# Package: ycevo (via r-universe)

October 27, 2026

**Type** Package

**Title** Nonparametric Estimation of the Yield Curve Evolution

**Version** 0.3.0

**Maintainer** Yangzhuoran Fin Yang <yangyangzhuoran@gmail.com>

**Description** Nonparametric estimation of discount functions and yield curves from transaction data of coupon paying bonds. Koo, B., La Vecchia, D., & Linton, O. B. (2021) <doi:10.1016/j.jeconom.2020.04.014> describe an application of this package using the Center for Research in Security Prices (CRSP) Bond Data and document its implementation.

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

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

**License** GPL-3

**Depends** R (>= 3.5.0)

**Encoding** UTF-8

**Imports** dplyr (>= 1.0.0), future.apply, generics, ggplot2, graphics, lubridate, Matrix, progressr, Rcpp (>= 0.12.18), rlang, scales, stats, tibble, tidyr, tidyselect

**LinkingTo** Rcpp, RcppArmadillo

**RoxygenNote** 7.3.2

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

**Language** en-AU

**Config/testthat/edition** 3

**NeedsCompilation** yes

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

**Repository** https://cran.r-universe.dev

**Date/Publication** 2025-08-19 13:50:02 UTC

**RemoteUrl** https://github.com/cran/ycevo

**RemoteRef** HEAD

**RemoteSha** 28ed71cabcc74301072447401f7e3924415b654d

# Contents

ycevo-package . . . 2
augment.ycevo . . . 3
autoplot.ycevo . . . 4
generate_yield . . . 5
plot.ycevo . . . 8
predict.ycevo . . . 9
vis_kernel . . . 10
ycevo . . . 11
ycevo_data . . . 15

**Index** **18**

---

`ycevo-package` *Nonparametric Estimation of the Yield Curve Evolution*

---

## Description

Nonparametric estimation of discount functions and yield curves.

## Author(s)

**Maintainer**: Yangzhuoran Fin Yang <yangyangzhuoran@gmail.com> (ORCID)

Authors:

- Bonsoo Koo <Bonsoo.Koo@monash.edu>

Other contributors:

- Nathaniel Tomasetti [contributor]
- Kai-Yang Goh [contributor]

## References

Koo, B., La Vecchia, D., & Linton, O. (2021). Estimation of a nonparametric model for bond prices from cross-section and time series information. Journal of Econometrics, 220(2), 562-588.

## See Also

Useful links:

- https://github.com/bonsook/ycevo
- Report bugs at https://github.com/bonsook/ycevo/issues

`augment.ycevo` *Augment data with predicted discount functions and yield curves*

## Description

Augment data with predicted discount functions and yield curves

## Usage

```
## S3 method for class 'ycevo'
augment(x, newdata = NULL, loess = TRUE, ...)
```

## Arguments

| | |
|---|---|
| `x` | A ycevo object |
| `newdata` | A data frame containing time-to-maturity in years `tau` and the quotation date at which the discount functions and the yield curves are to be predicted. The quotation date is to be named after the quotation date column in `x`, which is also the name of the quotation date column of the `data` argument in `ycevo()` that produces `x`. The default is `qdate`. |
| `loess` | Logical. If TRUE, the returned discount functions and yield curves are loess smoothed. |
| `...` | Additional arguments required for generic consistency. Currently not used. Warning: A misspelled argument will not raise an error. The misspelled argument will be either disregarded, or the default value will be applied if one exists. |

## Details

If `newdata` is not provided, returns the discount function and yield curve at the specified estimation points in `ycevo()`.

If `newdata` is provided, the discount functions at the time-to-maturities specified in `newdata` are generated from loess smoothing (see `stats::loess()`), and interpolated to produce the discount function values at the quotation date specified in `newdata`, before being converted to the yield curves.

## Value

`newdata` augmented with `.discount` and `.yield` for the discount function and the yield curve respectively.

## See Also

`ycevo()`, `predict.ycevo()`

## Examples

```
# Simulating bond data
bonds <- ycevo_data(n = 10)

# Estimation can take up to 30 seconds
res <- ycevo(bonds, x = lubridate::ymd("2023-03-01"))
# Augmentation
augment(res)
```

---

`autoplot.ycevo` *Plot the estimated discount functions and yield curves with ggplot2 or plotly*

---

## Description

Plots the estimated discount functions and yield curves from a `ycevo()` object with `ggplot2` or `plotly`.

## Usage

```
## S3 method for class 'ycevo'
autoplot(
  object,
  est = c("both", "discount", "yield"),
  against = c("tau", "x", "both"),
  loess = TRUE,
  ...
)
```

## Arguments

| | |
|---|---|
| `object` | A ycevo object |
| `est` | String. Indicates which estimated values to plot: discount function, yield curve, or both. Default is both. |
| `against` | String. Indicates which variable to plot against, i.e. what is on the x axis. Time-to-maturity `tau`, quotation date `x`, or both (requires package `plotly`). If both, an interactive 3D plot is generated. |
| `loess` | Logical. If TRUE, the returned discount functions and yield curves are loess smoothed. |
| `...` | Additional arguments required for generic consistency. Currently not used. Warning: A misspelled argument will not raise an error. The misspelled argument will be either disregarded, or the default value will be applied if one exists. |

## Value

A ggplot2::ggplot() object if only one dimension is specified in against. A plotly::plot_ly() object if against is set to both.

## See Also

ycevo()

## Examples

```
# Simulating bond data
bonds <- ycevo_data(n = 10)

# Estimation can take up to 30 seconds
res <- ycevo(bonds, x = lubridate::ymd("2023-03-01"))
# Plot
autoplot(res)
```

---

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

```
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
)

get_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
)

get_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
)
```

## Arguments

| | |
|---|---|
| n_qdate | Integer. Number of quotation dates to use in the data. Defaults to 12. |
| periods | Integer. Maximum number of time-to-maturity periods in 10 years that the yield curve is estimated for each quotation date. Defaults to 36 |
| b0 | Level term in yield curve equation, Defaults is 0. See Details. |
| b1 | Slope term in yield curve equation, Defaults is 0.05. See Details. |
| b2 | Curvature term in yield curve equation, Defaults is 2. See Details. |
| t1 | Scaling parameter in yield curve equation, Defaults is 0.75. See Details. |
| t2 | Scaling parameter in yield curve equation, Defaults is 125. See Details. |
| linear | Linear term in yield curve evolution, Defaults is -0.55. See Details. |
| quadratic | Quadratic term in yield curve evolution. Defaults is 0.55. See Details. |
| cubic | Cubic term in yield curve evolution. Defaults is -0.55. See Details. |
| time | Numeric value. |
| maturity | Numeric value. Maturity in years. |

## Details

The initial curve at time zero is generated from the following equation

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

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 time is obtained by multiplying this curve

by the cubic equation,

$$Yield_{i,time} = Yield_{i,0} * (1 + linear * time + quadratic * time^2 + cubic * time^3)$$

so the yield curve slowly changes over different quotation dates.

## Value

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

`get_yield_at()` Numeric vector.

`get_yield_at_vec()` Numeric vector.

## Functions

- `get_yield_at()`: Return the yield at specific points in time of specific maturities.
- `get_yield_at_vec()`: Deprecated. Vectorised version of `get_yield_at()`. Use `get_yield_at()` instead.

## References

Nelson, C. R., & Siegel, A. F. (1987). Parsimonious Modeling of Yield Curves. The Journal of Business, 60(4), 473-489.

Koo, B., La Vecchia, D., & Linton, O. (2021). Estimation of a nonparametric model for bond prices from cross-section and time series information. Journal of Econometrics, 220(2), 562-588.

## See Also

`ycevo_data()`

## Examples

```
out <- generate_yield()

# plots
library(ggplot2)
out <- data.frame(out)
colnames(out) <- 1:12
out <- dplyr::mutate(out, time = 1:36)
out <- tidyr::pivot_longer(out, -time, names_to = "qdate", values_to = "yield")
ggplot(out) +
  geom_line(aes(x=time, y=yield, color = qdate))
```

plot.ycevo *Plot the estimated discount functions and yield curves*

## Description

Plots the estimated discount functions and yield curves from a `ycevo()` object.

## Usage

```
## S3 method for class 'ycevo'
plot(
  x,
  est = c("both", "discount", "yield"),
  against = c("tau", "x"),
  loess = TRUE,
  ...
)
```

## Arguments

| | |
|---|---|
| x | A ycevo object |
| est | String. Indicates which estimated values to plot: discount function, yield curve, or both. Default is both. |
| against | String. Indicates which variable to plot against, i.e. what is on the x axis. Time-to-maturity `tau` or quotation date `x`. |
| loess | Logical. If TRUE, the returned discount functions and yield curves are loess smoothed. |
| ... | Additional arguments required for generic consistency. Currently not used. Warning: A misspelled argument will not raise an error. The misspelled argument will be either disregarded, or the default value will be applied if one exists. |

## Value

`NULL`. The functions produces a plot.

## See Also

`ycevo()`, `autoplot.ycevo()`

## Examples

```
# Simulating bond data
bonds <- ycevo_data(n = 10)

# Estimation can take up to 30 seconds
res <- ycevo(bonds, x = lubridate::ymd("2023-03-01"))
```

```
# Plot
plot(res)
```

---

`predict.ycevo` *Return predicted discount functions or yield curves*

---

## Description

Return predicted discount functions or yield curves

## Usage

```
## S3 method for class 'ycevo'
predict(
  object,
  newdata = NULL,
  type = c("discount", "yield"),
  loess = TRUE,
  ...
)
```

## Arguments

| | |
|---|---|
| `object` | A ycevo object |
| `newdata` | A data frame containing time-to-maturity in years `tau` and the quotation date at which the discount functions and the yield curves are to be predicted. The quotation date is to be named after the quotation date column in `x`, which is also the name of the quotation date column of the `data` argument in `ycevo()` that produces `x`. The default is `qdate`. |
| `type` | Whether to return the discount function or the yield curve. |
| `loess` | Logical. If TRUE, the returned discount functions and yield curves are loess smoothed. |
| `...` | Additional arguments required for generic consistency. Currently not used. Warning: A misspelled argument will not raise an error. The misspelled argument will be either disregarded, or the default value will be applied if one exists. |

## Details

If `newdata` is not provided, returns the discount function or yield curve at the specified estimation points in `ycevo()`.

If `newdata` is provided, the discount functions at the time-to-maturities specified in `newdata` are generated from loess smoothing (see `stats::loess()`), and interpolated to produce the discount function values at the quotation date specified in `newdata`, before being converted to the yield curves.

## Value

A numeric vector of the discount function or the yield curve based on type.

## See Also

ycevo(), augment.ycevo()

## Examples

```
# Simulating bond data
bonds <- ycevo_data(n = 10)

# Estimation can take up to 30 seconds
res <- ycevo(bonds, x = lubridate::ymd("2023-03-01"))
# Prediction
predict(res)
```

---

vis_kernel *Visualise kernel weights*

---

## Description

vis_kernel() visualises kernel weights assigned to the intervals surrounding specific grid points using the Epanechnikov kernel function and given bandwidths.

## Usage

```
vis_kernel(data, x = NULL, hx = NULL, tau = NULL, ht = NULL, ...)
```

## Arguments

| | |
|---|---|
| data | Bond data. If x and hx are not NULL, data needs to include one time index column qdate. If tau and ht are not NULL, data needs to include one column of time to maturity in days tupq. |
| x | Time grids at which the discount curve is evaluated. Should be specified using the same class of object as the quotation date (qdate) column in data. |
| hx | Numeric vector. Bandwidth parameters corresponding to each time point x. |
| tau | Numeric vector. Time-to-maturities in years where discount function and yield curve will be estimated for each of time points x. See Details. |
| ht | Numeric vector. Bandwidth parameters corresponding to each value of time-to-maturities tau. See Details. |
| ... | Specification of an additional covariate, taking the form of var = list(grid, bandwidth), where var is the name of the covariate in data, grid is the values at which the yield curve is estimated, similar to x, and bandwidth is the bandwidth parameter corresponding to each of the grid values, similar to hx. |

### Details

If x and hx are provided, the kernel weights assigned to the intervals surrounding each of time points x will be plotted.

If tau and ht are provided, the kernel weights assigned to the intervals surrounding each of time-to-maturities tau will be plotted.

If the grid and bandwidth of a covariate are provided in ..., the kernel weights of that covariate will be plotted.

The kernel weights can only be plotted in one dimension (time, time-to-maturity, or covariate) at a time.

### Value

A ggplot2::ggplot() object.

### See Also

ycevo_data(), ycevo()

### Examples

```
bonds <- ycevo_data()
vis_kernel(bonds, x = lubridate::ymd("2023-06-01"), hx = 0.2)
```

---

## ycevo *Estimate yield function*

---

### Description

Nonparametric estimation of discount functions and yield curves at given dates, time-to-maturities, and one additional covariate, usually interest rate.

### Usage

```
ycevo(
  data,
  x,
  span_x = 60,
  hx = NULL,
  tau = NULL,
  ht = NULL,
  tau_p = tau,
  htp = NULL,
  cols = NULL,
  ...
)
```

```
estimate_yield(
  data,
  xgrid,
  hx,
  tau,
  ht,
  tau_p = tau,
  htp = ht,
  rgrid = NULL,
  hr = NULL,
  interest = NULL,
  cfp_slist = NULL
)
```

## Arguments

| | |
|---|---|
| data | Data frame; bond data to estimate discount curve from. See ycevo_data() for an example bond data structure. Minimum required columns are qdate, id, price, tupq, and pdint. The columns can be named differently: see cols. |
| x | Time grids at which the discount curve is evaluated. Should be specified using the same class of object as the quotation date (qdate) column in data. |
| span_x | Half of the window size, or the distance from the centre x to the maximum (or the minimum) qdate with non-zero weight using the kernel function, measured by the number of regular interval between two consecutive qdate. Ignored if hx is specified. See Details. |
| hx | Numeric vector. Bandwidth parameters corresponding to each time point x. |
| tau | Numeric vector. Time-to-maturities in years where discount function and yield curve will be estimated for each of time points x. See Details. |
| ht | Numeric vector. Bandwidth parameters corresponding to each value of time-to-maturities tau. See Details. |
| tau_p | Numeric vector. Auxiliary time-to-maturities in years. See Details. |
| htp | Numeric vector. Bandwidth parameters corresponding to each of auxiliary time-to-maturities tau_p. See Details. |
| cols | <tidy-select> A named list or vector of alternative names of required variables, following the new_name = old_name syntax of the dplyr::rename(), where the new_nam takes one of the five column names required in data. This enables the user to provide data with columns named differently from required. |
| ... | Specification of an additional covariate, taking the form of var = list(grid, bandwidth), where var is the name of the covariate in data, grid is the values at which the yield curve is estimated, similar to x, and bandwidth is the bandwidth parameter corresponding to each of the grid values, similar to hx. |
| xgrid | Numeric vector. Values between 0 and 1. Time grids over the entire time horizon (percentile) of the data at which the discount function is evaluated. |
| rgrid | (Optional) Numeric vector. Interest rate grids in percentage at which the discount function is evaluated, e.g. 4.03 means at interest rate of 4.03%. |

| | |
|---|---|
| `hr` | (Optional) Numeric vector. 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. 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. |
| `cfp_slist` | (Internal) Experienced users only. A list of matrices, generated by the internal function 'get_cfp_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

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

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 $d(.)$ 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 `x` 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 `x` 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 $d(.)$:

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

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.

An alternative specification of bandwidth `hx` is `span_x`, which provides kernel coverage invariant to the length of `data`. `span_x` takes an absolute measure of time depending on the unit of `x`. The default value is 60. If the data is daily on trading days, i.e., the interval between every two consecutive `qdate` is one trading day, then the window of the kernel function allows the estimation at each point `x` to contain information from 60 trading days prior to and after the time point `x`.

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

## Value

A `tibble::tibble()` object of class `ycevo` with the following columns.

**qdate** The time points that user-specified as `x`. The name of this column will be consistent with the name of the time index column in the `data` input, if the user choose to provide a data frame with the time index column named differently from `qdate` with the `cols` argument.

**.est** A nested columns of estimation results containing a `tibble::tibble()` for each `qdate`. Each `tibble` contains three columns: `tau` for the time-to-maturity specified by the user in the `tau` argument, `.disount` for the estimated discount function at this time and this time-to-maturity, and `.yield` for the estimated yield curve.

## Functions

- `estimate_yield()`: Experienced users only. Yield estimation with interest rate and manually selected bandwidth parameters. Only length one x and length one hx are supported at a time. Returns a 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

## References

Koo, B., La Vecchia, D., & Linton, O. (2021). Estimation of a nonparametric model for bond prices from cross-section and time series information. Journal of Econometrics, 220(2), 562-588.

## See Also

`augment.ycevo()`, `autoplot.ycevo()`

## Examples

```
# Simulating bond data
bonds <- ycevo_data(n = 10)

# Estimation can take up to 30 seconds
ycevo(bonds, x = lubridate::ymd("2023-03-01"))
```

---

`ycevo_data` *Simulate bond data*

---

## Description

Simulates bond transaction data at each weekday throughout the year 2023, following the extended version of Nelson & Siegel model (Nelson, C. R., & Siegel, A. F., 1987).

## Usage

```
ycevo_data(
  n = 40,
  b0 = 0,
  b1 = 0.05,
  b2 = 2,
  t1 = 0.75,
  t2 = 125,
  linear = -0.55,
  quadratic = 0.55,
  cubic = -0.55
)
```

## Arguments

| | |
|---|---|
| `n` | Integer. Number of bonds of each maturity to simulation |
| `b0` | Level term in yield curve equation, Defaults is 0. See `Details`. |
| `b1` | Slope term in yield curve equation, Defaults is 0.05. See `Details`. |
| `b2` | Curvature term in yield curve equation, Defaults is 2. See `Details`. |
| `t1` | Scaling parameter in yield curve equation, Defaults is 0.75. See `Details`. |
| `t2` | Scaling parameter in yield curve equation, Defaults is 125. See `Details`. |
| `linear` | Linear term in yield curve evolution, Defaults is -0.55. See `Details`. |
| `quadratic` | Quadratic term in yield curve evolution. Defaults is 0.55. See `Details`. |
| `cubic` | Cubic term in yield curve evolution. Defaults is -0.55. See `Details`. |

## Details

n bonds for each of the following maturities are simulated: 20, 10, 5, 3, 2 and 0.8 years. The face value of all bonds is 100. Bonds with 0.8 years of maturity are similar to the US Treasury bills with no coupon. Bonds with maturity between 2 and 10 years correspond to the US Treasury notes. Their coupon rates are simulated from an Epanechnikov distribution with mean 4, and the distance from the mean to the boundary is 2.65. Bonds with maturity of 20 years corresponds to the US Treasury bonds. Their coupon rates are simulated from an Epanechnikov distribution with mean 7.5, and the distance from the mean to the boundary is 2.65. Coupons are payable every 6 months.

We artificially "observe" quotation data of bonds on every weekday through out 2023, starting with 2 Jan 2023. To ensure an adequate number of transactions are observed for the estimation of the yield curve, the earliest bond is issued prior to the beginning of 2023, determined by the length of maturity of that type of bond, such that the last payment can still be observed at the beginning of 2023. For example, the first bond with 20 years of maturity is issued at the beginning of 2003. The last bond within this type is issued at the end of 2023. The rest of the bonds have issue dates evenly distributed between the first and the last bonds.

The initial yield at the beginning of 2023 is generated from the following equation

$$Yield_{i,0} = b_0 + b_1 * ((1 - \exp(-\tau_i/t_1))/(\tau/t_1)) + b_2 * ((1 - \exp(-\tau_i/t_2))/(\tau_i/t_2) - \exp(-\tau_i/t_2))$$

where $\tau_i$ is the time to maturity in years. The yield curve at quotation time $t$ is obtained by multiplying this curve by the cubic equation,

$$Yield_{i,t} = Yield_{i,0} * (1 + linear * time + quadratic * time^2 + cubic * time^3)$$

so the yield curve slowly changes over different quotation dates. The time $t$ is a value between 0 and 1, the proportion of time that has passed by a quotation date, identifying the progression through 2023. For example, the time $t$ corresponding to 31 Mar 2023 is 0.25.

The discount function is then derived from the yield curve,

$$d_t(\tau) = \exp(-\tau y_t(\tau))$$

to discount all the future cash flows of a bond and calculate the price of that bond at the quotation date.

## Value

A `tibble::tibble()` object with 5 variables

**qdate** The quotation date of a bond in a `Date()` class.

**id** The unique identifier of a bond.

**price** The price of a bond.

**tupq** The remaining time until the given cash flow in days.

**pdint** The payment amount of the cash flow. For a non-coupon-paying bond, the only cash flow occurs on the maturity date with a payment of 100, i.e., the face value of the bond. For a coupon-paying bond, the stream of cash flows includes the coupon payable on the interest payment date before maturity and the face value 100 plus the coupon payment for the last cash flow on the maturity date.

## References

Nelson, C. R., & Siegel, A. F. (1987). Parsimonious Modeling of Yield Curves. The Journal of Business, 60(4), 473-489.

Koo, B., La Vecchia, D., & Linton, O. (2021). Estimation of a nonparametric model for bond prices from cross-section and time series information. Journal of Econometrics, 220(2), 562-588.

## See Also

`get_yield_at()`

## Examples

```
ycevo_data()
```

# Index

∗ **package**
ycevo-package, 2

augment.ycevo, 3
augment.ycevo(), *10*, *14*
autoplot.ycevo, 4
autoplot.ycevo(), *8*, *14*

Date(), *16*
dplyr::rename(), *12*

estimate_yield *(*ycevo*)*, 11

generate_yield, 5
get_yield_at *(*generate_yield*)*, 5
get_yield_at(), *17*
get_yield_at_vec *(*generate_yield*)*, 5
ggplot2::ggplot(), *5*, *11*

plot.ycevo, 8
plotly::plot_ly(), *5*
predict.ycevo, 9
predict.ycevo(), *3*

stats::loess(), *3*, *9*

tibble::tibble(), *14*, *16*

vis_kernel, 10

ycevo, *3*, *4*, *8*, *9*, 11
ycevo(), *3–5*, *8–11*
ycevo-package, 2
ycevo_data, 15
ycevo_data(), *7*, *11*, *12*