list with the populated fields of a Trade Object

Usage

GetTradeDetails(trade)

Arguments

trade

A trade Object

Value

A list of fields

Author(s)

Tasos Grivas <tasos@openriskcalculator.com>

Examples

example_trades = ParseTrades()
Equity_Index_Future_trade = example_trades[[18]]
populated_fields = GetTradeDetails(Equity_Index_Future_trade)

HashTable-class

Hashtable Class

Description

Creates a hashtable-like object so as to represent data with a key structure (for example addon tables, rating-based factors etc). Also, it includes methods for populating the object via a .csv file and finding a value based on a specific key on an interval of keys For examples of the format of the CSVs files, please view RatingsMapping.csv orAddonTable.csv on the extdata folder in the installation folder of the library

Arguments

keys

A vector of keys

values

A vector of values mapping to the keys

keys_type

The type of the keys

values_type

The type of the values
Value
An object of type HashTable

Author(s)
Tasos Grivas <tasos@openriskcalculator.com>

Examples

```r
## loading a ratings' mapping matrix from the extdata folder
rating_table = HashTable('RatingsMapping.csv', 'character', "numeric")
reg_weight = rating_table$FindValue("AAA")
```

InformationAdjustedBeta

**Information Adjusted Beta**

Description
Calculates the Information-Adjusted Beta between the track records of two assets/strategies which covers for cases whereby the 'typical' linearity and Gaussian I.I.D assumptions do not hold. The normalized cross sample entropy has been utilized for the mutual information estimation.

Usage

```r
InformationAdjustedBeta(x, y, m = 2, r = 0.2)
```

Arguments

- `x`: a vector containing the track record of the underlying asset/strategy (can be a data.table, data.frame, vector etc)
- `y`: a vector containing the track record of the underlying asset/strategy (can be a data.table, data.frame, vector etc)
- `m`: an integer value defining the embedding dimension for the sample entropy calculation, default value is 2
- `r`: a double value defining the tolerance for the sample entropy calculation, default value is 0.2

Value
The information adjusted Beta

Author(s)
Tasos Grivas <tasos@openriskcalculator.com>
InformationAdjustedCorr

References

Examples

```r
x = PerformanceAnalytics::edhec[,c("Short Selling")]
y = PerformanceAnalytics::edhec[,c("Convertible Arbitrage")]
Information_Adjusted_Beta = InformationAdjustedBeta = function(x, y, m=2, r=0.2)
```

---

InformationAdjustedCorr

*Information Adjusted Correlation*

Description

Calculates the Information-Adjusted Correlation between the track records of various assets/strategies which covers for cases whereby the 'typical' Pearson’s correlation assumptions do not hold. The normalized cross sample entropy has been utilized for the mutual information estimation.

Usage

```r
InformationAdjustedCorr(x, y, m = 2, r = 0.2)
```

Arguments

- **x**: a vector containing the track record of the underlying asset/strategy (can be a data.table, data.frame, vector etc)
- **y**: a vector containing the track record of the underlying asset/strategy (can be a data.table, data.frame, vector etc)
- **m**: an integer value defining the embedding dimension for the sample entropy calculation, default value is 2
- **r**: a double value defining the tolerance for the sample entropy calculation, default value is 0.2

Value

The information adjusted correlation

Author(s)

Tasos Grivas <tasos@openriskcalculator.com>

References

Examples

```r
x = PerformanceAnalytics::edhec[,c("Short Selling")]
y = PerformanceAnalytics::edhec[,c("Convertible Arbitrage")]
Information_Adjusted_Corr = InformationAdjustedCorr(x, y, m=2, r=0.2)
```

IRDFuture-class

**IRD Future Class**

**Description**

Creates an IRD Future Object with the relevant info needed to calculate the Exposure-at-Default (EAD)

**Arguments**

- **Notional**
  - The notional amount of the trade
- **MTM**
  - The mark-to-market valuation of the trade
- **Currency**
  - The currency set that the trade belongs to
- **Si**
  - The number of years that the trade will take to start (zero if already started)
- **Ei**
  - The number of years that the trade will expire
- **BuySell**
  - Takes the values of either 'Buy' or 'Sell'

**Value**

An object of type IRDFuture

IRDSwap-class

**IRD Swap Class**

**Description**

Creates an IRD Swap Object with the relevant info needed to calculate the Exposure-at-Default (EAD)

**Arguments**

- **Notional**
  - The notional amount of the trade
- **MTM**
  - The mark-to-market valuation of the trade
- **Currency**
  - The currency set that the trade belongs to
- **Si**
  - The number of years that the trade will take to start (zero if already started)
- **Ei**
  - The number of years that the trade will expire
- **BuySell**
  - Takes the values of either 'Buy' or 'Sell'
**Value**

An object of type IRDSwap

**Examples**

```
# the IRD Swap trade given in the Basel regulation IRD example
tr1 = IRDSwap(Notional=10000, MtM=30, Currency="USD", Si=0, Ei=10, BuySell='Buy')
```

---

**Description**

Creates an IRD Swaption Object with the relevant info needed to calculate the Exposure-at-Default (EAD)

**Arguments**

- **Notional**: The notional amount of the trade
- **MTM**: The mark-to-market valuation of the trade
- **Currency**: The currency set that the trade belongs to
- **Si**: The number of years that the trade will take to start (zero if already started)
- **Ei**: The number of years that the trade will expire
- **BuySell**: Takes the values of either 'Buy' or 'Sell'
- **OptionType**: Takes the values of either 'Put' or 'Call'
- **UnderlyingPrice**: The current price of the underlying
- **StrikePrice**: The strike price of the option

**Value**

An object of type IRDSwaption

**Author(s)**

Tasos Grivas <tasos@openriskcalculator.com>

**References**

Basel Committee: The standardised approach for measuring counterparty credit risk exposures
http://www.bis.org/publ/bcbs279.htm
Examples

```r
# the Swaption trade given in the Basel regulation IRD example
tr3 = IRDSwaption(Notional=5000,MtM=50,Currency="EUR",Si=1,Ei=11,BuySell='Sell',
                   OptionType='Put',UnderlyingPrice=0.06,StrikePrice=0.05)
```

### IRDSwapVol-class
**IRD Swap Volatility Class**

**Description**
The IRDSwapVol-class is used to create an IRD Swap Volatility-based Object with the relevant info needed to calculate the Exposure-at-Default (EAD).

**Value**
A object of type IRDSwapVol

### NormXASampEn
**Normalized Cross Sample Entropy**

**Description**
Calculates the Normalized Cross Sample Entropy of the track records of two assets/strategies based on the sample entropy.

**Usage**
```
NormXASampEn(x, y, m = 2, r = 0.2)
```

**Arguments**
- `x`: a vector containing the track record of the underlying asset/strategy, this will be normalized during the algorithm
- `y`: a vector containing the track record of the underlying asset/strategy, this will be normalized during the algorithm
- `m`: an integer value defining the embedding dimension, default value is 2
- `r`: a double value defining the tolerance, default value is 0.2

**Value**
A double value containing the Normalized Cross Sample Entropy

**Author(s)**
Tasos Grivas <tasos@openriskcalculator.com>
ParseTrades

References

Examples

\begin{verbatim}
x = PerformanceAnalytics::edhec[,c("Short Selling")]
y = PerformanceAnalytics::edhec[,c("Convertible Arbitrage")]
Normalized_Cross_Sample_Entropy = NormXASampEn(x, y, m=2, r=0.2)
\end{verbatim}

---

 ParseTrades  Parse trades through a .csv file.

Description
Parse trades through a .csv file. In case no file name is given, an example file is automatically loaded containing trades corresponding to Basel’s SA-CCR regulation (the example trades file can be found on the extdata folder in the installation library path)

Usage

\begin{verbatim}
ParseTrades(csvfilename)
\end{verbatim}

Arguments

- `csvfilename`: the name of csv file containing the trades

Value

A list of trades

Author(s)

Tasos Grivas <tasos@openriskcalculator.com>

Examples

\begin{verbatim}
## calling ParseTrades() without an argument loads a test file containing all
## the different trade types supported
example_trades = ParseTrades()
\end{verbatim}
**SampleEntropy**

### Description
Calculates the sample entropy of a track record. Sample entropy is an improvement of the approximate entropy and should produce accurate results for timeseries of smaller length like historical returns of strategies.

### Usage
```
SampleEntropy(returns, m = 2, r = 0.2)
```

### Arguments
- **returns**: a vector containing the track record of the underlying asset/strategy, these will be normalized during the algorithm
- **m**: an integer value defining the embedding dimension, default value is 2
- **r**: a double value defining the tolerance, default value is 0.2

### Value
The sample Entropy of the input returns

### Author(s)
Tasos Grivas <tasos@openriskcalculator.com>

### References

### Examples
```
# calling SampleEntropy() without an argument loads the historical edhec
# data for the "Short Selling" strategy
returns = PerformanceAnalytics::edhec[,c("Short Selling")]
Sample_Entropy = SampleEntropy(returns,m=2,r=0.2)
```
VariationOfInformation

Variation of Information

Description
Calculates the variation of information of the track records of two assets/strategies based on the sample entropy.

Usage
VariationOfInformation(x, y, m = 2, r = 0.2, normalized = TRUE)

Arguments
- x: a vector containing the track record of the underlying asset/strategy, this will be normalized during the algorithm.
- y: a vector containing the track record of the underlying asset/strategy, this will be normalized during the algorithm.
- m: an integer value defining the embedding dimension, default value is 2.
- r: a double value defining the tolerance, default value is 0.2.
- normalized: a boolean value so as to bound the return value between 0 and 1, default value is TRUE.

Value
A double value containing the variation of information.

Author(s)
Tasos Grivas <tasos@openriskcalculator.com>

References

Examples
```r
x = PerformanceAnalytics::edhec[,c("Short Selling")]
y = PerformanceAnalytics::edhec[,c("Convertible Arbitrage")]
variation_of_information = VariationOfInformation(x, y, m=2, r=0.2, normalized = TRUE)
```
Index

AngularDistance, 3
Bond (Bond-class), 4
Bond-class, 4
BondFuture (BondFuture-class), 5
BondFuture-class, 5
CDOTranche (CDOTranche-class), 6
CDOTranche-class, 6
Chebyshev_distance, 6
Collateral (Collateral-class), 7
Collateral-class, 7
Commodity (Commodity-class), 8
Commodity-class, 8
CommodityForward
(CommodityForward-class), 9
CommodityForward-class, 9
CommSwap (CommSwap-class), 10
CommSwap-class, 10
CreditIndex (CreditIndex-class), 10
CreditIndex-class, 10
CreditSingle (CreditSingle-class), 11
CreditSingle-class, 11
CrossSampleEntropy, 12
CSA (CSA-class), 13
CSA-class, 13
Curve (Curve-class), 14
Curve-class, 14
DynamicBeta, 15
Equity (Equity-class), 16
Equity-class, 16
EquityIndexFuture
(EquityIndexFuture-class), 16
EquityIndexFuture-class, 16
EquityOption (EquityOption-class), 17
EquityOption-class, 17
FXSwap (FXSwap-class), 18
FXSwap-class, 18

GetTradeDetails, 19
HashTable (HashTable-class), 19
HashTable-class, 19
InformationAdjustedBeta, 20
InformationAdjustedCorr, 21
IRDFuture (IRDFuture-class), 22
IRDFuture-class, 22
IRDSwap (IRDSwap-class), 22
IRDSwap-class, 22
IRDSwaption (IRDSwaption-class), 23
IRDSwaption-class, 23
IRDSwapVol (IRDSwapVol-class), 24
IRDSwapVol-class, 24
NormXASampEn, 24
ParseTrades, 25
SampleEntropy, 26
VariationOfInformation, 27