Package 'israelcurves'

October 6, 2016

Type Package

Title Calculates historical curves for the Israeli sovereign bond market

Version 0.1.0

Author Or Moses

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 *Add a bond object (class bond)*

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 *Extract bond by name*

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

build_curves *Wrapper function to calculate curves from data*

Description

A wrapper function that gets the data from Bloomberg and computes the curves

Usage

```
build_curves(srch_name, start_date, end_date = NULL, min_obs = 6,
  model = "NS", adj_dur = TRUE, adj_vol = FALSE, max_vol = NULL,
  ex_day = NULL)
```

Arguments

srch_name	A string. A bloomberg saved SRCH name.
start_date	A date to start getting the data from.
end_date	A date. Last date to get data from.
min_obs	an integer. The minimal number of bond's price observation in order to compute a curve for a certain date.
model	A string indicating the model to use - 'NS' for Nelson Siegel and 'NSS' for Svensson.
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.
ex_day	An integer indicating the Ex day of the month when a coupon payment occurs.

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

`calc_bond` *Calculate main bond attributes*

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 *Calculate bond attributes by name*

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 *Calculate a yield curve using a model*

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.

`check_boi` *Check results against Bank Of Israel zero curves data*

Description

A function to check the differences between the curves calculated by the package and the curves calculated by the Bank Of Israel

Usage

```
check_boi(result, boi_file)
```

Arguments

`result`	A list with the results of the package calculation
`boi_file`	a csv file with the zero curves data from the bank of israel

Value

A list with the model data, BOI data and differences

`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 *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.*

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` *Create vanilla bond object*

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 optionaly 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,
  ex_day = 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.
ex_day	An integer indicating the Ex day of the month when a coupon payment occurs.

Value

A vector of model parameters after optimization.

`get_bond_data` *Get SRCH data from bloomberg*

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_cpi_from_bloomberg` *Get CPI Index from bloomberg*

Description

A function to get the CPI Index from bloomberg using the last base

Usage

```
get_cpi_from_bloomberg()
```

get_daily_data *Get daily data for a list of bonds*

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, end_date = NULL)
```

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.
end_date	A date. Last date to get 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.

get_release_dates_from_bloomberg *Get CPI release dates from bloomberg*

Description

A function that gets the release and actual CPI dates from bloomberg (From 2001)

Usage

```
get_release_dates_from_bloomberg()
```

Value

a dataframe with release dates and actual dates

israelcurves *israelcurves: calculating curves for the Israeli sovereign bonds.*

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, ex_day = NULL)
```

Arguments

`bond_list`	a list of bond objects
`daily_data`	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`.
`min_obs`	an integer. The minimal number of bond's price observation in order to compute a curve for a certain date.
`model`	A string indicating the model to use - 'NS' for Nelson Siegel and 'NSS' for Svensson.
`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.
`ex_day`	An integer indicating the Ex day of the month when a coupon payment occurs.

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 *Plot method for bond class*

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

plot.zerocurve *Plot method for zerocurve class*

Description

Plot a curve of zerocurve class.

Usage

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

Arguments

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

plot_spreads *Spread plot of a time series of curves*

Description

The function plots a spread plot given a time series of curves (result of the package calculation) and two terms (min and max).

Usage

```
plot_spreads(result, min_term, max_term)
```

Arguments

result	list of results from the package calculation
min_term	a number. The lower term for the spread calculation
max_term	a number. The higher term for the spread calculation

Value

A plot

---

positive_CF *Bond Cashflow as of a certain date*

Description

Helper function to get the bond's cashflow as of a certain date.

Usage

```
positive_CF(thebond, thedate, ex_day = NULL, year_days = 365)
```

Arguments

thebond	a bond object
thedate	calculation date
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 the positive terms and payments

---

price_bond *Price a bond*

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 *Price a bond using rates from a model*

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 *Print method for bond class*

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 *Summary method for bond class*

Description

Summary method for bond class

Usage

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

Arguments

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

zerocurve *Create an object of a zerocurve class*

Description

A function that creates an object from zerocurve class - A list consisting of:

1. model - The name of the model ("NS" or "NSS")
2. params - The model parameters

Usage

```
zerocurve(model, params)
```

Arguments

`model`	a string representing the model name.
`params`	a numeric vector with the model fitted parameters.

Value

An object of type "zerocurve"

Index