SmithWilsonYieldCurve-package

Fit yield curves using the Smith-Wilson method

Description

A package to fit yield curves using the Smith-Wilson method

Details

The main function exposed in this package is fFitSmithWilsonYieldCurve, which takes market data in the form of a vector of cashflow times, a matrix of cashflows and a vector of market prices. It returns an object of class "SmithWilsonYieldCurve".

A convenience function fFitSmithWilsonYieldCurveToInstruments takes a dataframe containing market instrument data as type, tenor, frequency and rate. It extracts the required vectors and matrices and then calls fFitSmithWilsonYieldCurve.

Objects of class SmithWilsonYieldCurve are a list, the first element of which is a function P(t), which returns the zero coupon bond price of the fitted curve at time t.

Details including mathematics at http://www.not-normal-consulting.co.uk, or check the EIOPA document in references.

Author(s)

Phil Joubert <phil.joubert@not-normal-consulting.co.uk>

References


Examples

dfInstruments <- data.frame(c("SWAP", "SWAP"), c(1,10), c(1,1), c(0.025, 0.05))
colnames( dfInstruments ) <- c( "Type", "Tenor", "Frequency", "Rate" )
Curve <- fFitSmithWilsonYieldCurveToInstruments( dfInstruments, 0.04, 0.1 )
plot( Curve )
**fCreateCashflowMatrix**

*Returns the matrix of cashflows for the list of instruments*

---

**Description**

Returns the matrix of cashflows for the list of instruments

**Usage**

\[
fCreateCashflowMatrix(dfInstruments)
\]

**Arguments**

- `dfInstruments` A set of market instruments as a dataframe with columns Type, Tenor, Frequency and Rate with Type in (LIBOR, SWAP), Tenor the instrument maturity in years and rate the rate per annum

---

**fCreateKernelMatrix**

*Create the matrix of kernel functions*

---

**Description**

Creates a J x J matrix \( [w(u_i, u_j)] \) where J is the number of cashflow times in the calibration set

**Usage**

\[
fCreateKernelMatrix(times, fKernel)
\]

**Arguments**

- `times` a vector of cashflow times
- `fKernel` a kernel to apply (a function of times x times returning a matrix)
fCreateTimeVector  

*Extract a vector of cashflow times in years from a list of instruments*

**Description**

Assumes that LIBOR tenor is in days, with 365 days per year. Assumes that SWAPs are semi-annual

Returns a vector of all unique cashflow times in years

**Usage**

```r
fCreateTimeVector(dfInstruments)
```

**Arguments**

- `dfInstruments` A dataframe of instruments with at least columns Type and Tenor

---

fFitKernelWeights  

*Solve for the vector xi of kernel weights*

**Description**

Solve for the vector xi of kernel weights

**Usage**

```r
fFitKernelWeights(CashflowMatrix, KernelFunctionMatrix, MarketValueVector, BaseZeroVector)
```

**Arguments**

- `CashflowMatrix` A matrix of all cashflows, instruments in rows, times in columns
- `KernelFunctionMatrix` A matrix of kernel function values
- `MarketValueVector` A vector of market values of the instruments
- `BaseZeroVector` A vector of "base" values for the zeros
fFitSmithWilsonYieldCurve

*Construct the Smith-Wilson yield curve*

**Description**

Constructs the SmithWilson ZCB function based on the given market inputs and parameter choices.

**Usage**

```r
fFitSmithWilsonYieldCurve(TimesVector, CashflowMatrix, MarketValueVector, ufr, alpha)
```

**Arguments**

- **TimesVector**
  A vector of all cashflow times
- **CashflowMatrix**
  A matrix of all cashflows, instruments in rows, times in columns
- **MarketValueVector**
  A vector of market values of the instruments
- **ufr**
  The Ultimate Forward Rate (UFR) of the Smith-Wilson kernel
- **alpha**
  The rate of reversion of forward rates to the UFR in the Smith-Wilson kernel

**Value**

A list containing:

- "P" a function of time which gives the ZCB price to that term
- "xi" the vector of weights applied to the kernel functions to obtain the ZCB price
- "K" the (compound) kernel vector

---

fFitSmithWilsonYieldCurveToInstruments

*Construct the Smith-Wilson yield curve*

**Description**

Constructs the SmithWilson ZCB function based on the given market inputs and parameter choices. Primarily a convenience wrapper around other package functions.

**Usage**

```r
fFitSmithWilsonYieldCurveToInstruments(InstrumentSet, ufr, alpha)
```
Arguments

InstrumentSet  A set of market instruments as a dataframe with columns
  • "Type" One of (LIBOR, SWAP)
  • "Tenor" The instrument maturity in years
  • "Frequency" The payment frequency (ignored for Type=="LIBOR")
  • "Rate" The coupon rate per annum in percent

ufr  The Ultimate Forward Rate (UFR) of the Smith-Wilson kernel
alpha  The rate of reversion of forward rates to the UFR in the Smith-Wilson kernel

Value

a list containing:
  • "P" a function of time which gives the ZCB price to that term
  • "xi" the vector of weights applied to the kernel functions to obtain the ZCB price
  • "K" the (compound) kernel vector

ffityieldcurve  Constructs the ZCB function based on the given market inputs and a specific kernel and base function

Description

Constructs the ZCB function based on the given market inputs and a specific kernel and base function

Usage

ffityieldcurve(TimesVector, CashflowMatrix, MarketValueVector, fKernel, fBase)

Arguments

TimesVector  A vector of all cashflow times
CashflowMatrix  A matrix of all cashflows, instruments in rows, times in columns
MarketValueVector  A vector of market values of the instruments
fKernel  a function of two times used as the Kernel "basis" function
fBase  a function giving the base level of the curve

Value

a list comprising elements: a function of time which gives the ZCB price to that time
fGetCashflowsLibor

| fGetCashflowsLibor | Gets the cashflow schedule for a LIBOR agreement |

Description

Gets the cashflow schedule for a LIBOR agreement

Usage

fGetCashflowsLibor(dfInstrument)

Arguments

dfInstrument | A set of market instruments as a dataframe with columns Type, Tenor and Rate with Type in (LIBOR, SWAP), Tenor the instrument maturity in years and rate the rate per annum

fGetCashflowsSwap

| fGetCashflowsSwap | Gets the cashflow schedule for a swap |

Description

Gets the cashflow schedule for a swap

Usage

fGetCashflowsSwap(dfInstrument)

Arguments

dfInstrument | A set of market instruments as a dataframe with columns Type, Tenor and Rate with Type in (LIBOR, SWAP), Tenor the instrument maturity in years and rate the rate per annum
fGetTimesLibor  
*Extract the payment date of a LIBOR agreement in years*

**Description**

Extract the payment date of a LIBOR agreement in years

**Usage**

```
fGetTimesLibor(dfInstrument)
```

**Arguments**

- `dfInstrument` A dataframe of instruments with at least columns Type and Tenor

---

fGetTimesSwap  
*Extract the payment dates of a Swap agreement in years*

**Description**

Extract the payment dates of a Swap agreement in years

**Usage**

```
fGetTimesSwap(dfInstrument)
```

**Arguments**

- `dfInstrument` A dataframe of instruments with at least columns Type and Tenor

---

fWilson  
*Wilson function*

**Description**

Acts as a kernel for regression

**Usage**

```
fWilson(t, u, ufr, alpha)
```

**Arguments**

- `t` a time
- `u` another time
- `ufr` the ultimate forward rate
- `alpha` the speed of reversion to the ultimate forward rate
lines.SmithWilsonYieldCurve

Plot generic for SmithWilsonYieldCurve objects

Description
Plot generic for SmithWilsonYieldCurve objects

Usage
```r
## S3 method for class 'SmithWilsonYieldCurve'
lines(x, y, ..., 
    aspect = c("cts", "zero"))
```

Arguments
- **x**: An object of class SmithWilsonYieldCurve or a vector of terms to evaluate the curve at
- **y**: Optionally an object of class SmithWilsonYieldCurve
- **aspect**: either "cts" for continuously compounded spot rates, or "zero" for ZCB prices
- **...**: other arguments to pass to the default lines function

plot.SmithWilsonYieldCurve

Plot generic for SmithWilsonYieldCurve objects

Description
Plot generic for SmithWilsonYieldCurve objects

Usage
```r
## S3 method for class 'SmithWilsonYieldCurve'
plot(x, y, ..., 
    aspect = c("cts", "zero"))
```

Arguments
- **x**: An object of class SmithWilsonYieldCurve or a vector of terms to evaluate the curve at
- **y**: Optionally an object of class SmithWilsonYieldCurve
- **aspect**: either "cts" for continuously compounded spot rates, or "zero" for ZCB prices
- **...**: other arguments to pass to the default plot function
points.SmithWilsonYieldCurve

Plot generic for SmithWilsonYieldCurve objects

Description

Plot generic for SmithWilsonYieldCurve objects

Usage

```r
## S3 method for class 'SmithWilsonYieldCurve'
points(x, y, ...,
        aspect = c("cts", "zero"))
```

Arguments

- `x` An object of class SmithWilsonYieldCurve or a vector of terms to evaluate the curve at
- `y` Optionally an object of class SmithWilsonYieldCurve
- `aspect` either "cts" for continously compounded spot rates, or "zero" for ZCB prices
- `...` other arguments to pass to the default plot function
Index

fCreateCashflowMatrix, 3
fCreateKernelMatrix, 3
fCreateTimeVector, 4
fFitKernelWeights, 4
fFitSmithWilsonYieldCurve, 5
fFitSmithWilsonYieldCurveToInstruments, 5
fFitYieldCurve, 6
fGetCashflowsLibor, 7
fGetCashflowsSwap, 7
fGetTimesLibor, 8
fGetTimesSwap, 8
fWilson, 8

lines.SmithWilsonYieldCurve, 9
plot.SmithWilsonYieldCurve, 9
points.SmithWilsonYieldCurve, 10

SmithWilsonYieldCurve
   (SmithWilsonYieldCurve-package), 2
SmithWilsonYieldCurve-package, 2