

yieldcurves

Yield curve fitting and analysis in R

Charles Coverdale

London, United Kingdom

April 2026

The question

How should we fit, decompose, and analyse the term structure of interest rates in a single reproducible R interface?

Nineteen functions, one curve object, no heavy dependencies.

Why it matters

- **Central banks** publish fitted Nelson-Siegel and Svensson curves daily
- **Fixed-income portfolio managers** need duration, convexity, and key rate exposures
- **Relative-value traders** price carry and roll-down across the curve
- **Academic researchers** decompose curve movements via PCA and factor models

What is already out there

- **YieldCurve**: Nelson-Siegel and Svensson fits only; requires xts/zoo; no duration, convexity, PCA, or carry¹
- **termstrc**: broader coverage but archived from CRAN in 2018; no current maintenance
- **fBonds**: bond-pricing primitives only; no curve estimation
- **Bespoke scripts**: central-bank desks reimplement the same pipeline in-house

The gap: **no single package delivers fitting, risk measures, and decomposition in one interface.**

¹ Guirrerri (2022), *YieldCurve: Modelling and Estimation of the Yield Curve*, R package v5.2.

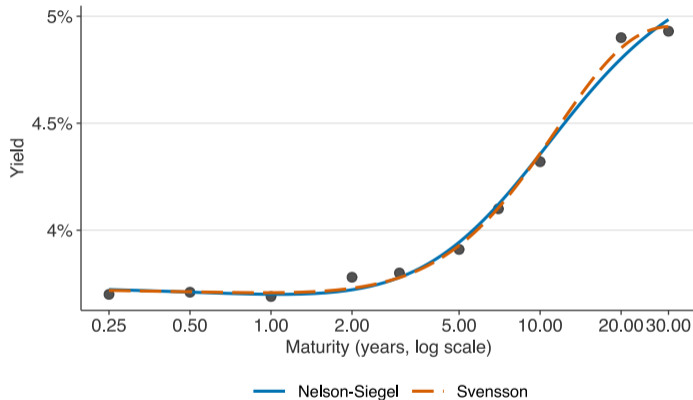
What yieldcurves offers

1. **Coverage:** 19 functions spanning fitting (Nelson-Siegel, Svensson, cubic spline), extraction (forwards, discount factors, par-zero), risk (duration, convexity, Z-spread, KRD), and decomposition (PCA, level-slope-curvature, carry)
2. **Interface:** one `yc_curve` S3 object flows through every downstream function; inputs are plain numeric vectors
3. **Provenance:** pure R, no compiled code, no heavy dependencies, 172 unit tests

On CRAN since March 2026. Parametric fits follow Nelson and Siegel (1987) and Svensson (1994)².

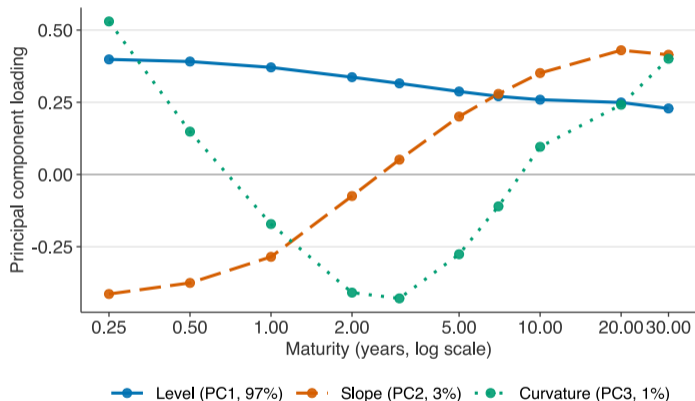
² Nelson & Siegel (1987), *Parsimonious modeling of yield curves*, J. Business 60(4); Svensson (1994), *Estimating and interpreting forward interest rates*, IMF WP 94/114.

Fitting: Nelson-Siegel and Svensson



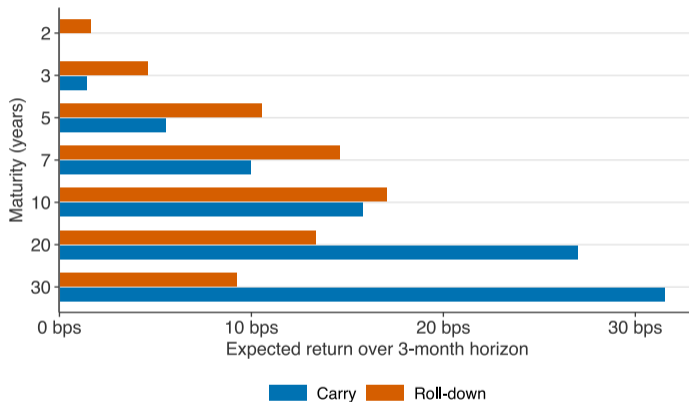
Parametric fits to a US Treasury curve. Exports: `yc_nelson_siegel()`, `yc_svensson()`, `yc_cubic_spline()`, `yc_fit()`.

Decomposition: PCA level, slope, curvature



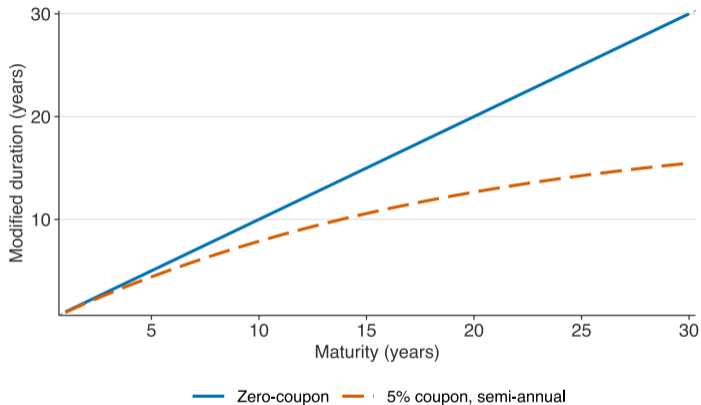
PCA of monthly US Treasury curve changes. Exports: `yc_pca()`, `yc_level_slope_curvature()`, `yc_slope()`.

Carry and roll-down



Carry and roll-down at a three-month horizon. Exports: `yc_carry()`, `yc_forward()`, `yc_discount()`, `yc_par_to_zero()`.

Risk: duration and convexity



Modified duration across maturities, zero-coupon versus par-coupon. Exports: `yc_duration()`, `yc_bond_duration()`, `yc_key_rate_duration()`.

Central formulas

Nelson-Siegel (three factors plus decay τ):

$$y(m) = \beta_0 + \beta_1 \left(\frac{1 - e^{-m/\tau}}{m/\tau} \right) + \beta_2 \left(\frac{1 - e^{-m/\tau}}{m/\tau} - e^{-m/\tau} \right) \quad (1)$$

Svensson extension (adds second curvature β_3, τ_2):

$$y(m) = y_{\text{NS}}(m) + \beta_3 \left(\frac{1 - e^{-m/\tau_2}}{m/\tau_2} - e^{-m/\tau_2} \right) \quad (2)$$

Modified duration (price sensitivity to parallel shift):

$$D_{\text{mod}} = \frac{1}{P} \sum_t \frac{t \cdot CF_t}{(1 + y/k)^{kt+1}} \quad (3)$$

Package at a glance

Function families:

- **Fitting:** `yc_nelson_siegel`, `yc_svensson`, `yc_cubic_spline`, `yc_fit`
- **Extraction:** `yc_predict`, `yc_forward`, `yc_discount`, `yc_par_to_zero`, `yc_zero_to_par`
- **Risk:** `yc_duration`, `yc_bond_duration`, `yc_zspread`, `yc_key_rate_duration`
- **Decomposition:** `yc_pca`, `yc_level_slope_curvature`, `yc_slope`, `yc_carry`

Deps: `cli`, `graphics`, `stats`. R \geq 4.1.0.

One curve object

Every fit returns a `yc_curve`:

`maturities` | `rates` | `params` |
`method`

All downstream functions dispatch on
this class.

Minimal working example

```
library(yieldcurves)

# Fit a Nelson-Siegel curve to the current US Treasury quotes
mats  <- c(0.25, 0.5, 1, 2, 5, 10, 30)
rates <- c(0.0495, 0.048, 0.046, 0.042, 0.041, 0.045, 0.050)
fit   <- yc_nelson_siegel(mats, rates)

# Derive forwards and three-month carry from the fitted curve
yc_forward(fit, maturities = c(2, 5, 10))
yc_carry(fit,   maturities = c(2, 5, 10), horizon = 0.25)
```

All three downstream calls dispatch on the `yc_curve` class, so the same idiom works whether you fit with Nelson-Siegel, Svensson, or a cubic spline.

US Treasury curve through the pandemic cycle

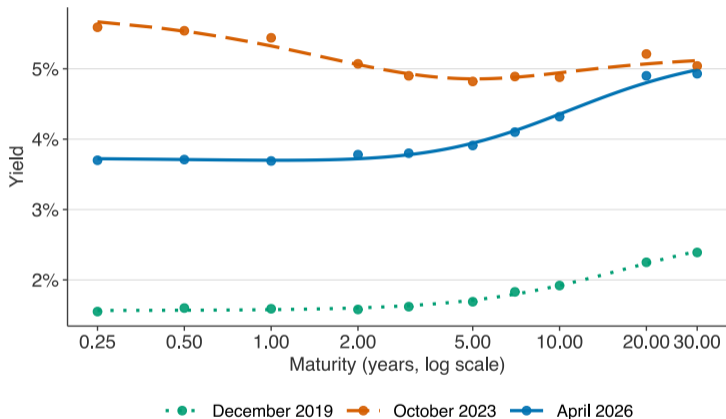
Data. FRED Constant Maturity Treasury rates at ten maturities, monthly from January 2019 to the most recent trading day. Source: Federal Reserve Bank of St. Louis.

Question. *How has the US Treasury term structure evolved across the pandemic monetary cycle?*

Why this case. Flagship global benchmark curve. Three distinct regimes in a five-year window: pre-pandemic flat-and-low, October 2023 peak inversion, current re-steepening. Reproducible with three public function calls from the `yc_curve` interface³.

³ Bauer & Rudebusch (2020), *Interest rates under falling stars*, American Economic Review 110(5).

US Treasury curve: three regimes in five years



US Treasury curves on three dates: pre-pandemic (Dec 2019), rate peak (Oct 2023), and most recent. Points are FRED CMT rates; lines are Nelson-Siegel fits.

What yieldcurves does not yet do

- **Single-date fitting only:** no dynamic term-structure models, no affine Gaussian class
- **No return forecasting:** Cochrane-Piazzesi and similar predictive regressions out of scope
- **Bootstrap stripping:** annual and semi-annual coupons only; exotic schedules need preprocessing
- **No real-time central-bank feed:** users ingest Gurkaynak-Sack-Wright curves separately if desired

v0.2.0 roadmap: arbitrage-free Nelson-Siegel following Christensen et al. (2011), Adrian-Crump-Moench term-premium regression, Cochrane-Piazzesi forecasting helpers, asset-swap spread computation.

Contact, code, paper

Charles Coverdale

`charles.f.coverdale@gmail.com`

`github.com/charlescoverdale/yieldcurves`

`charlescoverdale.github.io/publications`

Acknowledgements. Data: Federal Reserve Bank of St. Louis (FRED, public domain). Thanks to the CRAN maintainers, `cli` developers, and reviewers of *yieldcurves* v0.1.0.



Scan for paper PDF