

Package ‘israelcurves’

September 28, 2016

Type Package

Title Calculates historical curves for the Israeli sovereign bond market

Version 0.1.0

Author Or Moses

Author@R person("Or", "Moses", email="or.moses@gmail.com", role=c("aut", "cre"))

Maintainer Or Moses <or.moses@gmail.com>

Description Functions to calculate historical yield curves for the Israeli sovereign bond market. The calculation uses the Nelson-Siegel and Svensson models.

License GPL-3

LazyData TRUE

Imports ggplot2, Rblpapi, dplyr, Rsolnp, tidyr, lubridate

RoxygenNote 5.0.1

Suggests testthat

R topics documented:

bond	2
bond_by_name	2
calc_bond	3
calc_bond_name	3
calc_yields	4
create_all_bonds	5
create_bonds	5
create_bond_from_data	6
create_vanilla_bond	6
curve_model	7
get_bond_data	8
get_daily_data	9
israelcurves	9
make_params_for_all_dates	10
plot.bond	11
price_bond	11
price_bond_model	12
print.bond	12
summary.bond	13

Index**14**

bond	<i>Add a bond object (class bond)</i>
------	---------------------------------------

Description

bond is a helper function for building a bond class object.

Usage

```
bond(dates, payments, face_value = 100, name = NULL, issue_date = NULL,
     type = NULL, known_CPI = NULL)
```

Arguments

dates	A vector of payment dates
payments	A vector of payments (has to be same size as dates)
face_value	A number of the face value of the bond
name	(optional) The name of the bond
issue_date	(optional) The date the bond was issued
type	(optional) A string represents the type of the bond
known_CPI	(optional) A number represents the known CPI value at the time of issue of the bond

Value

An object of class "bond"

See Also

[create_vanilla_bond](#)

bond_by_name	<i>Extract bond by name</i>
--------------	-----------------------------

Description

A function that gets a list of bonds (bond class) and returns a bond object by its name.

Usage

```
bond_by_name(bond_list, name)
```

Arguments

bond_list	a list containing bond objects
name	A character contains the name of the bond.

Value

A bond class object

calc_bond	<i>Calculate main bond attributes</i>
-----------	---------------------------------------

Description

A function that calculates 3 main attributes for a bond:

- Yield to Maturity (ytm)
- Duration And modified duration
- Convexity

Usage

```
calc_bond(thebond, calc_date, market_price, ex_day = NULL, year_days = 365)
```

Arguments

thebond	A bond object.
calc_date	A date. The calculation date.
market_price	A number. The price of the bond to calculate by.
ex_day	(optional) A number indicating the Ex-day in the month where the bond pays coupon.
year_days	A number. The number of days in each year.

Value

A list with 4 items: yield to maturity, duration, modified duration and convexity

See Also

[calc_bond_name](#)

calc_bond_name	<i>Calculate bond attributes by name</i>
----------------	--

Description

A function that calculates the bond attributes from [calc_bond](#) by bonds' name. The function gets a list of bonds, name of bond, date and market price and calculates the attributes.

Usage

```
calc_bond_name(bonds_list, bond_name, calc_date, market_price, ex_day = NULL,  
year_days = 365)
```

Arguments

bonds_list	a list containing bond objects
bond_name	a string. The name of the bond.
calc_date	A date. The calculation date.
market_price	A number. The price of the bond to calculate by.
ex_day	(optional) A number indicating the Ex-day in the month where the bond pays coupon.
year_days	A number. The number of days in each year.

Value

A list with 4 items: yield to maturity, duration, modified duration and convexity

See Also

[calc_bond](#)

calc_yields	<i>Calculate a yield curve using a model</i>
-------------	--

Description

A function that calculates yields for a vector of terms using a model (Nelson Siegel or Svensson)

Usage

```
calc_yields(maturities, params, model = "NS")
```

Arguments

maturities	a numeric vector of positive maturities
params	a numeric vector. The model parameters. A 4-length for Nelson Siegel (NS) model and a 6-length for Svensson (NSS).
model	A string indicating the model to use - "NS" for Nelson Siegel and "NSS" for Svensson.

Value

A numeric vector of yields matching each number in the maturities vector.

create_all_bonds	Create a list of bond objects using Bloomberg SRCH
------------------	--

Description

A function that takes a Bloomberg save SRCH and creates a list of bond objects from this search.

Usage

```
create_all_bonds(srch_name)
```

Arguments

srch_name A string. A bloomberg saved SRCH name.

Value

a list of bond objects

See Also

[get_bond_data](#) for getting the data from Bloomberg, [create_bond_from_data](#) for creating one bond and [create_bonds](#) for creating a list of bonds.

create_bonds	Create a list of bond objects from the bloomberg data
--------------	---

Description

A function that takes a list of bonds' cashflows and creates a list of bond objects

Usage

```
create_bonds(bond_cf)
```

Arguments

bond_cf a list. A list of bond cashflows created by [get_bond_data](#).

Value

a list of bond objects

See Also

[get_bond_data](#) for getting the data from Bloomberg, [create_bond_from_data](#) for creating one bond.

create_bond_from_data	<i>Convert the bloomberg data into bond class A function that takes a list of bonds' cashflow and an item number and creates a bond object from the matching item in the list.</i>
-----------------------	--

Description

Convert the bloomberg data into bond class A function that takes a list of bonds' cashflow and an item number and creates a bond object from the matching item in the list.

Usage

```
create_bond_from_data(bond_cf, n)
```

Arguments

bond_cf	a list. A list of bond cashflows created by get_bond_data .
n	A number indicates the item from the list to create bond from.

Value

a bond object

See Also

[get_bond_data](#) for getting the data from Bloomberg.

create_vanilla_bond	<i>Create vanilla bond object</i>
---------------------	-----------------------------------

Description

create_vanilla_bond is a simpler function than [bond](#). It creates a bond object using generic details instead of exact dates and payments. The bond is a vanilla one - with coupons and one principal payment at maturity.

Usage

```
create_vanilla_bond(issue_date, first_payment, term, coupon, name = NULL,
  eom = TRUE, payment_frequency = 1, face_value = 100, year_days = 365,
  type = NULL, known_CPI = NULL)
```

Arguments

issue_date	A date. The issue date of the bond.
first_payment	A date. The date of the first coupon payment.
term	A number. The term of the bond in years.
coupon	A number. The coupon in percentage (for 5% use 5)
name	(optional) The name of the bond
eom	logical. A logical variable that indicates if the payments are at the end of each month.
payment_frequency	A number. The number of payments per year.
face_value	A number of the face value of the bond
year_days	A number. The number of days in each year.
type	(optional) A string represents the type of the bond
known_CPI	(optional) A number represents the known CPI value at the time of issue of the bond

Value

An object of class "bond"

See Also

[bond](#)

curve_model

Calculate a zero yield curve using a model for a certain day

Description

create a Nelson Siegel or Svensson interpolation zero curve using a bonds list, market prices and optionally trade volumes for a certain date. The optimization is done using Rsolnp package. The basic cost function is:

$$(P_{market} - P_{model})^2$$

Duration adjusted equation is:

$$\frac{(P_{market} - P_{model})^2}{Duration}$$

Volume Adjusted equation is:

$$(P_{market} - P_{model})^2 \cdot \frac{Volume}{TotalVolume}$$

With both adjustment the equation is:

$$\frac{(P_{market} - P_{model})^2}{Duration} \cdot \frac{Volume}{TotalVolume}$$

Usage

```
curve_model(bonds_list, market_data, calc_date, model = "NS",
  init_guess = NULL, adj_dur = TRUE, adj_vol = FALSE, max_vol = NULL)
```

Arguments

bonds_list	a list of bond objects
market_data	A dataframe. The known daily market data for the calculation date. The dataframe should have a "name" column that has names from the bonds list, a "market_price" column and optionally a "trade_volume" column.
calc_date	The calculation date.
model	A string indicating the model to use - "NS" for Nelson Siegel and "NSS" for Svensson.
init_guess	the initial guess for the optimization algorithm.
adj_dur	A logical indicates weather the calculation should be duration adjusted.
adj_vol	A logical indicates weather the calculation should be volume adjusted.
max_vol	A numeric indicates a maximal trade volume to be considered when calculating the volume adjustment for each bond.

Value

A vector of model parameters after optimization.

get_bond_data	<i>Get SRCH data from bloomberg</i>
---------------	-------------------------------------

Description

Code to get the data from the bloomberg SRCH using bsrch function (a custom SRCH needed to be saved) The code gets the main attributes of each bonnd found in the search as well as each bond's cashflow.

Usage

```
get_bond_data(srch_name)
```

Arguments

srch_name	A string. A bloomberg saved SRCH name.
-----------	--

Value

a list containing 2 items:

1. A dataframe contains the bonds main data
2. A list where each item is a bond's cashflow (normalized to 100)

get_daily_data	<i>Get daily data for a list of bonds</i>
----------------	---

Description

A function that gets a Bloomberg SRCH, list of bonds and a start date and returns a data frame with data for each date on each bond (market prices,trade volumes,time to maturity)

Usage

```
get_daily_data(srch_name, bond_list, start_date)
```

Arguments

srch_name	A string. A bloomberg saved SRCH name.
bond_list	A list of bond objects.
start_date	A date to start getting the data from.

Value

A dataframe with the following columns:

- date
- market price
- trade volume
- name of the bond
- maturity date of the bond
- time to maturity of the bond in the date

See Also

[create_all_bonds](#) for a function that creates a bond list that can be used in `get_daily_data` from a Bloomberg SRCH.

israelcurves	<i>israelcurves: calculating curves for the Israeli sovereign bonds.</i>
--------------	--

Description

The package uses Nelson Siegel and Svensson interpolation methods to calculate historical zero curves for the Israeli sovereign bond market.

israelcurves functions

bond calc_all

`make_params_for_all_dates`*Compute daily curves from data*

Description

Split the daily data by date and compute a curve for each day using a chosen model ("NS"/"NSS" for Nelson Siegel and Svensson)

Usage

```
make_params_for_all_dates(bond_list, daily_data, min_obs = 6, model = "NS",  
  adj_dur = TRUE, adj_vol = FALSE, max_vol = NULL)
```

Arguments

<code>bond_list</code>	a list of bond objects
<code>daily_data</code>	a dataframe contains daily data of the bonds constructs to be compatible with curve_model and can be extracted from Bloomberg using get_daily_data .
<code>min_obs</code>	an integer. The minimal number of bond's price observation in order to compute a curve for a certain date.
<code>model</code>	A string indicating the model to use - "NS" for Nelson Siegel and "NSS" for Svensson.
<code>adj_dur</code>	A logical indicates weather the calculation should be duration adjusted.
<code>adj_vol</code>	A logical indicates weather the calculation should be volume adjusted.
<code>max_vol</code>	A numeric indicates a maximal trade volume to be considered when calculating the volume adjustment for each bond.

Value

a list containing:

1. model name (string)
2. is duration adjusted (logical)
3. is volume adjusted (logical)
4. maximum volume for consideration (numeric)
5. dataframe of the results

plot.bond	<i>Plot method for bond class</i>
-----------	-----------------------------------

Description

Plot a cashflow of a bond.

Usage

```
## S3 method for class 'bond'
plot(x, y = NULL, ...)
```

Arguments

x, y	a bond object (class "bond").
...	Additional parameters

price_bond	<i>Price a bond</i>
------------	---------------------

Description

A function that gets a bond, discount_date and a vector of rates to discount and returns the bond's price as of the discount date

Usage

```
price_bond(thebond, disc_date, rates, ex_day = NULL, year_days = 365)
```

Arguments

thebond	A bond object
disc_date	The discount date
rates	A numeric vector of the discount rates corresponding to the payment dates
ex_day	(optional) A number indicating the Ex-day in the month where the bond pays coupon.
year_days	A number. The number of days in each year.

Value

The price of the bond for the discount date (numeric)

See Also

[price_bond_model](#) to price a bond using a model calculated discount rates.

price_bond_model	<i>Price a bond using rates from a model</i>
------------------	--

Description

A function that prices a bond using a discount rates calculated from a model: Nelson Siegel or Svensson.

Usage

```
price_bond_model(thebond, disc_date, model, model_params, ex_day = NULL,
  year_days = 365)
```

Arguments

thebond	A bond object
disc_date	The discount date
model	A string indicating the model to use - "NS" for Nelson Siegel and "NSS" for Svensson.
model_params	a numeric vector indicates the model parameters. A 4-length for Nelson Siegel (NS) model and a 6-length for Svensson (NSS).
ex_day	(optional) A number indicating the Ex-day in the month where the bond pays coupon.
year_days	A number. The number of days in each year.

See Also

[price_bond](#) to price a bond using a manual vector of discount rates.

print.bond	<i>Print method for bond class</i>
------------	------------------------------------

Description

Print method for bond class

Usage

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

Arguments

x	a bond object (class "bond").
...	Additional parameters

summary.bond	<i>Summary method for bond class</i>
--------------	--------------------------------------

Description

Summary method for bond class

Usage

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

Arguments

object	a bond object (class "bond").
...	Additional parameters

Index

bond, [2](#), [6](#), [7](#)
bond_by_name, [2](#)

calc_bond, [3](#), [3](#), [4](#)
calc_bond_name, [3](#), [3](#)
calc_yields, [4](#)
create_all_bonds, [5](#), [9](#)
create_bond_from_data, [5](#), [6](#)
create_bonds, [5](#), [5](#)
create_vanilla_bond, [2](#), [6](#)
curve_model, [7](#), [10](#)

get_bond_data, [5](#), [6](#), [8](#)
get_daily_data, [9](#), [10](#)

israelcurves, [9](#)
israelcurves-package (israelcurves), [9](#)

make_params_for_all_dates, [10](#)

plot.bond, [11](#)
price_bond, [11](#), [12](#)
price_bond_model, [11](#), [12](#)
print.bond, [12](#)

summary.bond, [13](#)