

Package ‘rifexpectile’

September 1, 2026

Title Density-Free RIF Decompositions for Unconditional Expectiles

Version 0.1.1

Description Implements a density-free recentered influence function (RIF) regression framework for unconditional expectiles, and embeds it in a two-sample Oaxaca-Blinder decomposition indexed continuously by the expectile level. Unlike quantile-based RIF decompositions, which require estimating an inverse density term at each quantile, the expectile RIF depends only on primitive moments of the outcome distribution and requires no density estimation, no bandwidth selection, and no kernel smoothing. The package provides expectile estimation by iteratively reweighted least squares, closed-form RIF construction, two-sample composition/structure decomposition across a grid of expectile levels, bootstrap-based inference, and plotting methods. The underlying methodology is described in Ndoye (2025), “Semi-Nonparametric Expectile RIF Regression for Distributional Decomposition,” presented at the 2025 World Congress of the Econometric Society, Seoul, Korea, <<https://www.econometricsociety.org/regional-activities/conference-papers/view/282/943>>.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports stats, graphics

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

VignetteBuilder knitr

Config/testthat/edition 3

URL <https://github.com/aandoye/rifexpectile>

BugReports <https://github.com/aandoye/rifexpectile/issues>

Config/roxygen2/version 8.1.0

LazyData true

NeedsCompilation no

Author Abdoul Aziz Ndoye [aut, cre]

Maintainer Abdoul Aziz Ndoye <abdoulaziz1.ndoye@gmail.com>
Repository CRAN
Date/Publication 2026-08-31 22:00:21 UTC

Contents

plot.rifdecomp	2
rif_decompose	3
rif_expectile	5
rif_expectile_fit	6
rif_expectile_grid	7
simulate_income_data	8
wage_gap_sim	9
Index	10

plot.rifdecomp	<i>Plot a density-free RIF decomposition</i>
----------------	--

Description

Plots the total, composition, and structure effects across the grid of expectile levels, with bootstrap confidence bands if available.

Usage

```
## S3 method for class 'rifdecomp'  
plot(x, which = c("total", "composition", "structure"), ...)
```

Arguments

- x An object of class "rifdecomp", as returned by [rif_decompose](#).
- which One of "total", "composition", or "structure", the component to plot.
- ... Additional arguments passed to [plot](#).

Value

Invisibly returns x (the input "rifdecomp" object, unmodified). Called primarily for its side effect of producing a plot on the current graphics device.

rif_decompose	<i>Two-sample RIF-Oaxaca-Blinder decomposition of unconditional expectiles</i>
---------------	--

Description

Decomposes the gap in the α -expectile of an outcome between two groups into a composition (endowment) effect and a structure (coefficient) effect, at each of a grid of expectile levels, using the density-free expectile RIF of [rif_expectile](#) embedded in a standard two-fold Oaxaca-Blinder decomposition.

Usage

```

rif_decompose(
  formula,
  data,
  group,
  ref_group = NULL,
  alpha = seq(0.1, 0.9, by = 0.1),
  weights = NULL,
  boot_reps = 200,
  conf_level = 0.95,
  seed = NULL
)

```

Arguments

formula	A formula of the form $y \sim x_1 + x_2 + \dots$ giving the outcome and covariates. An intercept is always included.
data	A data frame containing the outcome, covariates, group indicator, and (optionally) weights, for both groups.
group	A string naming the column of data identifying group membership. Must have exactly two distinct values.
ref_group	The value of group to use as the reference group B (whose coefficients are used for the composition term). If NULL (the default), the first value in <code>sort(unique(data[[group]]))</code> is used.
alpha	Numeric vector of expectile levels in $(0, 1)$ at which to compute the decomposition. Default <code>seq(0.1, 0.9, by = 0.1)</code> .
weights	Optional string naming a column of data with nonnegative sampling weights. If NULL (the default), the decomposition is unweighted.
boot_reps	Integer, number of bootstrap replications for standard errors and confidence intervals, stratified by group (each replication resamples n_A observations from group A and n_B from group B, with replacement). Default 200. Set to 0 to skip bootstrap inference (point estimates only, computed much faster).
conf_level	Confidence level for the bootstrap percentile intervals. Default 0.95.

seed Optional integer seed for the bootstrap random number generator, for reproducibility.

Details

For each group $g \in \{A, B\}$ and expectile level α , the α -expectile is estimated by `rif_expectile_fit`, the recentered influence function is constructed by `rif_expectile`, and a linear regression of the RIF on the covariates in formula (including an intercept) is estimated by (weighted) least squares. Writing $\bar{X}_g$ for the sample mean covariate vector and $\hat{\beta}_{g,\alpha}$ for the estimated coefficients in group g , with group B as the reference group, the two-fold decomposition is

$$\underbrace{\hat{e}_{A,\alpha} - \hat{e}_{B,\alpha}}_{\text{Total}} = \underbrace{(\bar{X}_A - \bar{X}_B)' \hat{\beta}_{B,\alpha}}_{\text{Composition}} + \underbrace{\bar{X}_A' (\hat{\beta}_{A,\alpha} - \hat{\beta}_{B,\alpha})}_{\text{Structure}}.$$

Because an intercept is included, the Total term above coincides exactly (to floating-point precision, not only asymptotically) with the difference of the two sample expectiles; see the package vignette for a demonstration and discussion.

Value

An object of class "rifdecomp", a list with components:

alpha The grid of expectile levels used.

total, composition, structure Numeric vectors (one entry per value of alpha) of the aggregate total, composition, and structure effects.

se_total, se_composition, se_structure Bootstrap standard errors, or NULL if boot_reps = 0.

ci_total, ci_composition, ci_structure Two-column matrices of bootstrap percentile confidence interval bounds, or NULL if boot_reps = 0.

beta_ref, beta_other Matrices of covariate-level regression coefficients (rows = covariates including the intercept, columns = alpha grid) for the reference and comparison groups.

detail_composition, detail_structure Matrices of covariate-level composition and structure contributions (same layout as beta_ref), so that each column sums to the corresponding aggregate composition/structure entry.

group_labels The two group values, in (other, reference) order.

call The matched call.

Examples

```
set.seed(42)
n <- 500
dat <- data.frame(
  grp = rep(c("2000", "2010"), each = n),
  x1 = rnorm(2 * n),
  x2 = rbinom(2 * n, 1, 0.4)
)
dat$y <- with(dat, 1 + 0.5 * x1 + 0.3 * x2 +
  0.2 * (grp == "2010") * x1 + rnorm(2 * n, sd = 0.5))
fit <- rif_decompose(y ~ x1 + x2, data = dat, group = "grp",
```

```

ref_group = "2000", alpha = c(0.1, 0.5, 0.9),
boot_reps = 50, seed = 1)

print(fit)

```

rif_expectile

*Closed-form recentered influence function for the expectile***Description**

Constructs the recentered influence function (RIF) of the α -expectile at each observation, using the closed-form expression that depends only on primitive moments of the outcome and requires no density estimation.

Usage

```
rif_expectile(y, e_alpha, alpha, w = NULL)
```

Arguments

y	Numeric vector of observations.
e_alpha	Numeric scalar, typically the output of rif_expectile_fit applied to the same y (and the same weights, if any).
alpha	Numeric scalar in (0, 1), the expectile level.
w	Optional numeric vector of sampling weights, the same length as y. If NULL (the default), all observations are equally weighted.

Details

The population influence function of the α -expectile is

$$\text{IF}(y; e_\alpha, F_Y) = \frac{|\alpha - 1\{y < e_\alpha\}|(y - e_\alpha)}{E[|\alpha - 1\{Y < e_\alpha\}|]},$$

and the recentered influence function is $\text{RIF}(y; e_\alpha, F_Y) = e_\alpha + \text{IF}(y; e_\alpha, F_Y)$. The empirical RIF used here replaces population quantities with their weighted sample analogues. By construction, and unlike the quantile RIF, this expression never requires an estimated density: the denominator is a sample average of an asymmetric 0/1-type weight, not a density evaluation. When y and e_alpha come from the same sample with the same weights, and an intercept is included in any subsequent regression of the RIF on covariates, the sample mean of the returned vector equals e_alpha exactly (a finite-sample identity, not an asymptotic approximation); see the package vignette for a numerical illustration of this identity and its consequence for two-sample decompositions.

Value

A numeric vector of the same length as y (after removing non-finite entries), the recentered influence function evaluated at each observation.

See Also

[rif_expectile_fit](#), [rif_decompose](#)

Examples

```
set.seed(1)
y <- rlnorm(1000)
e50 <- rif_expectile_fit(y, 0.5)
r <- rif_expectile(y, e50, 0.5)
mean(r) - e50 # approximately 0, exactly 0 up to floating point
```

rif_expectile_fit	<i>Sample expectile via iteratively reweighted least squares</i>
-------------------	--

Description

Computes the sample α -expectile of a numeric vector by iteratively reweighted least squares (IRLS), the standard algorithm for the asymmetric least-squares problem of Newey and Powell (1987).

Usage

```
rif_expectile_fit(y, alpha, w = NULL, tol = 1e-10, maxit = 500)
```

Arguments

y	Numeric vector of observations.
alpha	Numeric scalar in $(0, 1)$, the expectile level. $\alpha = 0.5$ recovers the sample mean exactly.
w	Optional numeric vector of nonnegative sampling weights, the same length as y. If NULL (the default), all observations are equally weighted.
tol	Convergence tolerance on successive iterates. Default $1e-10$.
maxit	Maximum number of IRLS iterations. Default 500.

Details

The α -expectile e_α of a random variable Y is defined as the unique solution to the asymmetric least-squares minimization problem

$$e_\alpha = \arg \min_t E \left[|\alpha - 1\{Y < t\}|(Y - t)^2 \right].$$

Differentiating and setting the first-order condition to zero gives an estimating equation solved here by IRLS: starting from the sample mean, each iteration recomputes asymmetric weights $w_i = |\alpha - 1\{y_i < \hat{e}\}|$ and updates $\hat{e} \leftarrow \sum w_i y_i / \sum w_i$ until convergence.

Value

A numeric scalar, the estimated α -expectile.

References

Newey, W. K. and Powell, J. L. (1987). Asymmetric least squares estimation and testing. *Econometrica*, 55(4), 819-847.

Examples

```
set.seed(1)
y <- rlnorm(1000)
rif_expectile_fit(y, 0.5)      # approximately mean(y)
rif_expectile_fit(y, 0.9)      # upper-tail location
```

rif_expectile_grid	<i>Vectorized sample expectiles over a grid of levels</i>
--------------------	---

Description

Convenience wrapper computing [rif_expectile_fit](#) at several expectile levels.

Usage

```
rif_expectile_grid(y, alpha, w = NULL, ...)
```

Arguments

y	Numeric vector of observations.
alpha	Numeric vector of expectile levels in (0, 1).
w	Optional numeric vector of sampling weights (see rif_expectile_fit).
...	Additional arguments passed to rif_expectile_fit .

Value

A named numeric vector of estimated expectiles, one per element of alpha.

Examples

```
set.seed(1)
y <- rlnorm(1000)
rif_expectile_grid(y, c(0.1, 0.5, 0.9))
```

simulate_income_data *Simulate two-group income data for examples and testing*

Description

Generates a synthetic two-group data set with a log-normal outcome and covariates, group-specific coefficients, and (optionally) sampling weights, for use in examples, tests, and the package vignette. No real or proprietary data are used or required anywhere in this package.

Usage

```
simulate_income_data(
  n = 1000,
  group_diff = 0.15,
  shift = 0.3,
  sd = 0.4,
  with_weights = TRUE,
  seed = NULL
)
```

Arguments

n	Integer, number of observations per group. Default 1000.
group_diff	Numeric scalar controlling how much the two groups' slope on x1 differs (a stylized "returns" shift analogous to a structure effect). Default 0.15.
shift	Numeric scalar added to the second group's intercept (a stylized aggregate shift). Default 0.3.
sd	Standard deviation of the residual on the log scale. Default 0.4.
with_weights	Logical; if TRUE (the default), a weight column is included, drawn to be mildly correlated with x2 so that weighted and unweighted decompositions can differ, mimicking a stylized survey design effect.
seed	Optional integer seed for reproducibility.

Details

The outcome is generated as

$$\log y_i = \beta_{0,g} + \beta_{1,g}x_{1i} + \beta_{2,g}x_{2i} + \varepsilon_i,$$

with group-specific intercepts and slopes controlled by `shift` and `group_diff`, and ε_i drawn from a normal distribution with standard deviation `sd`, so that y_i is log-normal within each group and each covariate combination.

Value

A data frame with columns `group` (a two-level factor, "A" and "B"), `y` (the level of the outcome), `logy` (its log), `x1` (a continuous covariate), `x2` (a binary covariate), and, if `with_weights = TRUE`, `weight`.

Examples

```
dat <- simulate_income_data(n = 300, seed = 1)
fit <- rif_decompose(logy ~ x1 + x2, data = dat, group = "group",
                     ref_group = "A", boot_reps = 50, seed = 2)
print(fit)
```

wage_gap_sim

*Simulated two-wave income data***Description**

A synthetic data set with 2,000 observations per wave ("2000" and "2010"), generated by [simulate_income_data](#), for use in examples and the package vignette. No real or proprietary microdata are used anywhere in this package; this is a purely synthetic illustration of a stylized structure effect (a group-specific slope shift) combined with a stylized aggregate shift.

Usage

```
wage_gap_sim
```

Format

A data frame with 4,000 rows and 6 columns:

group Factor with levels "2000" and "2010".

y Numeric, the outcome level (log-normal by construction).

logy Numeric, $\log(y)$.

x1 Numeric, a continuous covariate.

x2 Integer (0/1), a binary covariate.

weight Numeric, a stylized sampling weight mildly correlated with x2, included so that weighted and unweighted decompositions can be compared.

Source

Generated by data-raw/wage_gap_sim.R; see [simulate_income_data](#) for the generating process.

Examples

```
data(wage_gap_sim)
fit <- rif_decompose(logy ~ x1 + x2, data = wage_gap_sim, group = "group",
                     ref_group = "2000", boot_reps = 50, seed = 1)
print(fit)
```

Index

* **datasets**

wage_gap_sim, [9](#)

plot, [2](#)

plot.rifdecomp, [2](#)

rif_decompose, [2](#), [3](#), [6](#)

rif_expectile, [3](#), [4](#), [5](#)

rif_expectile_fit, [4-6](#), [6](#), [7](#)

rif_expectile_grid, [7](#)

simulate_income_data, [8](#), [9](#)

wage_gap_sim, [9](#)