Package 'termstrc'

October 18, 2009

Type Package

Title Zero-coupon Yield Curve Estimation

Version 1.1.1

Date 2009-10-16

Author Robert Ferstl, Josef Hayden

Maintainer Josef Hayden <josef.hayden@wiwi.uni-regensburg.de>

Description Zero-coupon yield curves and spread curves are important inputs for various financial models, e.g. pricing of securities, risk management, monetary policy issues. Since zero-coupon rates are rarely directly observable, they have to be estimated from market data. The literature broadly distinguishes between parametric and spline-based estimation methods for the zero-coupon yield curve. Our package consists of several widely-used approaches, i.e. the parametric Nelson and Siegel (1987) method with the Svensson (1994) extension, and the McCulloch (1975) cubic splines approach. Extensive summary statistics and plots are provided to compare the results of the different estimation methods.

Depends R (>= 2.9.0)

License GPL

URL http://R-Forge.R-project.org/projects/termstrc/

Repository CRAN

Date/Publication 2009-10-18 10:23:06

R topics documented:

`termstrc-package` *Zero-coupon Yield Curve Estimation*

Description

Zero-coupon yield curves and spread curves are important inputs for various financial models, e.g. pricing of securities, risk management, monetary policy issues. Since zero-coupon rates are rarely directly observable, they have to be estimated from market data. The literature broadly distinguishes between parametric and spline-based estimation methods for the zero-coupon yield curve. Our package consists of several widely-used approaches, i.e. the parametric Nelson and Siegel (1987) method with the Svensson (1994) extension, and the McCulloch (1975) cubic splines approach. Extensive summary statistics and plots are provided to compare the results of the different estimation methods.

References

Bank for International Settlements (2005). Zero-Coupon Yield Curves: Technical Documentation. *BIS Papers,* ***No. 25***.

Robert R. Bliss (2007): Testing Term Structure Estimation Methods. *Advances in Futures and Options Research,* ***9*** *197–232*.

David Bolder and David Streliski (1999): Yield Curve Modelling at the Bank of Canada. *Bank of Canada Technical Report,* ***No. 84***.

Alois Geyer and Richard Mader (1999): Estimation of the Term Structure of Interest Rates - A Parametric Approach. *OeNB Working Paper,* ***No. 37***.

Michalis Ioannides (2003): A Comparison of Yield Curve Estimation Techniques Using UK Data. *Journal of Banking & Finance,* ***27*** *1–26*.

J. Huston McCulloch (1971): Measuring the Term Structure of Interest Rates. *The Journal of Business,* ***44*** *19–31*.

J. Huston McCulloch (1975): The Tax-Adjusted Yield Curve. *The Journal of Finance,* ***30*** *811–830*.

Charles R. Nelson and Andrew F. Siegel (1987): Parsimonious Modeling of Yield Curves. *The Journal of Business*, **60(4)**:473–489.

Lars E.O. Svensson (1994): Estimating and Interpreting Forward Interest Rates: Sweden 1992 -1994. *Technical Reports 4871, National Bureau of Economic Research*.

`aabse` *Average Absolute Mean Error*

Description

Calculation of the average absolute mean error (AABSE). The AABSE may also be called mean absolute error (MAE).

Usage

```
aabse(actual, estimated)
```

Arguments

`actual`	vector, consisting of the observed values.
`estimated`	vector, consisting of the estimated values.

Details

Calculation of the AABSE according to the formula:

$$\text{AABSE} = \frac{1}{m}|\boldsymbol{\epsilon}|\boldsymbol{\iota},$$

whereas $\boldsymbol{\epsilon}$ is the vector of the yield or price errors of the bonds and $\boldsymbol{\iota}$ is a column vector filled with ones. m is the number of bonds, for which $\boldsymbol{\epsilon}$ has been calculated.

See Also

rmse

---

bond_prices *Bond Price Calculation*

Description

Function for the calculation of bond prices according to the chosen approach (Nelson and Siegel or Svensson) based on the cashflows and maturities of the bonds.

Usage

```
bond_prices(method = "Nelson/Siegel", beta, m, cf)
```

Arguments

method	defines the desired method, "Nelson/Siegel" for the Nelson/Siegel approach or "Svensson" for the Svensson approach.
beta	parameter vector, is linked to the chosen approach.
m	maturities matrix, consists of the maturity dates which are appended to the cashflows of the bonds.
cf	cashflows matrix.

Value

Returns a list with:

spot_rates	spot rates.
discount_factors	discount factors.
bond_prices	bond prices.

See Also

svensson, nelson_siegel

bond_yields *Bond Yield Calculation*

Description

Function for the calculation of bond yields.

Usage

```
bond_yields(cashflows, m, searchint = c(-1, 1), tol = 1e-10)
```

Arguments

cashflows	matrix with the bonds cashflows.
m	maturity matrix.
searchint	search interval for root finding.
tol	desired accuracy for function uniroot.

Value

The function returns a matrix with the bond yields and the associated maturities.

See Also

uniroot

corpbonds *Corporate Bonds*

Description

Corporate bonds

Usage

```
data(corpbonds)
```

Details

The data set eurobonds consists of bonds of the rating classes AAA ,AA+, AA, AA-, A+, A, A-, BBB+, BBB, BBB-

Note

If you use your own data set, make sure that the structure is identical to the provided data sets. Use the function `str()` to explore the data set.

Every element of the list except $RATING and $NAME is required for the estimation. The elements $COUPONRATE and $PRICE are not necessarily required. The calculation of the accrued interest is not implemented and has therefore be provided by the user. For the provided data sets the list element $ACCRUED includes the accrued interest.

See Also

eurobonds, govbonds

Examples

```
data(corpbonds)
str(corpbonds)

# The following code may use this to generate an empty data set,
# which can be filled with bond data:

ISIN <- vector()
MATURITYDATE <- vector()
STARTDATE <- vector()
COUPONRATE <- vector()
PRICE <- vector()
ACCRUED <- vector()

CFISIN <- vector()
CF <- vector()
DATE <- vector()

CASHFLOWS <- list(CFISIN,CF,DATE)
names(CASHFLOWS) <- c("ISIN","CF","DATE")

TODAY <- vector()

mycountry1 <- list(ISIN,MATURITYDATE,STARTDATE,
                   COUPONRATE,PRICE,ACCRUED,CASHFLOWS,TODAY)
mycountry2 <- list(ISIN,MATURITYDATE,STARTDATE,
                   COUPONRATE,PRICE,ACCRUED,CASHFLOWS,TODAY)

names(mycountry1) <- c("ISIN","MATURITYDATE","STARTDATE","COUPONRATE",
                       "PRICE","ACCRUED","CASHFLOWS","TODAY")
names(mycountry2) <- c("ISIN","MATURITYDATE","STARTDATE","COUPONRATE",
                       "PRICE","ACCRUED","CASHFLOWS","TODAY")

mybonds <- list(mycountry1,mycountry2)

names(mybonds) <- c("mycountry1","mycountry2")
```

create_cashflows_matrix

Cashflows Matrix Creation

Description

Creates a matrix of cashflows for a specified group of bonds. The number of rows is the number of cashflows for the bond with the longest maturity.

Usage

```
create_cashflows_matrix(group, include_price = FALSE)
```

Arguments

group	group name.
include_price	if TRUE the dirty price is included (default: FALSE).

Value

Returns a matrix which consits of the calculated cashflows.

See Also

create_maturities_matrix

create_maturities_matrix

Maturity Matrix Creation

Description

Creates a matrix of maturities for a specified group of bonds. The number of rows is the number of cashflows for the bond with the longest maturity.

Usage

```
create_maturities_matrix(group, include_price = FALSE)
```

Arguments

group	name of the group of bonds (default: FALSE).
include_price	if TRUE the dirty price is included.

Value

The maturity matrix is returned.

See Also

create_cashflows_matrix

---

duration *Duration, modified Duration and Duration based Weights*

Description

The function calculates the Macauly duration, modified duration and duration based weights.

Usage

```
duration(cf_p, m_p, y)
```

Arguments

cf_p	cashflows matrix including the prices of the bonds.
m_p	maturity matrix, the first row is filled with zeros.
y	yields of the bonds.

Details

The duration vector is calculated using the following formula:

$$\boldsymbol{d} = \frac{\boldsymbol{\iota}'(\boldsymbol{C} \cdot \boldsymbol{M} \cdot \boldsymbol{D})}{\boldsymbol{\iota}'(\boldsymbol{C} \cdot \boldsymbol{D})},$$

whereas $\boldsymbol{C}$ is the cashflow matrix and $\boldsymbol{M}$ is the maturity matrix. $\boldsymbol{\iota}$ is a column vector filled with ones. $(\cdot)$ denotes a elementwise matrix mulitplication and "$'$" the transpose of a vector (matrix).

The weight ω_j for one bond j is defined as

$$\omega_j = \frac{\frac{1}{d_j}}{\sum_{i=1}^{m} \frac{1}{d_i}},$$

where d_j is the duration of the j-th bond.

Value

The function returns a matrix with three columns, i.e. duration, modified duration and duration based weights.

eurobonds *European Government Bonds*

Description

European government bonds.

Usage

```
data(eurobonds)
```

Details

The data set `eurobonds` consists of German, Austrian, Italian and Hungarian government bonds.

Note

If you use your own data set, make sure that the structure is identical to the provided data sets. Use the function `str()` to explore the data set.

See Also

`corpbonds`, `govbonds`

Examples

```
data(corpbonds)
str(corpbonds)
```

forwardrates *Forward Rate Calculation*

Description

Calculates forward rates according to the Nelson/Siegel or Svensson approach.

Usage

```
forwardrates(method, beta, m)
```

Arguments

`method`	method used for the forward rate calculation (`Nelson/Siegel` or `Svensson`).
`beta`	parameters corresponding to the chosen method.
`m`	one maturity or a vector of maturities.

Value

The function returns a vector with the calculated forward rates.

See Also

`fwr_ns`, `fwr_sv`

`fwr_ns` *Forward Rate Calculation according to Nelson/Siegel*

Description

Calculate forward rates according to the Nelson/Siegel (1987) approach.

Usage

```
fwr_ns(beta, m)
```

Arguments

`beta`	parameter vector $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \tau_1)$.
`m`	maturity or maturity vector.

Details

The forward rate for a maturity m is calculated using the following relationsship:

$$f(m, \boldsymbol{\beta}) = \beta_0 + \beta_1 \exp\left(-\frac{m}{\tau_1}\right) + \beta_2 \left[\left(\frac{m}{\tau_1}\right) \exp\left(-\frac{m}{\tau_1}\right)\right].$$

Value

The function returns the calculated forward rate (vector).

References

Charles R. Nelson and Andrew F. Siegel (1987): Parsimonious Modeling of Yield Curves. *The Journal of Business*, **60(4)**:473–489.

See Also

`fwr_sv`, `forwardrates`

`fwr_sv` *Forward Rate Calculation according to Svensson (1994).*

Description

Calculate forward rates according to Svensson (1994).

Usage

```
fwr_sv(beta, m)
```

Arguments

`beta`	parameter vector $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \tau_1, \beta_3, \tau_2)$.
`m`	maturity or vector of maturities.

Details

The forward rate for a maturity m is calculated according to the following formula:

$$f(m, \boldsymbol{\beta}) = \beta_0 + \beta_1 \exp\left(-\frac{m}{\tau_1}\right) + \beta_2 \left[\left(\frac{m}{\tau_1}\right) \exp\left(-\frac{m}{\tau_1}\right)\right] + \beta_3 \left[\left(\frac{m}{\tau_2}\right) \exp\left(-\frac{m}{\tau_2}\right)\right].$$

Value

Returns the a vector with the calculated forward rate (vector).

References

Lars E.O. Svensson (1994): Estimating and Interpreting Forward Interest Rates: Sweden 1992 -1994. *Technical Reports 4871, National Bureau of Economic Research*.

See Also

`fwr_ns`, `forwardrates`

gi *Cubic Functions*

Description

Calculation of the cubic functions according to an approach of McCulloch (1975).

Usage

```
gi(t, T, i, s)
```

Arguments

t	maturity.
T	knot points.
i	index.
s	number of basis functions.

References

J. Huston McCulloch (1975): The Tax-Adjusted Yield Curve. *The Journal of Finance,* ***30*** *811–830.*

govbonds *European Government Bonds*

Description

European government bonds.

Usage

```
data(eurobonds)
```

Details

The data set `bonds` consists of German, Austrian, French, Belgian, Finnish and Spanish, government bonds obtained from Thomson Datastream.

Note

If you use your own data set, make sure that the structure is identical to the provided data sets. Use the function `str()` to explore the data set.

See Also

`eurobonds`, `corpbonds`

Examples

```
data(govbonds)
str(govbonds)
```

---

impl_fwr *Implied Forward Rate Calculation*

Description

Calculates the implied forward rates from given spot rates.

Usage

```
impl_fwr(m, s)
```

Arguments

m	maturity vector.
s	spot rate vector.

Details

Implied forward rates can be calculated using the following relationship:

$$f(t', T) = \frac{s(m_T)m_T - s(m_{t'})m_{t'}}{m_T - m_{t'}},$$

whereas $s(m_T), s(m_{t'})$ is the spot rate for a maturity $m_T, m_{t'}$ respectively.

Value

The function returns the calculated forward rate vector.

---

loss_function *Loss Function used for the Term Structure Estimation*

Description

The loss function defines the objective function used for the optimisation. In case of term structure estimation the objective function is either the minimsation of the (weighted) squared price or the yield errors.

Usage

```
loss_function(p, phat, omega, weights)
```

Arguments

p	vector of observed prices (yields).
phat	vector of estimated prices (yields).
omega	weights vector.
weights	if "none" the squared deviation of p and phat is not weighted, use "duration" for a duration weighted optimisation.

---

maturity_range *Restricting a Bond Dataset*

Description

The function restricts a bond data set to a specified maturity range.

Usage

```
maturity_range(bonddata, lower, upper)
```

Arguments

bonddata	bond data set.
lower	lower bound of maturity spectrum.
upper	upper bound of maturity spectrum.

---

nelson_estim *Zero-coupon Yield Curve Estimation with the Nelson/Siegel, Svensson Method*

Description

Zero-coupon yield curve estimation with the parametric Nelson and Siegel (1987) method and the Svensson (1994) extension.

Usage

```
nelson_estim(group,
             bonddata,
             matrange = "all",
             method = "Nelson/Siegel",
             fit = "prices",
             weights = "duration",
             startparam,
             control = list(eval.max = 1000, iter.max= 500))
```

Arguments

group	vector defining the group of bonds used for the estimation, e.g. c("GERMANY","AUSTRIA").
bonddata	a data set of bonds in list format.
matrange	use "all" for no restrictions, or restrict the maturity range used for the estimation with c(lower,upper).
method	"Nelson/Siegel" or "Svensson".
fit	use "prices" ("yields") for minimizing the squared pricing (yield) errors.
weights	weighting of the errors used in the optimization ("none" or "duration").
startparam	matrix of start parameters, for the Nelson/Siegel (Svensson) method 4 (6) parameters for each each group are required (one row per group).
control	list of control parameters for the function nlminb().

Details

group The first element of the vector will be used as the reference country for the spread curve calculation. group can be either a vector of bond groups or a scalar.

bonddata The package is designed to work with a certain list data structure. For more information use the function str() to explore the structure of the example data sets.

startparam A matrix of start parameters must be provided for the estimation. Due to the possible existence of multiple local minima, a great effort should be invested in finding a good start parameter set. For rolling estimation procedures, the previous optimal parameters should be used as inputs for the next iteration.

control Please refer to the documentation of the function nlminb.

Value

The function nelson_estim returns an object of the class "nelson". The object contains the following items (mainly lists):

group	group of bonds (e.g. countries) used for the estimation.
matrange	"none" or a vector with the maturity range.
method	estimation method ("Nelson/Siegel" or "Svensson").
fit	objective function ("prices", or "yields").
weights	weighting of the errors used in the optimization ("none" or "duration").
n_group	length of object group, i.e. the number of countries.
spot	zero-coupon yield curves as object of the class "spot_curves".
spread	spread curves as object of the class "s_curves".
forward	forward curves as object of the class "fwr_curves".
discount	discount curves as object of the class "df_curves".
expoints	extrapolation points for Nelson/Siegel method.
cf	cashflow matrices.
m	maturity matrices.

p	dirty prices.
phat	estimated bond prices.
perrors	pricing errors and maturities as object of the class "error".
y	bond yields.
yhat	one list for each group with the theoretical bond yields calculated with the estimated bond prices phat.
yerrors	yield errors and maturities as object of the class "error".
opt_result	optimization results from nlminb, e.g. optimal parameters, convergence info.

Note

An error message concerning the function uniroot() is in general caused by wrongly specified start parameters.

For objects of the class "spot_curves", "s_curves", "df_curves", "fwr_curves", "error" appropriate plot methods are offered. For objects of the class "nelson" print, summary and plot methods are available. Another term structure estimation method is provided by the function splines_estim.

References

Charles R. Nelson and Andrew F. Siegel (1987): Parsimonious Modeling of Yield Curves. *The Journal of Business*, **60(4)**:473–489.

Lars E.O. Svensson (1994): Estimating and Interpreting Forward Interest Rates: Sweden 1992 -1994. *Technical Reports 4871, National Bureau of Economic Research*.

See Also

print.nelson, summary.nelson, plot.nelson, splines_estim, plot.spot_curves, plot.s_curves, plot.df_curves, plot.fwr_curves, plot.error, uniroot.

Examples

```
# load data set
data(eurobonds)

# define countries, for which the estimation
# of the zero-coupon yield curves will be carried out
group <- c("GERMANY", "AUSTRIA", "ITALY")

# define data set
bonddata <- eurobonds

# set maturity range
matrange <- c(2,12)

# define the used method
method <- "Nelson/Siegel"
```

```
# the weighted squared price errors will be minimized
fit <- "prices"
weights <- "duration"

# naive start parameters
b <- matrix(rep(c(0,0,0, 1),3),nrow=3,byrow=TRUE)
rownames(b) <- group
colnames(b) <- c("beta0","beta1","beta2","tau1")

# perform estimation
x <- nelson_estim(group, bonddata, matrange,
                  method, fit, weights, startparam=b)

# prints the obtained parameters of the estimation
print(x)

# goodness of fit measures
summary(x)

# plots the zero-coupon yield curves for each country
plot(x,errors="none")

# plots all zero-coupon yield curves togther
plot(x,multiple=TRUE, errors="none")

# spread curve splot
plot(x,ctype="spread",errors="none")

# price error plot for all countries
plot(x,ctype="none")
```

`nelson_siegel` *Spot Rate Function according to Nelson/Siegel*

Description

This function calculates the spot rates for certain maturity dates and a parameter vector according to Nelson/Siegel (1987).

Usage

```
nelson_siegel(beta, m)
```

Arguments

`beta`	a vector of parameters $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \tau_1)$.
`m`	one maturity (or a vector of maturities).

Details

The spot rate according to Nelson/Siegel for a maturity m is defined as:

$$s(m, \boldsymbol{\beta}) = \beta_0 + \beta_1 \frac{1 - \exp(-\frac{m}{\tau_1})}{\frac{m}{\tau_1}} + \beta_2 \left(\frac{1 - \exp(-\frac{m}{\tau_1})}{\frac{m}{\tau_1}} - \exp(-\frac{m}{\tau_1}) \right).$$

Value

Returns a vector consisting of the calculated spot rates.

References

Charles R. Nelson and Andrew F. Siegel (1987): Parsimonious Modeling of Yield Curves. *The Journal of Business*, **60(4)**:473–489.

Examples

```
nelson_siegel(rep(0.01,4),1:30)
```

---

plot.cubicsplines *S3 Plot Method for Cubic Splines*

Description

S3 plot method for an object of the class `"cubicsplines"`.

Usage

```
## S3 method for class 'cubicsplines':
plot(x, matrange = c(min(mapply(function(i) min(x$y[[i]][, 1]),
        seq(x$n_group))), max(mapply(function(i) max(x$y[[i]][, 1]),
        seq(x$n_group)))), multiple = FALSE,
        ctype = "spot", lwd=2, lty=1, type = "l",
        errors = "price", inset = c(0.8, 0.1), ask=TRUE, ...)
```

Arguments

`x`	object of the class `"cubicsplines"`.
`matrange`	maturity range for the plot, e.g. c(2,10).
`multiple`	if `TRUE` all curves are plotted together (default: `FALSE`).
`ctype`	parameter setting for the desired curve type, `"spot"` (`"forward"`, `"discount"`, `"spread"`) for the spot rate (forward rate, discount factor, spread) curves. Use `"none"` if no curve plot is desired.
`errors`	Specify the type of the error plot. If `"price"` (`"yield"`) the price (yield) errors will be plot. Use `"none"` if no error plot is desired.

lwd	the line width, for details see par.
lty	the line type, for details see par.
type	1-character string giving the type of plot desired, for details see plot.default.
inset	inset distance(s) from the margins as a fraction of the plot region, for details see legend.
ask	if TRUE (and the R session is interactive) the user is asked for input, before a new figure is drawn, see par for details.
...	other graphical parameters, see par.

Details

Depending on the choice of the curve type ("spot", "forward", "discount", "spread") the corresponding curves will be plot. Either separately or together (mulitple = TRUE). If the curves are plotted separately also the knot points used for the estimation of the cubic splines and the yield-to-maturities will be plotted. In addition, with a zero-coupon yield curve plot the 95 % confidence interval of the curve will be plot. To ease the analysis of the goodness of the estimation, serveral error plots for the yield and price error are offered.

See Also

plot.df_curves, plot.error, plot.fwr_curves, plot.ir_curve, plot.s_curves, plot.spot_curves, plot.cubicsplines

---

plot.df_curves *S3 Plot Method*

Description

S3 plot method for an object of the class "df_curves".

Usage

```
## S3 method for class 'df_curves':
plot(x, multiple = FALSE,
        ylim = c(range(mapply(function(i) range(x[[i]][, 2]),
        seq(x)))) * 100, xlim = c(), type = "l", lty = 1, lwd = 2,
        expoints = NULL, ylab = "Discount factor (percent)",
        xlab = "Maturity (years)", main = "Discount factor curves", ...)
```

Arguments

x	object of the class "df_curves".
multiple	if TRUE all discount factor curves are plotted togehter (default: FALSE).
ylim	the y limits of the plot, for details see plot.default.
xlim	the x limits of the plot, for details see plot.default.

type	1-character string giving the type of plot desired, for details see plot.default.
lty	the line type, for details see par.
lwd	the line width, for details see par.
expoints	extrapolation points (default: NULL).
ylab	a label for the y axis, for details see plot.default.
xlab	a label for the x axis, for details see plot.default.
main	a main title for the plot, for details see title.
...	other graphical parameters, see par.

See Also

plot.fwr_curves, plot.s_curves, plot.spot_curves

---

plot.error *S3 Plot Method*

Description

S3 plot method for an object of the class error.

Usage

```
## S3 method for class 'error':
plot(x, type = "b", main = "", mar = c(7, 6, 6, 2) + 0.1,
        oma = c(4, 2, 2, 2) + 0.1, ylab = "Error", ...)
```

Arguments

x	object of the class error.
type	1-character string giving the type of plot desired, for details see plot.default.
main	a main title for the plot, for details see title.
mar	A numerical vector of the form 'c(bottom, left, top, right)' which gives the number of lines of margin to be specified on the four sides of the plot, for details see par.
oma	A vector of the form 'c(bottom, left, top, right)' giving the size of the outer margins in lines of text.
ylab	a label for the y axis, for details see plot.default.
...	other graphical parameters, see par.

Details

Absolute yield and price errors as a resuslt of the term structure estimation can be plotted. The scaling of the x axis depends on the maturity of the bonds, each bond is labeled with its ISIN number. The error plots seems especially useful in identifying misspriced bonds. For removing them, the function rm_bond may be applied.

See Also

rm_bond

---

plot.fwr_curves *S3 Plot Method*

Description

S3 plot method for an object of the class "fwr_curves".

Usage

```
## S3 method for class 'fwr_curves':
plot(x, multiple = FALSE,
        ylim = c(range(mapply(function(i) range(x[[i]][, 2]),
        seq(x)))) * 100, xlim = c(), type = "l", lty = 1,
        lwd = 2, expoints = NULL, ylab = "Forward rate (percent)",
        xlab = "Maturity (years)", main = "Forward rate curves", ...)
```

Arguments

x	object of the class "fwr_curves".
multiple	if TRUE all forward rate curves are plotted together (default: FALSE).
ylim	the y limits of the plot, for details see plot.default.
xlim	the x limits of the plot, for details see plot.default.
type	1-character string giving the type of plot desired, for details see plot.default.
lty	the line type, for details see par.
lwd	the line width, for details see par.
expoints	extrapolation points (default: NULL).
ylab	a label for the y axis, for details see plot.default.
xlab	a label for the x axis, for details see plot.default.
main	a main title for the plot, for details see title.
...	other graphical parameters, see par.

See Also

plot.df_curves, plot.s_curves, plot.spot_curves

plot.ir_curve *S3 Plot Method*

Description

S3 plot method for an object of the class "ir_curve".

Usage

```
## S3 method for class 'ir_curve':
plot(x, ylim = c(), xlim = c(), lwd = 2, type = "l",
        xlab = "Maturity (years)", ylab = "Percent",
        col = "steelblue", lty = 1, ...)
```

Arguments

x	object of the class "ir_curve".
ylim	the y limits of the plot, for details see plot.default.
xlim	the x limits of the plot, for details see plot.default.
lwd	the line width, for details see par.
type	1-character string giving the type of plot desired, for details see plot.default.
xlab	a label for the x axis, for details see plot.default.
ylab	a label for the y axis, for details see plot.default.
col	the colors for lines and points.
lty	the line type, for details see par.
...	other graphical parameters, see par.

plot.nelson *S3 Plot Method*

Description

S3 plot method for an object of the class "nelson".

Usage

```
## S3 method for class 'nelson':
plot(x, matrange = c(min(mapply(function(i) min(x$y[[i]][, 1]),
        seq(x$n_group))),max(mapply(function(i) max(x$y[[i]][, 1]),
        seq(x$n_group)))), multiple = FALSE, expoints = unlist(x$expoints),
        ctype = "spot",errors = "price", lwd = 2, lty = 1, type ="l",
        inset = c(0.8, 0.1), ask = TRUE, ...)
```

Arguments

x	object of the class "nelson".
matrange	maturity range for the plot, e.g. c(2,10).
multiple	if TRUE all curves are plotted together (default: FALSE).
expoints	extrapolation points (default: NULL).
ctype	parameter setting for the desired curve type, "spot" ("forward", "discount", "spread") for the spot rate (forward rate, discount factor, spread) curves. Use "none" if no curve plot is desired.
errors	Specify the type of the error plot. If "price" ("yield") the pricing (yield) errors will be plotted . Use "none" if no error plot is desired.
lwd	the line width, for details see par.
lty	the line type, for details see par.
type	1-character string giving the type of plot desired, for details see plot.default.
inset	inset distance(s) from the margins as a fraction of the plot region, for details see legend.
ask	if TRUE (and the R session is interactive) the user is asked for input, before a new figure is drawn, see par for details.
...	other graphical parameters, see par.

Details

Depending on the choice of the curve type ("spot", "forward", "discount", "spread") the corresponding curves will be plot. Either separately or together (multiple = TRUE). If the curves are plotted together a dashed line indicates that the corresponding curve has been extrapolated. In addition, with a separate zero-coupon yield curve plot the yield-to-maturity will be plot. To ease the analysis of the goodness of the estimation, several error plots are offered.

See Also

plot.df_curves, plot.error, plot.fwr_curves, plot.ir_curve, plot.s_curves, plot.spot_curves, plot.nelson

---

plot.spot_curves *S3 Plot Method*

Description

S3 plot method for an object of the class "spot_curves".

Usage

```
## S3 method for class 'spot_curves':
plot(x, multiple = FALSE,
        ylim = c(range(mapply(function(i) range(x[[i]][, 2]),
        seq(x)))) * 100, xlim = c(), type = "l", lty = 1,
        lwd = 2, expoints = NULL, ylab = "Zero-coupon yields (percent)",
        xlab = "Maturity (years)", main = "Zero-coupon yield curves", ...)
```

Arguments

x	object of the class "spot_curves".
multiple	if TRUE all zero-coupon yield curves are plotted together (default: FALSE).
ylim	the y limits of the plot, for details see plot.default.
xlim	the x limits of the plot, for details see plot.default.
type	1-character string giving the type of plot desired, for details see plot.default.
lty	the line type, for details see par.
lwd	the line width, for details see par .
expoints	extrapolation points (default: NULL.)
ylab	a label for the y axis, for details see plot.default.
xlab	a label for the x axis, for details see plot.default.
main	a main title for the plot, for details see title.
...	other graphical parameters, see par.

See Also

plot.df_curves, plot.fwr_curves, plot.s_curves

---

plot.s_curves *S3 Plot Method*

Description

S3 plot method for an object of the class "s_curves".

Usage

```
## S3 method for class 's_curves':
plot(x, xlim = c(range(mapply(function(i) range(x[[i]][, 1]), seq(x)))),
        ylim = c(range(mapply(function(i) range(x[[i]][, 2]),
        seq(x)))) * 10000, expoints = NULL,
        xlab = "Maturity (years)", ylab = "Spread (basis points)",
        lwd = 2, lty = 1, main = "Spread curves", ...)
```

Arguments

x	object of the class "s_curves".
ylim	the y limits of the plot, for details see plot.default.
xlim	the x limits of the plot, for details see plot.default.
lty	the line type, for details see par.
lwd	the line width, for details see par.
expoints	extrapolation points (default: NULL).
ylab	a label for the y axis, for details see plot.default.
xlab	a label for the x axis, for details see plot.default.
main	a main title for the plot, for details see title.
...	other graphical parameters, see par.

Details

The spread curves (the difference of zero-coupon yield curves) are plotted, if at least two groups of bonds were specified.

See Also

plot.df_curves, plot.fwr_curves, plot.spot_curves

---

print.cubicsplines *S3 Print Method for Cubicsplines*

Description

S3 print method for an object of the class "cubicsplines".

Usage

```
## S3 method for class 'cubicsplines':
print(x,...)
```

Arguments

x	object of the class "cubicsplines".
...	other arguments.

Details

The print method for an object of the class "cubicsplines" prints the results of the regression analysis. (i.e. the parameter of the cubic spline functions).

See Also

plot.cubicsplines, summary.cubicsplines

print.nelson *S3 Print Method*

Description

S3 print method for an object of the class "nelson".

Usage

```
## S3 method for class 'nelson':
print(x,...)
```

Arguments

x	object of the class "nelson"
...	other arguments.

Details

The print method for an object of the class "nelson" prints important input parameters of the optimisation and the results of the optimisation (i.e. the optimal parameter vector).

See Also

plot.nelson, summary.nelson

print.summary.cubicsplines *S3 Print Method*

Description

S3 print method for an object of the class "summary.cubicsplines".

Usage

```
## S3 method for class 'summary.cubicsplines':
print(x, ...)
```

Arguments

x	object of the class "summary.nelson".
...	other arguments.

See Also

print.summary.nelson

print.summary.nelson *S3 Print Method*

Description

S3 print method for an object of the class `"summary.nelson"`.

Usage

```
## S3 method for class 'summary.nelson':
print(x, ...)
```

Arguments

`x`	object of the class `"summary.nelson"`.
`...`	other arguments.

See Also

`print.summary.cubicsplines`

rmse *Root Mean Squared Error*

Description

Calulates the root mean squared error (RMSE).

Usage

```
rmse(actual, estimated)
```

Arguments

`actual`	vector, consisting of the observed values.
`estimated`	vector, consisting of the estimated values.

Details

Calculation of the RMSE according to the formula:

$$\text{RMSE} = \sqrt{\frac{1}{m}\boldsymbol{\epsilon}^2\boldsymbol{\iota}},$$

whereas $\boldsymbol{\epsilon}$ is the vector of the yield or price errors of the bonds and $\boldsymbol{\iota}$ is a column vector filled with ones. m is the number of bonds, for which $\boldsymbol{\epsilon}$ has been calculated.

See Also

aabse

---

rm_bond *Remove Bonds from a Dataset*

Description

Specified bonds and their associated data are removed from the bond data set.

Usage

```
rm_bond(bdata, ISIN, gr)
```

Arguments

bdata	bond data set.
ISIN	the ISIN numbers of the bonds to remove.
gr	the group where the bonds to be removed belong to.

Value

The function returns the new bond data set.

---

splines_estim *Discount Curve Estimation with McCulloch Cubic Splines*

Description

Discount curve estimation with the cubic splines approach by McCulloch (1975).

Usage

```
splines_estim(group,
              bonddata,
              matrange = "all")
```

Arguments

group	vector defining the group of bonds used for the estimation, e.g. c("GERMANY","AUSTRIA").
bonddata	a data set of bonds in list format.
matrange	use "all" for no restrictions, or restrict the maturity range used for the estimation with c(lower,upper).

Details

`group` The first element of the vector will be used as the reference country for the spread curve calculation. `group` can be either a vector of bond groups or a scalar.

`bonddata` The package is designed to work with a certain list data structure. For more information use the function `str()` to explore the structure of the example data sets.

Value

The function `splines_estim` returns an object of the class `"cubicsplines"`. The object contains the following items (mainly lists):

`group`	group of bonds (e.g. countries) used for the estimation.
`matrange`	`"none"` or a vector with the maturity range.
`n_group`	length of object `group`, i.e. the number of countries.
`knotpoints`	selected knot points for the cubic splines estimation.
`spot`	zero-coupon yield curves as object of the class `"spot_curves"`.
`spread`	spread curves as object of the class `"s_curves"`.
`forward`	forward curves as object of the class `"fwr_curves"`.
`discount`	discount curves as object of the class `"df_curves"`.
`cf`	cashflow matrices.
`m`	maturity matrices.
`p`	dirty prices.
`phat`	estimated bond prices.
`perrors`	pricing errors and maturities as object of the class `"error"`.
`y`	bond yields.
`yhat`	one list for each group with the theoretical bond yields calculated with the estimated bond prices `phat`.
`yerrors`	yield errors and maturities as object of the class `"error"`.
`alpha`	OLS coefficients of cubic splines estimation.
`regout`	OLS estimation results as object of the class `"lm"`.

Note

For objects of the class `"spot_curves"`, `"s_curves"`, `"df_curves"`, `"fwr_curves"`, `"error"` appropriate plot methods are offered. For objects of the list item `regout` standard `lm` methods apply. For objects of the class `"cubicsplines"` print, summary and plot methods are available. Another term structure estimation method is provided by the function `nelson_estim`.

References

J.Huston McCulloch (1971): Measuring the Term Structure of Interest Rates. *The Journal of Business,* ***44*** *19–31.*

J. Huston McCulloch (1975): The Tax-Adjusted Yield Curve. *The Journal of Finance,* ***30*** *811–830.*

See Also

print.cubicsplines, summary.cubicsplines, plot.cubicsplines, nelson_estim, plot.spot_curves, plot.s_curves, plot.df_curves, plot.fwr_curves, plot.error, summary.lm, print.lm, plot.lm.

Examples

```
# load data set
data(eurobonds)

# define countries, for which the estimation
# of the zero-coupon yield curves will be carried out
group <- c("GERMANY", "AUSTRIA", "ITALY")

# define data set
bonddata <- eurobonds

# set maturtiy range
matrange <- c(0, 19)

# perform estimation
x <- splines_estim(group, bonddata, matrange)

# print the obtained parameters of the estimation
print(x)

# goodness of fit measures
summary(x)

# plot the zero-coupon yield curve for each country
plot(x,errors="none")

# plot all zero-coupon yield curves together
plot(x,multiple=TRUE,errors="none")

# spread curve splot
plot(x,ctype="spread",errors="none")

# price error plot for all countries
plot(x,ctype="none")
```

---

spotrates *Function for the Calculation of the Spot Rates*

Description

The function calculates the spot rates for the chosen approach, a provided maturity vector and parameter set.

Usage

```
spotrates(method, beta, m)
```

Arguments

`method`	`"Nelson/Siegel"` or `"Svensson"`.
`beta`	parameter set $\boldsymbol{\beta}$.
`m`	maturity or a vector of maturities.

Details

The spot rates according to Nelson/Siegel are defined as:

$$s(m, \boldsymbol{\beta}) = \beta_0 + \beta_1 \frac{1 - \exp(-\frac{m}{\tau_1})}{\frac{m}{\tau_1}} + \beta_2 \left(\frac{1 - \exp(-\frac{m}{\tau_1})}{\frac{m}{\tau_1}} - \exp(-\frac{m}{\tau_1}) \right).$$

Svensson defines the spot rate function as follows:

$$s(m, \boldsymbol{\beta}) = \beta_0 + \beta_1 \frac{1 - \exp(-\frac{m}{\tau_1})}{\frac{m}{\tau_1}} + \beta_2 \left(\frac{1 - \exp(-\frac{m}{\tau_1})}{\frac{m}{\tau_1}} - \exp(-\frac{m}{\tau_1}) \right) + \beta_3 \left(\frac{1 - \exp(-\frac{m}{\tau_2})}{\frac{m}{\tau_2}} - \exp(-\frac{m}{\tau_2}) \right)$$

Value

Returns a vector with the calculated spot rates.

References

Charles R. Nelson and Andrew F. Siegel (1987): Parsimonious Modeling of Yield Curves. *The Journal of Business*, **60(4)**:473–489.

Lars E.O. Svensson (1994): Estimating and Interpreting Forward Interest Rates: Sweden 1992 -1994. *Technical Reports 4871, National Bureau of Economic Research*.

`summary.cubicsplines`

S3 Summary Method for Cubicsplines

Description

S3 summary method for objects of the class `"cubicsplines"`.

Usage

```
## S3 method for class 'cubicsplines':
summary(object,...)
```

Arguments

object	object of the class "cubicsplines".
...	other arguments.

Details

The summary method for an object of the class "cubicsplines" prints goodness of fit statistics (RMSE, AABSE) of the optimisation. Additionally summary statistics of the regression analysis of the parameters are printed.

See Also

plot.cubicsplines, print.cubicsplines, rmse, aabse, summary.lm

---

summary.nelson *S3 Summary Method*

Description

S3 summary method for objects of the class "nelson".

Usage

```
## S3 method for class 'nelson':
summary(object,...)
```

Arguments

object	object of the class "nelson".
...	other arguments.

Details

The summary method for an object of the class "nelson" prints the solution of the goodness of fit statistics (RMSE,AABSE) of the optimisation. Moreover a convergence information of the used optimiser (nlminb) is printed.

See Also

nlminb, plot.nelson, print.nelson, rmse, aabse

`svensson` *Spot Rate Function according to Svensson*

Description

This function calculates the spot rates for certain maturity dates and a parameter vector according to Svensson (1994).

Usage

```
svensson(beta, m)
```

Arguments

`beta`	a vector of parameters $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \tau_1, \beta_3, \tau_2)$.
`m`	one maturity (or a vector of maturities).

Details

The spot rate according to Svensson for a maturity m is calculated using the following function:

$$s(m, \boldsymbol{\beta}) = \beta_0 + \beta_1 \frac{1 - \exp(-\frac{m}{\tau_1})}{\frac{m}{\tau_1}} + \beta_2 \left(\frac{1 - \exp(-\frac{m}{\tau_1})}{\frac{m}{\tau_1}} - \exp(-\frac{m}{\tau_1}) \right) + \beta_3 \left(\frac{1 - \exp(-\frac{m}{\tau_2})}{\frac{m}{\tau_2}} - \exp(-\frac{m}{\tau_2}) \right)$$

Value

Returns a vector consisting of the calculated spot rates.

References

Lars E.O. Svensson (1994): Estimating and Interpreting Forward Interest Rates: Sweden 1992 -1994. *Technical Reports 4871, National Bureau of Economic Research*.

Examples

```
svensson(c(0.07,0.3,0.05,0.1,0.08,0.2),1:30)
```

Index