Package ‘empfin’

May 19, 2013

Type Package

Title Utility function for Empirical Finance e-book

Version 1.0

Date 2010-09-22

Author P. Henaff

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

Description functions and data for the e-book ‘‘Topics in Empirical Finance with R and Rmetrics’’

License GPL(>=2)

LazyLoad yes

Depends fOptions,fExoticOptions,fAsianOptions,timeSeries,methods,timeDate,fImport,RCurl,Hmisc

Collate 'BondUtils.R''DateUtils.r''LatexUtils.r''OptionUtils.R'

R topics documented:

`addDays` *add days*

Description

add days to timeDate or Date objects

Usage

```
addDays(dt, days)
```

Arguments

`dt`	a Date or timeDate object
`days`	to be added

Value

a Date or timeDate object

Examples

```
d1 <- myDate('23-10-2010')
d2 <- addDays(d1, 1)
```

`Bond` *Bond cash flow generator*

Description

Bond Definition

Usage

```
Bond(id, dtIssue, dtMaturity, couponRate, nominal,
  frequency)
```

Arguments

`id`	(string) the identifier of the bond
`dtIssue`	(date) bond issue date
`dtMaturity`	(date) bond maturity date
`couponRate`	(real) coupon rate, in decimal form
`nominal`	(real) nominal amount
`frequency`	(string) 'A' for annual, 'S' for semi-annual.

Details

This function constructs a data structure that describes the cash flow schedule of a bond.

Value

a list with the following elements:

id the bond identifier

cf vector of cash flows

dt vector of payment dates for each cash flow

freq frequency ("a" or "s")

coupon coupon rate

nominal nominal amount

Examples

```
b374 <- Bond('374', myDate('01/01/2000'),
            myDate('04/01/2019'), .0375, 100, 'a')
```

`BondAC` *Bond accrued interest*

Description

This function computes the accrued interest on a bond by the formula

$$ac = cN\frac{m}{365}$$

Usage

```
BondAC(bond, dtSettlement)
```

Arguments

`bond`	data structure generated by `Bond`
`dtSettlement`	settlement date

Details

with

c coupon rate

N nominal amount

m actual number of days since last coupon payment

Value

the accrued interest

Examples

```
b374 <- Bond('374', myDate('01/01/2000'),
             myDate('04/01/2019'), .0375, 100, 'a')
a <- BondAC(b374, myDate('03/01/2012'))
```

BondCvx *Bond convexity*

Description

This function computes the convexity of a bond by the formula

$$Cx = \frac{\partial^2 P(y)}{\partial y^2}$$

given the bond price or yield.

Usage

```
BondCvx(bond, dtSettlement, yield, price = NULL)
```

Arguments

bond	data structure generated by Bond
dtSettlement	settlement date
yield	(optional) bond yield
price	(optional) bond price

Details

with

$P(y)$ bond price, function of yield

Value

Bond convexity

Examples

```
b374 <- Bond('374', myDate('01/01/2000'),
             myDate('04/01/2019'), .0375, 100, 'a')
Cvx <- BondCvx(b374, myDate('03/01/2012'), yield=.04)
```

BondPV01 *Bond PV01*

Description

This function computes the PV01 (present value of one basis point) of a bond by the formula

$$PV01 = \frac{1}{10000} \frac{\partial P(y)}{\partial y}$$

given the bond price or yield.

Usage

```
BondPV01(bond, dtSettlement, yield, price = NULL)
```

Arguments

bond	data structure generated by Bond
dtSettlement	settlement date
yield	(optional) bond yield
price	(optional) bond price

Details

with

$P(y)$ bond price, function of yield

Value

PV01

Examples

```
b374 <- Bond('374', myDate('01/01/2000'),
             myDate('04/01/2019'), .0375, 100, 'a')
pv01 <- BondPV01(b374, myDate('03/01/2012'), yield=.04)
```

`BondYield2Price` *Bond price from yield*

Description

This function computes the bond (dirty) price, given its yield, by the formula

$$P = \sum_{i=1}^{n} F_i (1+y)^{-\frac{d_i - d_0}{365}}$$

with:

F_i Cash flow paid at date d_i

d_i Cash flow date, measured in days

d_0 settlement date

Usage

```
BondYield2Price(bond, dtSettlement, yield)
```

Arguments

`bond`	data structure generated by `Bond`
`dtSettlement`	settlement date
`yield`	bond yield

Value

dirty price

Examples

```
b374 <- Bond('374', myDate('01/01/2000'),
             myDate('04/01/2019'), .0375, 100, 'a')
p <- BondYield2Price(b374, myDate('03/01/2012'), yield=.04)
```

CRRTrinomial — *Trinomial tree model*

Description

Trinomial model

Usage

```
CRRTrinomial(TypeFlag = c("ce", "pe", "ca", "pa"), S, X,
  Time, r, b, sigma, n)
```

Arguments

`TypeFlag`	the option type: `ce` European call `pe` European put `ca` American call `pa` American put
`S`	price of underlying asset
`X`	strike
`Time`	time to expiry
`r`	risk-free interest rate
`b`	cost of carry
`sigma`	annualized volatility
`n`	number of steps in tree

Details

Trinomial model for pricing European or American vanilla options.

Value

a data structure with the following elements:

`param`	list of input parameters
`price`	price
`delta`	delta
`gamma`	gamma

Examples

```
res <- CRRTrinomial(TypeFlag="ce", S=100, X=100, Time=1, r=.03,
   b=.03, sigma=.3, n=100)
```

CRRWithPayOff *Cox-Ross-Rubinstein binomial model with payoff function*

Description

Cox-Ross-Rubinstein binomial model with payoff function

Usage

```
CRRWithPayOff(TypeFlag = c("e", "a"), PayOff, S, Time, r,
  b, sigma, n)
```

Arguments

`TypeFlag`	e:European exercise, a: American
`PayOff`	PayOff function: f(underlying value)
`S`	Spot
`Time`	Time to maturity
`r`	interest rate
`b`	cost of carry
`sigma`	volatility
`n`	number of time steps

Value

PV of option

myDate *Date constructor*

Description

Date creation

Usage

```
myDate(dt)
```

Arguments

`dt`	(string) a date. Legal formats: `ddmmmYYYY` ex: 23jun2010 `dd-mm-YYYY` ex: 23-12-2010 `dd/mm/YYYY` ex: 23/12/2010 `YYYY/mm/dd` ex: 2010/12/23 `dd.mm.YYYY` ex: 23.12.2010

Details

A shortcut for date creation, locale independent validates day and month ranges

Value

a Date object

Examples

```
d1 <- myDate('30-10-2010')
d2 <- myDate('30/10/2010')
d3 <- myDate('30.10.2010')
d4 <- myDate('30oct2010')
(d1 == d2) & (d1 == d3) & (d1 == d4)
```

---

mytDate   *dateTime constructor*

Description

timeDate creation

Usage

```
mytDate(dt)
```

Arguments

dt	(string) a date. Legal formats: **ddmmmYYYY** ex: 23jun2010 **dd-mm-YYYY** ex: 23-12-2010 **dd/mm/YYYY** ex: 23/12/2010 **dd.mm.YYYY** ex: 23.12.2010

Details

A shortcut for date creation, locale independent validates day and month ranges

Value

a timeDate object

Examples

```
d1 <- mytDate('30-10-2010')
d2 <- mytDate('30/10/2010')
d3 <- mytDate('30.10.2010')
d4 <- mytDate('30oct2010')
(d1 == d2) & (d1 == d3) & (d1 == d4)
```

`Price2BondYield` *Bond yield from price*

Description

This function computes the bond yield, given its dirty price by solving for y the equation:

$$P = \sum_{i=1}^{n} F_i (1+y)^{-\frac{d_i - d_0}{365}}$$

with:

F_i Cash flow paid at date d_i

y yield

d_i Cash flow date, measured in days

d_0 settlement date

Usage

```
Price2BondYield(bond, dtSettlement, price)
```

Arguments

`bond`	data structure generated by Bond
`dtSettlement`	settlement date
`price`	bond dirty price

Value

bond yield

Examples

```
b374 <- Bond('374', myDate('01/01/2000'),
             myDate('04/01/2019'), .0375, 100, 'a')
y <- Price2BondYield(b374, myDate('03/01/2012'),
     price=101)
```

`print.xb` *Table pretty printer*

Description

Table pretty printer

Usage

```
print.xb(xtab, ...)
```

Arguments

`xtab`	a table

`SimpleBondPrice` *Bond price*

Description

Simplified bond price calculation

Usage

```
SimpleBondPrice(coupon, n, yield)
```

Arguments

`coupon`	(real) coupon rate (.05: 5%)
`n`	(integer) number of years to expiry
`yield`	(real) yield to maturity

Details

Price of a bond with annual coupon, computed on coupon payment date

$$P = \sum_{i=1}^{n} \frac{c}{(1+y)^i} + \frac{1}{(1+r)^n}$$

Value

price of $1 nominal

Examples

```
p <- SimpleBondPrice(.05, 10, .05)
```

SimpleDuration *Bond duration*

Description

Simplified bond duration calculation

Usage

```
SimpleDuration(coupon, n, yield)
```

Arguments

coupon	(real) coupon rate (.05: 5%)
n	(integer) number of years to expiry
yield	(real) yield to maturity

Details

Modified duration of a bond price. It is the weighted maturity, where the weights are the present value of the cash flow paid at each date. Let F_i be the cash flow paid at time i. Duration is defined as

$$d = \sum_{i=1}^{n} \frac{F_i(1+y)^{-i}}{P}$$

Value

derivative of price with respect to yield

Examples

```
d <- SimpleDuration(.05, 10, .05)
```

SimpleSensitivity *Bond sensitivity*

Description

Simplified bond sensitivity calculation

Usage

```
SimpleSensitivity(coupon, n, yield)
```

Arguments

`coupon`	(real) coupon rate (.05: 5%)
`n`	(integer) number of years to expiry
`yield`	(real) yield to maturity

Details

Bond sensitivity with respect to yield, ie.

$$s = -\frac{frac\partial P\partial y}{P}$$

Value

derivative of price with respect to yield

Examples

```
s <- SimpleSensitivity(.05, 10, .05)
```

`SimpleVariation` *Bond variation*

Description

Simplified bond variation calculation

Usage

```
SimpleVariation(coupon, n, yield)
```

Arguments

`coupon`	(real) coupon rate (.05: 5%)
`n`	(integer) number of years to expiry
`yield`	(real) yield to maturity

Details

Derivative of bond price with respect to yield

$$v = \frac{\partial P}{\partial y}$$

Value

derivative of price with respect to yield

Examples

```
p <- SimpleVariation(.05, 10, .05)
```

---

tDiff *time difference*

Description

time difference in fraction of years, given 2 timeDate or Date objects

Usage

```
tDiff(dtFirst, dtLast)
```

Arguments

dtFirst	a date or timeDate
dtLast	a date or timeDate

Value

the time difference in year fraction

TrinomialTreePlot *Trinomial tree plot*

Description

Trinomial tree plot

Usage

```
TrinomialTreePlot(TrinomialTreeValues, dx = -0.025,
  dy = 0.4, cex = 1, digits = 2, ...)
```

Arguments

TrinomialTreeValues	tree values grid
dx	step
digits	format

Index