Package ‘Trading’

January 12, 2019

Type Package

Title Trades, Curves, Rating Tables, Add-on Tables, CSAs

Version 1.2

Date 2019-01-12

Author Tasos Grivas

Maintainer Tasos Grivas <info@openriskcalculator.com>

Description Contains trades from the five major assets 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

URL www.openriskcalculator.com

License GPL-3

LazyData TRUE

Collate 'Swap.R' 'Future.R' 'Vol.R' 'Option.R' 'Trade.R' 'IRD.R' 'Bond.R' 'CSA.R' 'Collateral.R' 'Commodity.R' 'Credit.R' 'Curve.R' 'Equity.R' 'FX.R' 'GetTradeDetails.R' 'HashTable.R' 'ParseTrades.R'

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-01-12 19:50:02 UTC

Repository/R-Forge/Project ccr

Repository/R-Forge/Revision 17

Repository/R-Forge/DateTimeStamp 2019-01-12 13:35:47
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
BondFuture-class

### Description

Creates a Bond Future 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 (zero if already started)</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>yield</td>
<td>The yield of the Underlying Bond</td>
</tr>
<tr>
<td>isin</td>
<td>The ISIN of the Underlying Bond</td>
</tr>
<tr>
<td>payment_frequency</td>
<td>the frequency that the bond pays coupon (Quarter, SA etc)</td>
</tr>
<tr>
<td>maturity_date</td>
<td>the maturity date of the bond</td>
</tr>
<tr>
<td>coupon_type</td>
<td>The coupon type of the bond (fixed, floating, flipper etc)</td>
</tr>
<tr>
<td>issuer</td>
<td>The issuer of the bond</td>
</tr>
</tbody>
</table>

### Value

An object of type Bond

### Author(s)

Tasos Grivas <tasos@openriskcalculator.com>

### Examples

```r
ttr1 = 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")
```
Value

An object of type Bond

Author(s)

Tasos Grivas <tasos@openriskcalculator.com>

Examples

e.example_trades = ParseTrades()
  bondfuture_trades = 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 CDOtranche

Examples

```r
# a CDO tranche 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)
```
Collateral-class

Collateral Class

Description

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

Arguments

- **ID**
  - The ID of each object
- **Amount**
  - The collateral amount
- **csa_id**
  - The csa_id that this object is linked with
- **type**
  - Describes the type of the collateral: can be "ICA", "VariationMargin" etc

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**

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

**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**

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

*Commodity Forward 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=8,Ei=0.75,
  BuySell='Buy',SubClass='Energy',commodity_type='Oil/Gas')
```
**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

---

**CreditIndex-class**  
*Credit Index Class*

**Description**

Creates a Credit Index 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'

**Value**

An object of type CreditIndex
CreditSingle-class

Examples

```r
## the CreditIndex trade given in the Basel regulation Credit example
t3 = 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
t1 = CreditSingle(Notional=10000, MtM=20, Currency="USD", Si=0, Ei=3, BuySell='Buy',
    SubClass='AA', RefEntity='FirmA')
```
CSA-class  CSA Class

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_P0**: 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_P0**: The minimum transfer amount for the processing organization
- **IM_cpty**: The initial margin that is posted by the counterparty
- **IM_P0**: The initial margin that is posted by the processing organization
- **mport_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
Curve-class

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

## 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)
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 (zero if already started)</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
example_trades <- ParseTrades()
Equity_Index_Future_trade <- example_trades[[18]]
```

EquityOption-class

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 (zero if already started)</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>

---

**FXSwap-class**  
**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,cyPair="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

```r
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 or AddonTable.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
ing_table = HashTable('RatingsMapping.csv',"character","numeric")
reg_weight = rating_table$FindValue("AAA")
```

---

<table>
<thead>
<tr>
<th>IRDFuture-class</th>
<th>IRD Future Class</th>
</tr>
</thead>
</table>

Description

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

Arguments

<table>
<thead>
<tr>
<th>Notional</th>
<th>The notional amount of the trade</th>
</tr>
</thead>
<tbody>
<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 (zero if already started)</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>
</tbody>
</table>

Value

An object of type IRDFuture
**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**

```r
# 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')
```

---

**IRD Swaption Class**

**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'
Description

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

Value

An object of type IRDSwapVol
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

ParseTrades(csvfilename)

Arguments

csvfilename the name of csv file containing the trades

Value

A list of trades

Author(s)

Tasos Grivas <tasos@openriskcalculator.com>

Examples

## calling ParseTrades() without an argument loads a test file containing all
## the different trade types supported
example_trades = ParseTrades()
Index

Bond (Bond-class), 2
Bond-class, 2
BondFuture (BondFuture-class), 3
BondFuture-class, 3

CDOTranche (CDOTranche-class), 4
CDOTranche-class, 4
Collateral (Collateral-class), 5
Collateral-class, 5
Commodity (Commodity-class), 6
Commodity-class, 6
CommodityForward

[CommodityForward-class], 7
CommodityForward-class, 7
CommSwap (CommSwap-class), 8
CommSwap-class, 8
CreditIndex (CreditIndex-class), 8
CreditIndex-class, 8
CreditSingle (CreditSingle-class), 9
CreditSingle-class, 9
CSA (CSA-class), 10
CSA-class, 10
Curve (Curve-class), 11
Curve-class, 11

Equity (Equity-class), 12
Equity-class, 12
EquityIndexFuture

[EquityIndexFuture-class], 12
EquityIndexFuture-class, 12
EquityOption (EquityOption-class), 13
EquityOption-class, 13

FXSwap (FXSwap-class), 14
FXSwap-class, 14

GetTradeDetails, 15

HashTable (HashTable-class), 15
HashTable-class, 15

IRDFuture (IRDFuture-class), 16
IRDFuture-class, 16
IRDSwap (IRDSwap-class), 17
IRDSwap-class, 17
IRDSwaption (IRDSwaption-class), 17
IRDSwaption-class, 17
IRDSwapVol (IRDSwapVol-class), 18
IRDSwapVol-class, 18

ParseTrades, 19