Package ‘ycevo’

October 14, 2022

Type Package

Title Nonparametric Estimation of the Yield Curve Evolution

Version 0.1.2

Maintainer Yangzhuoran Fin Yang <yangyangzhuoran@gmail.com>

Description Nonparametric estimation of the discount rate and yield curve.
describe the application with the Center for Research in Security Prices (CRSP) Bond Data
and document the methods of this package.

URL https://github.com/bonsook/ycevo

BugReports https://github.com/bonsook/ycevo/issues

License GPL-3

Depends R (>= 3.5.0)

Encoding UTF-8

LazyData true

Imports dplyr, magrittr, Matrix, Rcpp (>= 0.12.18), rlang, stats

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.1.2

Suggests testthat (>= 3.0.0), knitr, plotly, rmarkdown, tidyverse

Language en-AU

Config/testthat/edition 3

NeedsCompilation yes

Author Bonsoo Koo [aut],
Nathaniel Tomasetti [aut],
Kai-Yang Goh [aut],
Yangzhuoran Fin Yang [aut, cre]
(<https://orcid.org/0000-0002-1232-8017>)

Repository CRAN

Date/Publication 2022-06-28 08:20:02 UTC
**Nonparametric Estimation of the Yield Curve Evolution**

**Description**

Nonparametric estimation of the discount rate and yield curve.

**Author(s)**

Bonsoo Koo <bonsoo.koo@monash.edu>
Kai-Yang Goh <kai-yang.goh@monash.edu>
Nathaniel Tomasetti <nathaniel.tomasetti@gmail.com>
Yangzhuoran Fin Yang (Maintainer) <fin.yang@monash.edu>

**References**


**See Also**

Useful links:

- [https://github.com/bonsook/ycevo](https://github.com/bonsook/ycevo)
- Report bugs at [https://github.com/bonsook/ycevo/issues](https://github.com/bonsook/ycevo/issues)
**generate_yield**  

*Generate a yield curve with cubic time evolution*

**Description**

Generate a yield curve using the extended version of Nelson & Siegel model (Nelson, C. R., & Siegel, A. F., 1987). This has been used in the simulation setting (Equation (30)) of Koo, B., La Vecchia, D., & Linton, O. (2021). See Details and References.

**Usage**

```r
generate_yield(
  n_qdate = 12,
  periods = 36,
  b0 = 0,
  b1 = 0.05,
  b2 = 2,
  t1 = 0.75,
  t2 = 125,
  linear = -0.55,
  quadratic = 0.55,
  cubic = -0.55
)
```

```r
going_yield_at(
  time,
  maturity,
  b0 = 0,
  b1 = 0.05,
  b2 = 2,
  t1 = 0.75,
  t2 = 125,
  linear = -0.55,
  quadratic = 0.55,
  cubic = -0.55
)
```

```r
going_yield_at_vec(
  time,
  maturity,
  b0 = 0,
  b1 = 0.05,
  b2 = 2,
  t1 = 0.75,
  t2 = 125,
  linear = -0.55,
  quadratic = 0.55,
  cubic = -0.55
)
```
cubic = -0.55 )

Arguments

n_qdate  Integer giving the number of quotation dates to use in the data. Defaults to 12.
periods Integer giving the maximum number of time-to-maturity periods the yield curve
is estimated for each quotation date. Defaults to 36
b0  Level term in yield curve equation, Defaults to 0. See Details.
b1  Slope term in yield curve equation, Defaults to 0.05. See Details.
b2  Curvature term in yield curve equation, Defaults to 2. See Details.
t1  Scaling parameter in yield curve equation, Defaults to 0.75. See Details.
t2  Scaling parameter in yield curve equation, Defaults to 125. See Details.
linear  Linear term in yield curve evolution, Defaults to -0.55. See Details.
quadratic  Quadratic term in yield curve evolution. Defaults to 0.55. See Details.
cubic  Cubic term in yield curve evolution. Defaults to -0.55. See Details.
time  Numeric value.
maturity Numeric value. Maturity in years.

Details

Returns a matrix where each column corresponds to a yield curve at a different point in time. The
initial curve at time to maturity zero is estimated from the following equation

\[ Yield_{i,0} = b_0 + b_1 \times (1 - \exp(\tau_i/t_1))/\tau_i + b_2 \times ((1 - \exp(-\tau_i/t_2))/\tau_i/t_2) - \exp(-\tau_i/t_2) \]

where \( \tau_i \) is the time to maturity, usually measured in years. This defines the yield curve for the
quotation date \( = 0 \). The yield curve for quotation dates \( = 1, 2, \ldots, \max_q_date \) multiplies this curve
by the cubic equation,

\[ Yield_{i,t} = Yield_{i,0} \times (1 + \text{linear} \times t + \text{quadratic} \times t^2 + \text{cubic} \times t^3) \]

so the yield curve slowly changes over different quotation dates.

Value

generate_yield Numeric matrix. Each column is a yield curve in a point in time (a quotation
date). Each row is for a time-to-maturity. For example, the number in the second column third
row is the yield for the yield curve at the second quotation date, for the third time-to-maturity
ranking from shortest to longest. See Details for the equation to generate the yield curve.

get_yield_at Numeric scalar.

generate_yield Numeric vector.
USbonds

Functions

- get_yield_at: Return the yield at a specific point in time of a specific maturity.
- get_yield_at_vec: Vectorised version of get_yield_at.

References


Examples

```r
out <- generate_yield()

# plots
library(tidyverse)
out <- data.frame(out)
colnames(out) <- 1:12
out <- mutate(out, time = 1:36)
out <- pivot_longer(out, -time, names_to = "qdate", values_to = "yield")
ggplot(out) +
  geom_line(aes(x=time, y=yield, color = qdate))
```

USbonds

CRSP US Bond Dataset from 02/01/2007 to 31/12/2007

Description

A dataset containing the prices and other attributes of CRSP US treasury bills, notes, and bonds. Columns qdate, crspid, tumat, mid.price, accint, pdint and tupq are required for estimation.

Usage

USbonds

Format

A data frame

- **qdate** Quotation date
- **crspid** Bond identifier
- **type** 1: Treasury Bonds, 2: Treasury Notes, 4: Treasury Bills
- **couppt** Coupon rate
- **matdate** Bond maturity date
tumat  Number of days to maturity from quotation date  
mid.price  Mid-Price, average between quoted bid and ask prices  
accint  The accumulated interest on payments  
issuedate  Bond issue date  
pqdate  Bond payment date. One entry for each payment.  
pdint  Bond payment amount  
tupq  Time until a given payment, given in days

Source
https://wrds-www.wharton.upenn.edu/

**Description**

[Experimental]
Nonparametric estimation of discount functions at given dates, time-to-maturities, and interest rates (experienced users only) and their transformation to the yield curves.

**Usage**

ycevo(data, xgrid, tau, ..., loess = length(tau) > 10)

estimate_yield(
  data,  
xgrid,  
hx,  
tau,  
ht,  
rgrid = NULL,  
hr = NULL,  
interest = NULL,  
loess = TRUE,  
price_slist = NULL,  
cf_slist = NULL  
)

**Arguments**

data  Data frame; bond data to estimate discount curve from. See ?USbonds for an example bond data structure.

xgrid  Numeric vector of values between 0 and 1. Time grids over the entire time horizon (percentile) of the data at which the discount curve is evaluated.
**tau**  Numeric vector that represents time-to-maturities in years where discount function and yield curve will be found for each time point xgrid. See Details.

...  Reserved for exogenous variables.

**loess**  Logical. Whether the output estimated discount and yield are to be smoothed using locally estimated scatterplot smoothing (LOESS)

**hx**  Numeric vector of values between 0 and 1. Bandwidth parameter determining the size of the window that corresponds to each time point (xgrid). See Details. The selection of bandwidth parameter is crucial in non-parametric estimation. If not sure, please use ycevo to allow the function choose it for you.

**ht**  Numeric vector that represents bandwidth parameter determining the size of the window in the kernel function that corresponds to each time-to-maturities (tau). The same unit as tau. See Details. The selection of bandwidth parameter is crucial in non-parametric estimation. If not sure, please use ycevo to allow the function choose it for you.

**rgrid**  (Optional) Numeric vector of interest rate grids in percentage at which the discount curve is evaluated, e.g. 4.03 means at interest rate of 4.03%.

**hr**  (Optional) Numeric vector of bandwidth parameter in percentage determining the size of the window in the kernel function that corresponds to each interest rate grid (rgrid).

**interest**  (Optional) Numeric vector of daily short term interest rates. The length is the same as the number of quotation dates included in the data, i.e. one interest rate per day.

**price_slist**  (Internal) Experienced users only. A list of matrices, generated by the internal function calc_price_slist.

**cf_slist**  (Internal) Experienced users only. A list of matrices, generated by the internal function calc_cf_slist.

**Details**

Suppose that a bond \( i \) has a price \( p_i \) at time \( t \) with a set of cash payments, say \( c_1, c_2, \ldots, c_m \) with a set of corresponding discount values \( d_1, d_2, \ldots, d_m \). In the bond pricing literature, the market price of a bond should reflect the discounted value of cash payments. Thus, we want to minimise

\[
(p_i - \sum_{j=1}^{m} c_j \times d_j)^2.
\]

For the estimation of \( d_k (k = 1, \ldots, m) \), solving the first order condition yields

\[
(p_i - \sum_{j=1}^{m} c_j \times d_j)c_k = 0,
\]

and

\[
\hat{d}_k = \frac{p_ic_k}{c_k^2} - \frac{\sum_{j=1, k \neq k}^{m} c_k c_j d_j}{c_k^2}.
\]

There are challenges: \( \hat{d}_k \) depends on all the relevant discount values for the cash payments of the bond. Our model contains random errors and our interest lies in expected value of \( d(\cdot) \) where
the expected value of errors is zero. $d(.)$ is an infinite-dimensional function not a discrete finite-dimensional vector. Generally, cash payments are made biannually, not dense at all. Moreover, cash payment schedules vary over different bonds.

Let $d(\tau, X_t)$ be the discount function at given covariates $X_t$ (dates $xgrid$ and interest rates $rgrid$), and given time-to-maturities $\tau$ ($tau$). $y(\tau, X_t)$ is the yield curve at given covariates $X_t$ (dates $xgrid$ and interest rates $rgrid$), and given time-to-maturities $\tau$ ($tau$).

We pursue the minimum of the following smoothed sample least squares objective function for any smooth function $d(.)$:

$$Q(d) = \sum_{t=1}^{T} \sum_{i=1}^{n} \left( p_{it} - \sum_{j=1}^{m_i} c_{it}(\tau_{ij}) d(s_{ij}, x) \right)^2 \sum_{k=1}^{m_i} \left( K_h(s_{ik} - \tau_{ik}) d s_{ik} \right) K_h(x - X_t) dx,$$

where a bond $i$ has a price $p_i$ at time $t$ with a set of cash payments $c_1, c_2, \ldots, c_m$ with a set of corresponding discount values $d_1, d_2, \ldots, d_m$, $K_h(.) = K(./h)$ is the kernel function with a bandwidth parameter $h$, the first kernel function is the kernel in space with bonds whose maturities $s_{ik}$ are close to the sequence $\tau_{ik}$, the second kernel function is the kernel in time and in interest rates with $x$, which are close to the sequence $X_t$. This means that bonds with similar cash flows, and traded in contiguous days, where the short term interest rates in the market are similar, are combined for the estimation of the discount function at a point in space, in time, and in "interest rates".

The estimator for the discount function over time to maturity and time is

$$\hat{d} = \arg \min_{d} Q(d).$$

This function provides a data frame of the estimated yield and discount rate at each combination of the provided grids. The estimated yield is transformed from the estimated discount rate.

For more information on the estimation method, please refer to References.

**Value**

Data frame of the yield and discount rate at each combination of the provided grids.

- **discount** Estimated discount rate
- **xgrid** Same as input $xgrid$
- **tau** Same as input $tau$
- **yield** Estimated yield

**Functions**

- `estimate_yield`: Experienced users only. Yield estimation with interest rate and manually selected bandwidth parameters.

**Author(s)**

Nathaniel Tomasetti, Bonsoo Koo, and Yangzhuoran Fin Yang

**References**

Examples

```r
library(dplyr)
# Simulate 4 bonds issued at 2020-01-01
# with maturity 180, 360, 540, 720 days
# Apart from the first one,
# each has coupon 2,
# of which half is paid every 180 days.
# The yield curve is simulated from `get_yield_at_vec` 
# Quotation date is also at 2020-01-01
exp_data <- tibble(
  qdate = "2020-01-01",
  crspid = rep(1:4, 1:4),
  pdint = c(100, 1, 101, 1, 101, 1, 1, 1, 101),
  tupq = unlist(sapply(1:4, seq_len)) * 180,
  accint = 0
) %>%
mutate(discount = exp(-tupq/365 * get_yield_at_vec(0, tupq/365))) %>%
group_by(crspid) %>%
mutate(mid.price = sum(pdint * discount)) %>%
ungroup()

# Only one quotation date so time grid is set to 1
xgrid <- 1
# Discount function is evaluated at time to maturity of each payment in the data
tau <- unique(exp_data$tupq/365)
ycevo(
  exp_data, 
  xgrid = xgrid, 
  tau = tau
)
```
Index

* datasets
  USbonds, 5

* package
  ycevo-package, 2
  _PACKAGE (ycevo-package), 2

  estimate_yield (ycevo), 6

  generate_yield, 3
  get_yield_at (generate_yield), 3
  get_yield_at_vec (generate_yield), 3

USbonds, 5

ycevo, 6
ycevo-package, 2