qfmix reference manual

June 19, 2026

Type Package

Title Sieve Quantile-Function Mixtures and Distributional Synthetic Controls

Version 0.1.2

Description Nonparametric mixtures of quantile functions estimated by a sieve generalized method of L-moments, following Alvarez and Orestes (2024). A target quantile function is approximated by a weighted combination of quantile basis functions (polynomial, shifted-Legendre, Pareto, generalized-extreme-value, or empirical control quantiles); the weights are found by matching sample L-moments to mixture L-moments, which reduces to a convex quadratic program over unconstrained, non-negative, simplex, or ridge weight sets. Inference uses the numerical bootstrap (a simulation from the strong-approximation limit of the quantile process); the unconstrained case is the exact closed form. A distributional synthetic-control wrapper provides counterfactual quantile functions with pointwise variance-component confidence bands. This v0.1 ships the estimator and the pointwise bands; the bias-corrected and Euclidean-ball anti-concentration bands are planned.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1)

Imports quadprog,
stats,
graphics,
grDevices,
utils

Suggests ggplot2,
testthat (>= 3.0.0),
knitr,
rmarkdown

RoxygenNote 7.3.3

Roxygen list(markdown = TRUE)

Config/testthat/edition 3

Config/Needs/website pkgdown

VignetteBuilder knitr

URL https://github.com/kvenkita/qfmix

BugReports https://github.com/kvenkita/qfmix/issues

Contents

`density.qfmix` *Mixture density by quantile inversion*

Description

Inverts the (monotone) mixture quantile function and returns the implied density `q(x) = 1 / Q_mu'(Q_mu^{-1}(x))` (Empirical-Bayes f-modelling, Alvarez & Orestes 2024 sec. 6.1).

Usage

```
## S3 method for class 'qfmix'
density(x, at = NULL, ...)
```

Arguments

`x`	A "`qfmix`" object.
`at`	Numeric points at which to evaluate the density.
`...`	Unused.

Value

A list with `x` and `density`.

`dsc` *Distributional synthetic control with numerical-bootstrap bands*

Description

Builds a counterfactual quantile function for a treated unit as a weighted mixture of control units' quantile functions, with weights fit on the pre-treatment period by sieve GMLM (ridge / simplex). Post-treatment counterfactuals and distributional treatment effects come with **pointwise variance-component** confidence bands from the numerical bootstrap; the effect bands also include the treated unit's own quantile sampling error.

Usage

```
dsc(
  data,
  id_col,
  time_col,
  y_col,
  treated,
  t0,
  controls = NULL,
  constraint = c("ridge", "simplex"),
  M = 1,
  L = NULL,
  weight = "identity",
  boot = TRUE,
  S = 500L,
  level = 0.95,
  band = c("pointwise", "uniform"),
  bias_aware = TRUE,
  n_grid = 500L
)
```

Arguments

`data`	Long data frame.
`id_col`, `time_col`, `y_col`	Column names for unit id, time, and outcome.
`treated`	The treated unit id.
`t0`	Last pre-treatment period (the fitting period).
`controls`	Control unit ids (default: all non-treated units).
`constraint`	Weight set: `"ridge"` (default, `M`) or `"simplex"`.
`M`	Ridge radius. Default `1`.
`L`	Number of L-moments. Default `length(controls) + 2`.
`weight`	L-moment weighting (`"identity"` / `"optimal"`).
`boot`	Compute bootstrap bands. Default `TRUE`.
`S`, `level`	Bootstrap draws and confidence level.

band	"pointwise" (default) or "uniform" (simultaneous sup-t band over the quantile grid).
bias_aware	If TRUE (default), widen the bands by the pre-treatment sieve-approximation residual – a conservative bias bound that fixes the tail under-coverage of variance-only bands where the control mixture cannot represent the treated distribution.
n_grid	Integration grid for the fit.

Value

An object of class "qfmix_dsc".

Inference scope (read this)

The bands carry (i) the weight-estimation variance (numerical bootstrap), (ii) the treated-unit quantile sampling error, and – with bias_aware = TRUE – (iii) a conservative sieve-approximation bias bound estimated from the pre-treatment fit, which restores tail coverage where a finite control mixture cannot represent the treated distribution. Use band = "uniform" for simultaneous (sup-t) coverage over the quantile grid. Still **not** modelled: the control-sampling (stochastic-basis) variance and the exact Euclidean-ball anti-concentration constant for the ridge constraint (the uniform band is the practical substitute). The sqrt(n) scaling uses the treated unit's pre-period sample size, valid under a donor-pool dilution condition (Ferman 2021) that is assumed, not checked.

References

Gunsilius, F. F. (2023). Distributional synthetic controls. *Econometrica* 91, 1105–1117.

---

plot.qfmix *Plot a qfmix fit*

Description

Plot a qfmix fit

Usage

```
## S3 method for class 'qfmix'
plot(x, band = NULL, u = stats::ppoints(200), ...)
```

Arguments

x	A "qfmix" object.
band	Optional qfmix_boot() result (target = "quantile") for a confidence ribbon.
u	Probability grid.
...	Passed to graphics::plot().

Value

Invisibly x.

plot.qfmix_dsc *Plot a distributional synthetic-control result*

Description

Plot a distributional synthetic-control result

Usage

```
## S3 method for class 'qfmix_dsc'
plot(x, period = NULL, which = c("counterfactual", "effect"), ...)
```

Arguments

x	A "qfmix_dsc" object.
period	Which post-period to plot (default first).
which	"counterfactual" (observed vs. counterfactual quantile) or "effect" (the distributional treatment effect with band).
...	Passed to graphics::plot().

Value

Invisibly x.

predict.qfmix *Mixture quantile function at probabilities*

Description

Mixture quantile function at probabilities

Usage

```
## S3 method for class 'qfmix'
predict(object, u = stats::ppoints(100), ...)
```

Arguments

object	A "qfmix" object.
u	Probabilities in [0, 1].
...	Unused.

Value

The fitted mixture quantiles Q_mu(u).

qfmix *Fit a sieve quantile-function mixture (GMLM)*

Description

Approximates a target quantile function by `Q_mu(u) = sum_j lambda_j Q_j(u)` over a quantile basis, estimating the weights by matching the first `L` sample L-moments to the mixture's L-moments (Alvarez & Orestes 2024, eq. 2). The problem reduces to a convex quadratic program.

Usage

```
qfmix(
  q,
  basis = "legendre",
  p = 4L,
  L = NULL,
  constraint = c("unconstrained", "ridge", "nonneg", "simplex"),
  M = 1,
  weight = c("identity", "optimal"),
  trim = c(0, 1),
  monotone = FALSE,
  n_grid = 1000L,
  controls = NULL,
  control = qfmix_control()
)
```

Arguments

`q`	Target: a numeric sample, a quantile function, or a `qfmix` object.
`basis`	A basis name (see `qfmix_basis()`) or a `qfmix_basis` object.
`p`	Sieve dimension (number of basis functions). Default `4`.
`L`	Number of L-moments matched (`>= p`). Default `max(p, p + 2)`.
`constraint`	Weight set: `"unconstrained"`, `"ridge"`, `"nonneg"`, or `"simplex"`. (The `"lasso"` set and an `osqp` backend are planned; this version solves the constrained programs with `quadprog`.)
`M`	Radius for `"ridge"` (default `1`, the DSC choice).
`weight`	L-moment weighting matrix: `"identity"` or `"optimal"` (two-step).
`trim`	Trimming window `c(a, b)` for the L-moment integrals.
`monotone`	If `TRUE`, add grid constraints forcing `Q_mu` nondecreasing.
`n_grid`	Integration grid resolution.
`controls`	For the empirical basis: control observations (see `qfmix_basis()`).
`control`	A `qfmix_control()` list.

Value

An object of class `"qfmix"`.

References

Alvarez, L. A. F. & Orestes, V. M. (2024). Quantile Mixture Models: Estimation and Inference.

Examples

```
set.seed(1)
x <- qexp(runif(500))                       # an exponential sample
fit <- qfmix(x, basis = "legendre", p = 4)
predict(fit, c(0.25, 0.5, 0.75))
```

`qfmix_basis` *Construct a quantile basis*

Description

Construct a quantile basis

Usage

```
qfmix_basis(
  type = c("polynomial", "legendre", "pareto", "gev", "empirical"),
  p = 4L,
  controls = NULL
)
```

Arguments

`type`	One of `"polynomial"`, `"legendre"`, `"pareto"`, `"gev"`, `"empirical"`.
`p`	Number of basis functions (sieve dimension). Ignored for `"empirical"` (set by the controls).
`controls`	For `"empirical"`: a list of numeric vectors (or a matrix with one column per control unit) of control observations whose empirical quantile functions form the basis.

Value

An object of class `"qfmix_basis"` with `eval(u)` and `deriv(u)`.

qfmix_boot *Numerical-bootstrap inference for a qfmix fit*

Description

Runs Algorithm 1 of Alvarez & Orestes (2024): draws `Z_s ~ N(0, V_hat)` from the L-moment covariance and re-solves the (local) program to obtain the sampling law of the weights, then forms confidence bands for the weights or for the mixture quantile function.

Usage

```
qfmix_boot(
  fit,
  S = 500L,
  kappa = "sqrt_n",
  level = 0.95,
  target = c("quantile", "weights"),
  band = c("pointwise", "uniform"),
  u = ppoints(100)
)
```

Arguments

`fit`	A `"qfmix"` object (fit on a numeric sample, so `n` is known).
`S`	Number of simulation draws. Default `500`.
`kappa`	Local-set scaling for constrained sets; `"sqrt_n"` (default) or a number. Ignored when `constraint = "unconstrained"`.
`level`	Confidence level. Default `0.95`.
`target`	`"quantile"` (bands for `Q_mu(u)`) or `"weights"`.
`band`	`"pointwise"` (default) or `"uniform"`. The uniform band is a simultaneous (sup-t) band: a single critical value from the bootstrap distribution of the max studentized deviation, valid jointly over the grid by the anti-concentration argument of Chernozhukov, Chetverikov & Kato (the hyperrectangle validity the paper cites via Nazarov).
`u`	Probability grid for `target = "quantile"`.

Value

A list with the point estimate, `lower`/`upper` bands, and the simulated `delta`.

qfmix_control *Control parameters for* qfmix()

Description

Control parameters for qfmix()

Usage

```
qfmix_control(ridge_jitter = 1e-08, mono_grid = 200L)
```

Arguments

ridge_jitter	Small value added to the QP Hessian diagonal for numerical positive-definiteness. Default 1e-8.
mono_grid	Number of grid points for the monotonicity constraint D lambda >= 0. Default 200.

Value

A list of class "qfmix_control".

shifted_legendre *Orthonormal shifted Legendre polynomials on the unit interval*

Description

Columns 1..L are phi_0, ..., phi_{L-1} with phi_k(u) = sqrt(2k+1) P_k(2u - 1), orthonormal on [0, 1]. The generalized moment integral Q(u) phi_{r-1}(u) du is the projection of the quantile function onto phi_{r-1}; it equals Hosking's (1990) classical L-moment up to the sqrt(2r-1) orthonormalisation factor (phi_0 = 1, so the first equals the mean). These orthonormal moments – not Hosking's classical scaling – are what qfmix matches and reports.

Usage

```
shifted_legendre(u, L)
```

Arguments

u	Numeric vector in [0, 1].
L	Number of polynomials.

Value

An length(u) x L matrix.

sim_qmix *Simulate from a known quantile-function mixture*

Description

Draws `u ~ U(0,1)` and returns `x = Q_mu(u)` for a mixture `Q_mu = sum_j lambda_j Q_j` on the chosen basis – a ground truth for recovery checks.

Usage

```
sim_qmix(n = 1000L, basis = "pareto", p = 3L, lambda = NULL)
```

Arguments

`n`	Sample size.
`basis`	Basis name (see `qfmix_basis()`).
`p`	Sieve dimension.
`lambda`	True weights (length `p`); default a decaying non-negative vector.

Value

A numeric sample with attribute `"true_lambda"`.

summary.qfmix *Summarize a qfmix fit*

Description

Summarize a qfmix fit

Usage

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

Arguments

`object`	A `"qfmix"` object.
`...`	Unused.

Value

An object of class `"summary.qfmix"`.

Index