

Package ‘fInstrument’

May 19, 2013

Type Package

Title Package for performing generic calculations on Rmetrics instruments

Version 0.1

Date 2010-09-22

Author P. Henaff

Maintainer P. Henaff <henaff.iae@univ-paris1.fr>

Description The fInstrument class provides an abstraction layer over the various types of financial claims available in Rmetrics. The class exposes methods for computing NPV and the “greeks”. With this class, you can perform calculations on a portfolio of financial instruments, without having to be concerned with the implementation details specific to each kind of instrument.

License GPL(>=2)

LazyLoad yes

Depends fOptions,fExoticOptions,fAsianOptions,timeSeries,empfin,methods

Collate 'StandardBarrier.r' 'Asian.r' 'Binary.r' 'Vanilla.r' 'DataProvider.r' 'fInstrument.r' 'fInstrument-package.r'

R topics documented:

fInstrument-package	2
Asian	2
Binary	3
DataProvider	4
fInstrument	4
fInstrumentFactory	5
getData	5
getValue	6
setData	6
show	7
StandardBarrier	7
Vanilla	8

Index**9**

fInstrument-package	<i>fInstrument Package</i>
---------------------	----------------------------

Description

The fInstrument package is useful for performing calculations on financial instruments defined in Rmetrics.

Author(s)

P. Henaff

Asian	<i>Asian European Option</i>
-------	------------------------------

Description

A helper function for creating a [fInstrument](#) of type arithmetic Asian European option. Calculations are performed by the function TurnbullWakemanAsianApproxOption from Rmetrics.

Usage

```
Asian(q, params)
```

Arguments

q	quantity >0 for long position, <0 for short
params	list of parameters that define a european arithmetic asian option: cp [string] c (call) or p (put) strike [numeric] strike avg [numeric] average so far dtExpiry [timeDate] expiry date dtEnd [timeDate] end of averaging dtStart [timeDate] start of averaging underlying [string] name of underlying asset discountRef [string] name of discount curve trace [boolean] print trace?

Value

an object of type [fInstrument](#)

Examples

```
v <- Asian(q=1, params=list(cp='c', strike=100, dtExpiry=as.timeDate('01-jan-2011'),
  dtEnd = as.timeDate('01-jan-2011'),
  dtStart = as.timeDate('01-jan-2010'),
  avg=0.0,
  underlying='IBM', discountRef='USD-LIBOR', trace=FALSE))
```

 Binary

Binary European Option

Description

A helper function for creating a [fInstrument](#) of type binary european option. Calculations are performed by the function `CashOrNothingOption` from `Rmetrics`.

Usage

```
Binary(q, params)
```

Arguments

<code>q</code>	quantity >0 for long position, <0 for short
<code>params</code>	list of parameters that define a vanilla option: <ul style="list-style-type: none"> <code>cp</code> [string] c (call) or p (put) <code>strike</code> [numeric] strike <code>dtExpiry</code> [timeDate] expiry date <code>underlying</code> [string] name of underlying asset <code>discountRef</code> [string] name of discount curve <code>trace</code> [boolean] print trace?

Value

an object of type [fInstrument](#)

Examples

```
v <- Binary(q=1, params=list(cp='c', strike=100, dtExpiry=as.timeDate('01-jan-2011'),
  underlying='IBM', discountRef='USD-LIBOR', trace=FALSE))
```

DataProvider	<i>A container class for storing market data</i>
--------------	--

Description

A DataProvider is a container of market data, to be used in conjunction with a `fInstrument` object. A data item is defined as the observation of a phenomenon on an instrument. Therefore, the data item is identified by the instrument being observed, the phenomenon and the observation date.

Constructor

Usage

```
DataProvider(parent = NULL)
```

Arguments

parent	a back-up DataProvider. If the data is not found in the current provider, the back-up provider is searched.
--------	---

Value

an object of type DataProvider

Author(s)

P. Henaff

fInstrument	<i>A class for representing financial instruments</i>
-------------	---

Description

fInstrument

Details

The fInstrument class provides an abstraction layer over the various types of financial claims available in Rmetrics. The class exposes methods for computing NPV and greeks. With this class, you can perform calculations on a portfolio of financial instruments, without having to be concerned with the implementation details specific to each kind of instrument.

Author(s)

P. Henaff

Examples

```
dtExpiry <- as.timeDate('01-jan-2011')
underlying <- 'IBM'
K<-100
a <- fInstrumentFactory("vanilla", quantity=1,
                        params=list(cp='c', strike=K,
                                    dtExpiry=dtExpiry,
                                    underlying=underlying,
                                    discountRef='USD.LIBOR', trace=FALSE))
show(a)
```

fInstrumentFactory	<i>fInstrument constructor</i>
--------------------	--------------------------------

Description

Factory method for constructing objects of type fInstrument

Usage

```
fInstrumentFactory(type, quantity, params)
```

Arguments

type	(char) the instrument type
quantity	(numeric) the position
params	(list) list of parameters specific to the instrument type

Examples

```
a <- fInstrumentFactory("vanilla", quantity=1,
                        params=list(cp='c', strike=100,
                                    dtExpiry=as.timeDate('01-jan-2011'),
                                    underlying='IBM',
                                    discountRef='USD.LIBOR', trace=FALSE))
```

getData	<i>getData method</i>
---------	-----------------------

Description

Extract data from a DataProvider object

Arguments

instrument	instrument name
phenomenon	phenomenon name
dtObs	observation date

Value

a vector of value(s)

getValue	<i>GetValue</i>
----------	-----------------

Description

Invoke a calculation on a financial instrument

Arguments

selection	The type of calculation ('Value', 'Delta', 'Gamma', 'Vega')
dtCalc	the calculation date
env	the data provider

Value

a time series of results

setData	<i>insert data into DataProvider objectsetData</i>
---------	--

Description

Insert data in a DataProvider object

Arguments

instrument	(char) instrument name
phenomenon	(char) phenomenon name
dtObs	(date) observation date
value	(numeric) value

show	<i>Display a description of the instrument</i>
------	--

Description

Show

Details

Provides a description of the instruments (i.e. the value of the main parameters found in the constructor

Value

A description of the instrument

StandardBarrier	<i>Standard Barrier European Option</i>
-----------------	---

Description

A helper function for creating a [fInstrument](#) of type standard barrier european option. Calculations are performed by the function StandardBarrierOption from Rmetrics.

Usage

StandardBarrier(q, params)

Arguments

- q
- quantity >0 for long position, <0 for short
- params
- list of parameters that define a european arithmetic asian option:
typeflag [string] 'c' (call) or 'p' (put) +
 di down-and-in
 ui up-and-in
 do down-and-out
 uo up-and-out
strike [numeric] strike
barrier [numeric] barrier
rebate [numeric] rebate paid out if the barrier has not been breached
dtExpiry [timeDate] expiry date
underlying [string] name of underlying asset
discountRef [string] name of discount curve
trace [boolean] print trace?

Value

an object of type `fInstrument`

Examples

```
v <- StandardBarrier(q=1, params=list(cp='cuo', strike=100, barrier=120,
  rebate=0, dtExpiry=as.timeDate('01-jan-2011'),
  underlying='IBM', discountRef='USD-LIBOR',
  trace=FALSE))
```

Vanilla

Vanilla European Option

Description

A helper function for creating a `fInstrument` of type vanilla european option. Calculations are performed by the function `GBSOption` and `GBSGreeks` from `Rmetrics`.

Usage

```
Vanilla(q, params)
```

Arguments

<code>q</code>	quantity >0 for long position, <0 for short
<code>params</code>	list of parameters that define a vanilla option: <ul style="list-style-type: none"> <code>cp</code> [<code>string</code>] c (call) or p (put) <code>strike</code> [<code>numeric</code>] strike <code>dtExpiry</code> [<code>timeDate</code>] expiry date <code>underlying</code> [<code>string</code>] name of underlying asset <code>discountRef</code> [<code>string</code>] name of discount curve <code>trace</code> [<code>boolean</code>] print trace?

Value

an object of type `fInstrument`

Examples

```
v <- Vanilla(q=1, params=list(cp='c', strike=100, dtExpiry=as.timeDate('01-jan-2011'),
  underlying='IBM', discountRef='USD-LIBOR', trace=FALSE))
```


Index

Asian, [2](#)

Binary, [3](#)

DataProvider, [4](#)

fInstrument, [2–4](#), [4](#), [7](#), [8](#)

fInstrument-package, [2](#)

fInstrumentFactory, [5](#)

getData, [5](#)

getValue, [6](#)

setData, [6](#)

show, [7](#)

StandardBarrier, [7](#)

Vanilla, [8](#)