Package ‘rbreak’

May 9, 2026

Title Restricted Structural Change Models

Version 1.0.7

Maintainer Zhongjun Qu <qu@bu.edu>

Description Methods for detecting structural breaks and estimating break locations for linear multiple regression models under general linear restrictions on the coefficient vector. Restrictions can be within regimes, across regimes, or both, and are supported in two forms: an affine parameterization (Form A: delta = S*theta + s) and explicit linear constraints (Form B: R*delta = r). Provides break date estimation with confidence intervals, a restricted sup-F test for the null of no structural change, simulation of critical values by Monte Carlo, and a bootstrap restart procedure to reduce the risk of convergence to spurious local optima. Also implements a generalized regression tree (linear model tree) procedure where each leaf contains a linear regression model rather than a local average. Reference: Perron, P., and Qu, Z. (2006). 'Estimating Restricted Structural Change Models.' Journal of Econometrics, 134(2), 373-399. <doi:10.1016/j.jeconom.2005.06.030>.

License GPL (>= 3)

Depends R (>= 4.3.0)

Suggests knitr, parallel, pbapply, rmarkdown

VignetteBuilder knitr

Encoding UTF-8

NeedsCompilation no

RoxygenNote 7.3.2

Imports MASS, stats

LazyData true

Author Zhongjun Qu [aut, cre],
Cheolju Kim [aut]

Repository CRAN

Date/Publication 2026-03-27 10:20:03 UTC

Contents

`build_null_from_R` *Build no-break (null) restrictions on β implied by $R\delta = r$*

Description

Given linear restrictions on the break-model coefficient vector

$$R\delta = r, \qquad \delta \in \mathbb{R}^{(m+1)q},$$

this function constructs the implied restriction system under the no-break null, where all regimes share a common coefficient vector $\beta \in \mathbb{R}^q$.

Usage

```
build_null_from_R(R, r, m, q, tol)
```

Arguments

`R`	Numeric matrix of dimension $k \times (m+1)q$ in $R\delta = r$.
`r`	Numeric vector of length k in $R\delta = r$.
`m`	Integer. Number of breaks (so number of regimes is $m+1$).
`q`	Integer. Number of regressors per regime.
`tol`	Numeric tolerance used in rank decisions. Default is `1e-10`.

Details

Under no break, $\delta = N\beta$ with $N = \mathbf{1}_{m+1} \otimes I_q$. Hence the implied null restrictions are

$$(RN)\beta = r.$$

The function reduces to independent rows and checks feasibility. It returns $R_0\beta = r_0$ with R_0 having q columns.

Value

A list with components:

- `R0`: Numeric matrix of dimension $k_0 \times q$ giving the reduced null restrictions $R_0\beta = r_0$.
- `r0`: Numeric vector of length k_0.
- `A_full`: The unreduced matrix RN (dimension $k \times q$).
- `rankA`: Integer. Numerical rank of RN.
- `p0`: Integer. Number of free parameters under the null: $p_0 = q - \mathrm{rank}(RN)$.
- `restricted_null`: Logical. `TRUE` iff $p_0 < q$.

Examples

```
m <- 1
q <- 2
R <- matrix(c(1, -1, 0, 0,
              0, 0, 1, -1), nrow = 2, byrow = TRUE)
r <- c(0, 0)
out <- build_null_from_R(R, r, m = m, q = q)
out$p0
out$R0
out$r0
```

`build_null_from_S` *Build no-break (null) restrictions implied by* $\delta = S\theta + s$

Description

Given an affine restriction set for the break-model coefficients,

$$\delta = S\theta + s,$$

this function computes an affine parameterization for the *no-break* (null) model in terms of the common coefficient vector $\beta \in \mathbb{R}^q$:

$$\beta = S_0\gamma + s_0.$$

Usage

```
build_null_from_S(S, s, m, q, tol)
```

Arguments

- `S` Numeric matrix of dimension $(m+1)q \times p_A$ in $\delta = S\theta + s$.
- `s` Numeric vector of length $(m+1)q$ in $\delta = S\theta + s$.
- `m` Integer. Number of breaks (so number of regimes is $m+1$).
- `q` Integer. Number of regressors per regime.
- `tol` Numeric tolerance used in rank decisions. Default is `1e-10`.

Details

The null model is "no break" (all regimes share the same β) possibly with additional restrictions implied by S, s. The function returns S_0, s_0 and the number of free parameters under the null.

Under no break, the stacked coefficient vector satisfies

$$\delta = N\beta, \qquad N = \mathbf{1}_{m+1} \otimes I_q.$$

Combining with $\delta = S\theta + s$ and eliminating θ yields a linear system on β. This function constructs an affine parameterization $\beta = S_0\gamma + s_0$ for that system, or returns the unrestricted no-break model when no additional restrictions are implied.

The function stops with an error if the implied no-break system is infeasible.

Value

A list with components:

- `S0`: Numeric matrix of dimension $q \times p_0$ in $\beta = S_0\gamma + s_0$. If the null is unrestricted, `S0` is `diag(q)`.
- `s0`: Numeric vector of length q in $\beta = S_0\gamma + s_0$. If the null is unrestricted, `s0` is a zero vector.
- `p0`: Integer. Number of free parameters under the null (equal to `ncol(S0)`).
- `rA`: Integer. Rank of the implied restriction system on β (so $p_0 = q - r_A$).
- `restricted_null`: Logical. `TRUE` if the null imposes restrictions beyond no-break (i.e., `p0 < q`), else `FALSE`.

Examples

```
m <- 1
q <- 2
S <- matrix(c(1, 0,
              1, 0,
              0, 1,
              0, 1), nrow = 4, byrow = TRUE)
s <- rep(0, 4)
out <- build_null_from_S(S, s, m = m, q = q)
out$p0
out$S0
out$s0
```

`deviation` *Taylor Rule Deviation Data*

Description

A dataset containing quarterly Taylor-rule deviations used in the restricted structural break illustration.

Usage

```
deviation
```

Format

A data frame with 193 rows and 2 variables:

date Quarterly date labels.

y Taylor-rule deviation series used in the application example.

Source

Constructed from the replication files for Nikolsko-Rzhevskyy et al. (2014).

References

Nikolsko-Rzhevskyy, A., Papell, D. H., and Prodan, R. (2014). Deviations from rules-based policy and their effects. *Journal of Economic Dynamics and Control*, 49, 4–17.

Examples

```
data(deviation)
head(deviation)
```

`estco` *Estimation with Standard Errors and Confidence Intervals (Form A)*

Description

Estimates break dates and coefficients under Form A restrictions, computes HAC standard errors, and constructs confidence intervals for break dates.

Usage

```
estco(m, q, z, y, trm, robust, prewhit, hetvar, S, s, Verbose)
```

Arguments

`m`	An integer of number of breaks.
`q`	An integer of number of regressors.
`z`	A numeric matrix ($T \times q$) of regressor.
`y`	A numeric vector ($T \times 1$) of dependent variable.
`trm`	A numeric value of trimming parameter.
`robust`	An integer (0 or 1). If 1, use HAC standard errors.
`prewhit`	An integer (0 or 1). If 1, apply AR(1) prewhitening (only when robust=1).
`hetvar`	An integer (0 or 1). If 1, allow heteroskedastic variance across segments.
`S`	Numeric matrix defining the linear reparameterization for Form A: $\delta = S\theta + s$, where δ stacks the q coefficients for each of the $m+1$ regimes (so $\delta \in \mathbb{R}^{(m+1)q}$). Must have full column rank. If there are no restrictions, set `S = diag((m+1) * q)`.
`s`	Numeric vector of constants in $\delta = S\theta + s$, of length $(m+1)q$. If there are no restrictions, set `s = rep(0, (m+1) * q)`.
`Verbose`	Logical. If `TRUE`, print detailed output (default `FALSE`).

Value

A list (returned invisibly) with the following components:

`breaks` An integer vector of length m giving the estimated break dates.

`coefficients` A numeric vector of length $(m+1)q$ giving the regime-specific coefficient estimates.

`vcov` A $(m+1)q \times (m+1)q$ variance-covariance matrix of `coefficients`.

`ci_breaks` An $m \times 4$ matrix of confidence intervals for the break dates. The first two columns give the 95\ (lower bound, upper bound) and the last two columns give the 90\ (lower bound, upper bound).

References

Andrews, D. W. K. (1991). Heteroskedasticity and autocorrelation consistent covariance matrix estimation. *Econometrica*, 59(3), 817-858.

Perron, P., and Qu, Z. (2006). Estimating restricted structural change models. *Journal of Econometrics*, 134(2), 373-399.

Examples

```
data(example)
y <- example$y
z <- matrix(1, length(y), 1)
m <- 3
q <- 1
trm <- 0.10
S <- matrix(c(1, 0,
```

```
              0, 1,
              1, 0,
              0, 1), nrow = 4, byrow = TRUE)
s <- rep(0, 4)
fit <- estco(m, q, z, y, trm, 0, 0, 1, S, s, FALSE)
fit$breaks
```

---

estco2 *Estimation with Standard Errors and Confidence Intervals (Form B)*

Description

Estimates break dates and coefficients under Form B restrictions, computes HAC standard errors, and constructs confidence intervals for break dates.

Usage

```
estco2(m, q, z, y, trm, robust, prewhit, hetvar, R, r, Verbose)
```

Arguments

`m`	An integer of number of breaks.
`q`	An integer of number of regressors.
`z`	A numeric matrix ($T \times q$) of regressor.
`y`	A numeric vector ($T \times 1$) of dependent variable.
`trm`	A numeric value of trimming parameter.
`robust`	An integer (0 or 1). If 1, use HAC standard errors.
`prewhit`	An integer (0 or 1). If 1, apply AR(1) prewhitening (only when robust=1).
`hetvar`	An integer (0 or 1). If 1, allow heteroskedastic variance across segments.
`R`	Numeric matrix defining linear restrictions for Form B, $R\delta = r$, where $\delta \in \mathbb{R}^{(m+1)q}$ stacks the q coefficients for each of the $m+1$ regimes. Each row of `R` represents one linear restriction. Must have full row rank. Do not use this form if there are no restrictions; use Form A instead.
`r`	Numeric vector of restriction constants in $R\delta = r$, with length equal to `nrow(R)`.
`Verbose`	Logical. If `TRUE`, print detailed output (default `FALSE`).

Value

A list (returned invisibly) with the following components:

`breaks` An integer vector of length m giving the estimated break dates.

`coefficients` A numeric vector of length $(m+1)q$ giving the regime-specific coefficient estimates.

`vcov` A $(m+1)q \times (m+1)q$ variance-covariance matrix of `coefficients`.

`ci_breaks` An $m \times 4$ matrix of confidence intervals for the break dates. The first two columns give the 95\ (lower bound, upper bound) and the last two columns give the 90\ (lower bound, upper bound).

References

Perron, P., and Qu, Z. (2006). Estimating restricted structural change models. *Journal of Econometrics*, 134(2), 373-399.

Examples

```
data(example)
y <- example$y
z <- matrix(1, length(y), 1)
m <- 3
q <- 1
trm <- 0.10
R <- matrix(c(1, 0, -1, 0,
              0, 1,  0, -1), nrow = 2, byrow = TRUE)
r <- c(0, 0)
fit <- estco2(m, q, z, y, trm, 0, 0, 1, R, r, FALSE)
fit$breaks
```

`example` *Example Time Series Data*

Description

A dataset containing 120 observations of a time series variable for illustrating restricted structural change models.

Usage

```
example
```

Format

A data frame with 120 rows and 1 variable:

y Numeric vector containing the time series observations

Details

This dataset is used in the examples in Perron and Qu (2006).

Source

Included with the rbreak package for illustration purposes.

References

Perron, P., and Qu, Z. (2006). Estimating restricted structural change models. *Journal of Econometrics*, 134(2), 373-399.

Examples

```
# Load the example data
data(example)


# Example: Test for 3 breaks with intercept only model
y <- example$y
z <- matrix(1, length(y), 1)  # Intercept only

# Form B restrictions: coefficients in regimes 1 and 3 are equal
m <- 3
q <- 1
R <- matrix(c(1, 0, -1, 0, 0, 1, 0, -1), nrow = 2, byrow = TRUE)
r <- c(0, 0)

result <- mainp(m = m, q = q, z = z, y = y, trm = 0.10,
                robust = 1, prewhit = 0, hetvar = 1,
                R = R, r = r,
                doestim = 1, dotest = 1, docv = 0,
                formb = 1)
```

`ltree` *Linear Regression Tree*

Description

Constructs a regression tree where each terminal node fits a linear model. The tree is built by greedy recursive binary partitioning: at each node, every partition variable is considered as a candidate split axis. For numeric variables, data are sorted by the candidate variable and an optimal one-break point is found (minimising SSR via `ssr()`). For categorical variables, all possible binary partitions of the levels are enumerated. BIC is used to decide whether splitting improves the fit. The process continues until no further BIC improvement is possible.

Usage

```
ltree(y, z, partition_vars, cat_vars, trm, max_depth, min_obs,
  prune, verbose)
```

Arguments

`y`	Numeric vector of length n. Dependent variable.
`z`	Numeric matrix ($n \times q$). Regressors for the linear model fitted in each segment. Include an intercept column (e.g. `cbind(1, X)`) if desired.
`partition_vars`	Data frame or matrix ($n \times p$). Variables considered for splitting. May contain both numeric and categorical (factor/character) columns when passed as a data frame. If `NULL` (default), `z` is used (all numeric).

`cat_vars` Integer vector of column indices in `partition_vars` that should be treated as categorical. Only needed when `partition_vars` is a matrix; ignored when it is a data frame (factor/character columns are auto-detected). Default `NULL`.

`trm` Numeric. Trimming parameter: minimum fraction of the current node's observations that must remain in each child segment after a split (default 0.15).

`max_depth` Integer. Maximum depth of the tree (default 10).

`min_obs` Integer or `NULL`. Minimum number of observations in a segment. If `NULL` (default), set to `max(q + 1, 5)`.

`prune` Logical. If `TRUE` (default), apply cost-complexity pruning after growing the full tree to remove spurious splits. If `FALSE`, return the greedily grown tree without pruning.

`verbose` Logical. If `TRUE`, print progress messages (default `FALSE`).

Details

Each node in the tree is a list with `type = "leaf"` or `type = "internal"`.

A leaf node contains: `n`, `indices`, `coefficients`, `ssr`, `bic`.

An internal node contains: `split_var`, `split_var_name`, `split_type` (`"numeric"` or `"categorical"`), `split_val` (for numeric) or `split_left_levels` (for categorical), `bic_nosplit`, `bic_split`, `left`, `right`, `n`.

The BIC criterion used is:

$$\mathrm{BIC} = n \log(\mathrm{SSR}/n) + k \log(n),$$

where $k = q$ for no split and $k = 2q$ for a single split.

Value

An object of class `"ltree"`, a list containing:

tree The recursive tree structure (see Details).

n Total number of observations.

q Number of regressors.

p Number of partition variables.

pv_names Names of the partition variables.

z_names Names of the regressors.

is_cat Logical vector indicating which partition variables are categorical.

trm Trimming parameter used.

max_depth Maximum depth used.

min_obs Minimum observations per segment used.

call The matched call.

S3 Methods

`print(x, ...)` Prints a summary of the fitted tree, including the number of leaves, tree depth, and the split structure.

`predict(object, newz, new_partition_vars = NULL, ...)` Returns predicted values for new data. `newz` is a numeric matrix of regressors and `new_partition_vars` is an optional data frame of partition variables (defaults to `newz`).

`plot(x, data = x$partition_vars, cex = 0.85, main = "", ...)` Draws a tree diagram of the fitted segmented linear regression tree. Internal nodes are shown as beige rectangles, leaf nodes as green ovals. `data` defaults to the partition variables stored in the fitted object, so `plot(res)` works out of the box. Right-side levels of categorical splits are labelled automatically.

References

Perron, P., and Qu, Z. (2006). Estimating restricted structural change models. *Journal of Econometrics*, 134(2), 373-399.

Quinlan, J.R. (1992). Learning with continuous classes. *In: Proceedings of the Australian Joint Conference on Artificial Intelligence*, World Scientific, pp 343–348.

Examples

```
data(ltree1)
fit <- ltree(ltree1$y[1:120],
             ltree1[1:120, c("x1", "x2", "x3", "x4")],
             trm = 0.1, max_depth = 2, prune = TRUE)
print(fit)
```

`ltree1` *Linear Model Tree Data*

Description

A dataset containing one realization from the piecewise linear design used to illustrate the generalized regression tree (linear model tree).

Usage

```
ltree1
```

Format

A data frame with 1500 rows and 5 variables:

y Response variable.

x1 Continuous regressor.

x2 Continuous regressor.

x3 Continuous regressor.

x4 Categorical regressor.

Details

The data are generated from a segmented linear regression function with true splits at $X_1 = 10$, $X_2 = 10$, $X_2 = 15$, and $X_4 \in \{a, b\}$ versus $\{c\}$.

Source

Generated from the design in Zheng and Chen (2019) and included for illustration purposes.

References

Quinlan, J.R. (1992). Learning with continuous classes. *In: Proceedings of the Australian Joint Conference on Artificial Intelligence*, World Scientific, pp 343–348.

Zheng, X. and Chen, S. X. (2019). Partitioning structure learning for segmented linear regression trees. *Advances in Neural Information Processing Systems* (NeurIPS 2019).

Examples

```
data(ltree1)
fit <- ltree(ltree1$y, ltree1[, c("x1", "x2", "x3", "x4")],
             trm = 0.1, max_depth = 10, prune = TRUE)
print(fit)
```

`ltree2` *Smooth Surface Regression Tree Example Data*

Description

A dataset containing one fixed sample for the smooth-surface generalized regression tree illustration.

Usage

```
ltree2
```

Format

A data frame with 1000 rows and 3 variables:

y Response variable with Gaussian noise.

x1 Continuous regressor.

x2 Continuous regressor.

Details

The underlying regression surface is

$$m(X) = \max\{e^{-10X_1^2}, e^{-50X_2^2}, 1.25e^{-5(X_1^2+X_2^2)}\}.$$

Source

Generated from the design in Zheng and Chen (2019) and included for illustration purposes.

References

Quinlan, J.R. (1992). Learning with continuous classes. *In: Proceedings of the Australian Joint Conference on Artificial Intelligence*, World Scientific, pp 343–348.

Zheng, X. and Chen, S. X. (2019). Partitioning structure learning for segmented linear regression trees. *Advances in Neural Information Processing Systems* (NeurIPS 2019).

Examples

```
data(ltree2)
fit <- ltree(ltree2$y,
             cbind(1, ltree2$x1, ltree2$x2),
             partition_vars = ltree2[, c("x1", "x2")],
             trm = 0.1, max_depth = 10, min_obs = 10, prune = TRUE)
print(fit)
```

`mainp` *Main Procedure for Restricted Structural Change Analysis*

Description

Function for restricted structural change models that performs estimation, hypothesis testing, and critical value simulation.

Usage

```
mainp(m, q, z, y, trm, robust, prewhit, hetvar, S, s,
  R, r, doestim, dotest, docv, Tstar, rep, bigt,
  forma, formb, Verbose)
```

Arguments

m	An integer of number of breaks.
q	An integer of number of regressors.
z	A numeric matrix ($T \times q$) of regressor.
y	A numeric vector ($T \times 1$) of dependent variable.

`trm`	A numeric value of trimming parameter.
`robust`	An integer (0 or 1). If 1, use HAC standard errors.
`prewhit`	An integer (0 or 1). If 1, apply AR(1) prewhitening (only when robust=1).
`hetvar`	An integer (0 or 1). If 1, allow heteroskedastic variance across segments.
`S`	Numeric matrix defining the linear reparameterization for Form A: $\delta = S\theta + s$, where δ stacks the q coefficients for each of the $m+1$ regimes (so $\delta \in \mathbb{R}^{(m+1)q}$). Must have full column rank. If there are no restrictions, set `S = diag((m+1) * q)`.
`s`	Numeric vector of constants in $\delta = S\theta + s$, of length $(m+1)q$. If there are no restrictions, set `s = rep(0, (m+1) * q)`.
`R`	Numeric matrix defining linear restrictions for Form B, $R\delta = r$, where $\delta \in \mathbb{R}^{(m+1)q}$ stacks the q coefficients for each of the $m+1$ regimes. Each row of `R` represents one linear restriction. Must have full row rank. Do not use this form if there are no restrictions; use Form A instead.
`r`	Numeric vector of restriction constants in $R\delta = r$, with length equal to `nrow(R)`.
`doestim`	An integer (0 or 1). If 1, perform estimation.
`dotest`	An integer (0 or 1). If 1, perform sup-F test.
`docv`	An integer (0 or 1). If 1, simulate critical values.
`Tstar`	An integer of sample size for critical value simulation (default: 500).
`rep`	An integer of number of Monte Carlo replications (default: 2000).
`bigt`	An integer of sample size (if NULL, derived from length of y).
`forma`	An integer (0 or 1). If 1, use Form A restrictions.
`formb`	An integer (0 or 1). If 1, use Form B restrictions.
`Verbose`	Logical. If `TRUE`, print detailed output (default `FALSE`).

Details

The function supports two forms of linear restrictions on the break-model coefficient vector $\delta \in \mathbb{R}^{(m+1)q}$:

- **Form A**: $\delta = S\theta + s$, an affine parameterization with basis matrix S and shift s. Use `forma = 1`.
- **Form B**: $R\delta = r$, explicit linear constraints. Use `formb = 1`.

Both forms can be used simultaneously (`forma = 1` and `formb = 1`), in which case both sets of results are computed and the last one overwrites shared list entries (`estimation`, `test_statistic`, `critical_values`). If both forms are used simultaneously, consider storing results separately.

For testing and critical value simulation, the implied no-break null restrictions on the common coefficient vector β are derived automatically via `build_null_from_S()` (Form A) or `build_null_from_R()` (Form B).

Value

A list (returned invisibly) with some or all of the following components, depending on the values of `doestim`, `dotest`, `docv`, `forma`, and `formb`:

- `estimation`: Output from the estimation procedure. Produced when `doestim = 1`. If `forma = 1`, this is the output of `estco()`; if `formb = 1`, this is the output of `estco2()`. A list containing:
  - `breaks`: An integer vector of length m giving the estimated break dates.
  - `coefficients`: A numeric vector of length $(m+1)q$ giving the regime-specific coefficient estimates.
  - `vcov`: A $(m+1)q \times (m+1)q$ variance-covariance matrix of `coefficients`.
  - `ci_breaks`: An $m \times 4$ matrix of confidence intervals for the break dates. The first two columns give the 95\ (lower bound, upper bound) and the last two columns give the 90\ (lower bound, upper bound).
- `test_statistic`: The value of the restricted structural change sup-F test statistic. A scalar. Produced when `dotest = 1`. If `forma = 1`, this is the output of `pftest()`; if `formb = 1`, this is the output of `pftest2()`. The null hypothesis is no structural break subject to the implied null restrictions derived from `S, s` (Form A) or `R, r` (Form B) via `build_null_from_S()` or `build_null_from_R()`.
- `critical_values`: A numeric vector of length 4 containing simulated critical values of the sup-F test at the 90\ in that order. Produced when `docv = 1`. If `forma = 1`, simulated by `rcv()`; if `formb = 1`, simulated by `rcv2()`. Obtained by Monte Carlo simulation with `rep` replications and sample size `Tstar`, under the null of no break with the implied null restrictions.

References

Perron, P., and Qu, Z. (2006). Estimating restricted structural change models. *Journal of Econometrics*, 134(2), 373-399.

Examples

```
data(example)
y <- example$y
z <- matrix(1, length(y), 1)
m <- 3
q <- 1
trm <- 0.10
R <- matrix(c(1, 0, -1, 0,
              0, 1,  0, -1), nrow = 2, byrow = TRUE)
r <- c(0, 0)
out <- mainp(m = m, q = q, z = z, y = y, trm = trm,
             robust = 0, prewhit = 0, hetvar = 1,
             R = R, r = r,
             doestim = 1, dotest = 0, docv = 0,
             bigt = length(y), forma = 0, formb = 1, Verbose = FALSE)
out$estimation$breaks
```

mbrbp *Bootstrap restart optimization*

Description

Applies bootstrap-based restart and re-optimization after an initial run of mainp. The procedure can be applied taking the break estimates from mainp as starting values. It generates bootstrap samples, re-estimates the break dates, and plugs the new estimates back into the original sample as starting values for re-optimization. This process is repeated to reduce the chance of convergence to a spurious local optimum.

Usage

```
mbrbp(m, q, z, y, trm, B, T_init, robust, prewhit, hetvar,
  S, s, R, r, doestim, dotest, docv, cvs,
  Tstar, rep, bigt, forma, formb, verbose, resi)
```

Arguments

m	Integer. Number of breaks.
q	Integer. Number of regressors.
z	Numeric matrix ($T \times q$). Regressors.
y	Numeric vector of length T. Dependent variable.
trm	Numeric. Trimming parameter.
B	Integer. Number of bootstrap replications.
T_init	Integer vector. Initial break dates (starting values for optimization).
robust	Integer (0 or 1). If 1, use HAC standard errors.
prewhit	Integer (0 or 1). If 1, apply AR(1) prewhitening (only when robust = 1).
hetvar	Integer (0 or 1). If 1, allow heteroskedastic variance across segments.
S	Numeric matrix. Restrictions for Form A (delta = S * theta + s).
s	Numeric vector. Restrictions for Form A.
R	Numeric matrix. Restrictions for Form B (R * delta = r).
r	Numeric vector. Restrictions for Form B.
doestim	Integer (0 or 1). If 1, perform estimation given (re-optimized) break dates.
dotest	Integer (0 or 1). If 1, compute the Sup-F test statistic.
docv	Integer (0 or 1). If 1, simulate critical values.
cvs	Optional. If provided, skip simulation and use these critical values directly.
Tstar	Integer. Sample size for critical value simulation (default: 500).
rep	Integer. Number of Monte Carlo replications (default: 2000).
bigt	Integer. Sample size; if NULL, set to length(y).

forma	Integer (0 or 1). If 1, use Form A restrictions.
formb	Integer (0 or 1). If 1, use Form B restrictions.
verbose	Logical. If TRUE, print progress/output.
resi	Logical. If TRUE, use residual bootstrap restarting (brbp_residual) instead of the default observation-resampling bootstrap (brbp). Residual bootstrap preserves the time ordering of the data. Default is FALSE.

Value

(Invisibly) a list with some or all of the following components, depending on the values of doestim, dotest, docv, forma, and formb:

- estimation: Output from the estimation procedure. Produced when doestim = 1. If forma = 1, this is the output of estco(); if formb = 1, this is the output of estco2(). A list containing:
  - breaks: An integer vector of length m giving the re-optimized break date estimates.
  - coefficients: A numeric vector of length $(m + 1)q$ giving the regime-specific coefficient estimates.
  - vcov: A $(m + 1)q \times (m + 1)q$ variance-covariance matrix of coefficients.
  - ci_breaks: An $m \times 4$ matrix of confidence intervals for the break dates. The first two columns give the 95\ (lower bound, upper bound) and the last two columns give the 90\ (lower bound, upper bound).
- test_statistic: The value of the restricted structural change sup-F test statistic. A scalar. Produced when dotest = 1. If forma = 1, this is the output of pftest(); if formb = 1, this is the output of pftest2(). The null hypothesis is no structural break subject to the implied null restrictions derived from S, s (Form A) or R, r (Form B) via build_null_from_S() or build_null_from_R().
- critical_values: A numeric vector of length 4 containing critical values of the sup-F test at the 90\ that order. Produced when docv = 1. If cvs is supplied, these are returned directly without simulation; otherwise obtained by Monte Carlo simulation with rep replications and sample size Tstar, under the null of no break with the implied null restrictions. If forma = 1, simulated by rcv(); if formb = 1, simulated by rcv2().

References

Perron, P., and Qu, Z. (2006). Estimating restricted structural change models. *Journal of Econometrics*, 134(2), 373–399.

Wood, S. N. (2001). Minimizing model fitting objectives that contain spurious local minima by bootstrap restarting. *Biometrics*, 57(1), 240–244.

Examples

```
data(example)
y <- example$y
z <- matrix(1, length(y), 1)
m <- 3
q <- 1
trm <- 0.10
```

```
R <- matrix(c(1, 0, -1, 0,
              0, 1,  0, -1), nrow = 2, byrow = TRUE)
r <- c(0, 0)
init <- mainp(m = m, q = q, z = z, y = y, trm = trm,
              robust = 0, prewhit = 0, hetvar = 1,
              R = R, r = r,
              doestim = 1, dotest = 0, docv = 0,
              bigt = length(y), forma = 0, formb = 1, Verbose = FALSE)
out <- mbrbp(m = m, q = q, z = z, y = y, trm = trm, B = 3,
             T_init = init$estimation$breaks,
             robust = 0, prewhit = 0, hetvar = 1,
             R = R, r = r,
             doestim = 1, dotest = 0, docv = 0,
             bigt = length(y), forma = 0, formb = 1, verbose = FALSE)
out$estimation$breaks
```

---

pftest *Restricted Structural Change Sup-F Test (Form A)*

Description

Constructs the supremum F-test for structural change under Form A restrictions.

Usage

```
pftest(S, S0, y, z, m, q, bigt, trm, robust, prewhit, hetvar, s, s0,
  Verbose)
```

Arguments

`S`	Numeric matrix defining the linear reparameterization for Form A: $\delta = S\theta + s$, where δ stacks the q coefficients for each of the $m+1$ regimes (so $\delta \in \mathbb{R}^{(m+1)q}$). Must have full column rank. If there are no restrictions, set `S = diag((m+1) * q)`.
`S0`	Numeric matrix of restrictions under the null hypothesis. This should be generated by first running `null_model <- build_null_from_S(S, s, m, q, tol = 1e-10)` and then setting `S0 <- null_model$S0`.
`y`	A numeric vector ($T \times 1$) of dependent variable.
`z`	A numeric matrix ($T \times q$) of regressor.
`m`	An integer of number of breaks under alternative.
`q`	An integer of number of regressors.
`bigt`	An integer of sample size.
`trm`	A numeric value of trimming parameter.
`robust`	An integer (0 or 1). If 1, use HAC standard errors.

`prewhit`	An integer (0 or 1). If 1, apply AR(1) prewhitening.
`hetvar`	An integer (0 or 1). If 1, allow heteroskedastic variance.
`s`	Numeric vector of constants in $\delta = S\theta + s$, of length $(m+1)q$. If there are no restrictions, set `s = rep(0, (m+1) * q)`.
`s0`	Numeric vector of restriction constants under the null hypothesis. Obtain via `s0 <- null_model$s0`.
`Verbose`	Logical. If `TRUE`, print detailed output (default `FALSE`).

Value

The F test statistic (returned invisibly).

References

Perron, P., and Qu, Z. (2006). Estimating restricted structural change models. *Journal of Econometrics*, 134(2), 373-399.

Examples

```
data(example)
y <- example$y
z <- matrix(1, length(y), 1)
m <- 3
q <- 1
trm <- 0.10
S <- matrix(c(1, 0,
              0, 1,
              1, 0,
              0, 1), nrow = 4, byrow = TRUE)
s <- rep(0, 4)
null_model <- build_null_from_S(S, s, m, q)
pftest(S, null_model$S0, y, z, m, q, length(y), trm,
       0, 0, 1, s, null_model$s0, FALSE)
```

`pftest2` *Restricted Structural Change Sup-F Test (Form B)*

Description

Constructs the supremum F-test for structural change under Form B restrictions.

Usage

```
pftest2(R, R0, y, z, m, q, bigt, trm, robust, prewhit, hetvar, r, r0,
  Verbose)
```

Arguments

- `R` — Numeric matrix defining linear restrictions for Form B, $R\delta = r$, where $\delta \in \mathbb{R}^{(m+1)q}$ stacks the q coefficients for each of the $m + 1$ regimes. Each row of R represents one linear restriction. Must have full row rank. Do not use this form if there are no restrictions; use Form A instead.
- `R0` — Numeric matrix of restrictions under the null hypothesis, written in Form B as $R_0\delta = r_0$. This should be generated by first running `null_model <- build_null_from_R(R, r, m, q, tol = 1e-10)` and then setting `R0 <- null_model$R0`. Do not use this form when there are no restrictions; use `pftest` instead.
- `y` — A numeric vector ($T \times 1$) of dependent variable.
- `z` — A numeric matrix ($T \times q$) of regressor.
- `m` — An integer of number of breaks under alternative.
- `q` — An integer of number of regressors.
- `bigt` — An integer of sample size.
- `trm` — A numeric value of trimming parameter.
- `robust` — An integer (0 or 1). If 1, use HAC standard errors.
- `prewhit` — An integer (0 or 1). If 1, apply AR(1) prewhitening.
- `hetvar` — An integer (0 or 1). If 1, allow heteroskedastic variance.
- `r` — Numeric vector of restriction constants in $R\delta = r$, with length equal to `nrow(R)`.
- `r0` — Numeric vector of restriction constants under the null hypothesis. Obtain via `r0 <- null_model$r0`.
- `Verbose` — Logical. If `TRUE`, print detailed output.

Value

The F test statistic (returned invisibly).

References

Perron, P., and Qu, Z. (2006). Estimating restricted structural change models. *Journal of Econometrics*, 134(2), 373-399.

Examples

```
data(example)
y <- example$y
z <- matrix(1, length(y), 1)
m <- 3
q <- 1
trm <- 0.10
R <- matrix(c(1, 0, -1, 0,
              0, 1,  0, -1), nrow = 2, byrow = TRUE)
r <- c(0, 0)
null_model <- build_null_from_R(R, r, m, q)
pftest2(R, null_model$R0, y, z, m, q, length(y), trm,
        0, 0, 1, r, null_model$r0, FALSE)
```

rcv *Simulate Critical Values of Sup-F Test (Form A)*

Description

Simulates critical values for the restricted structural change sup-F test using Monte Carlo methods with Form A restrictions.

Usage

```
rcv(Tstar, rep, q, m, S, s, S0, s0, trm, Verbose)
```

Arguments

`Tstar`	An integer of sample size for simulation (default: 500).
`rep`	An integer of number of Monte Carlo replications (default: 2000).
`q`	An integer of number of regressors in a single regime.
`m`	An integer of number of breaks under the alternative.
`S`	Numeric matrix defining the linear reparameterization for Form A: $\delta = S\theta + s$, where δ stacks the q coefficients for each of the $m{+}1$ regimes (so $\delta \in \mathbb{R}^{(m+1)q}$). Must have full column rank. If there are no restrictions, set `S = diag((m+1) * q)`.
`s`	Numeric vector of constants in $\delta = S\theta + s$, of length $(m + 1)q$. If there are no restrictions, set `s = rep(0, (m+1) * q)`.
`S0`	Numeric matrix of restrictions under the null hypothesis. This should be generated by first running `null_model <- build_null_from_S(S, s, m, q, tol = 1e-10)` and then setting `S0 <- null_model$S0`.
`s0`	Numeric vector of restriction constants under the null hypothesis. Obtain via `s0 <- null_model$s0`.
`trm`	A numeric value of trimming parameter.
`Verbose`	Logical. If `TRUE`, print progress messages.

Value

A numeric vector of length 4 containing critical values at 90%, 95%, 97.5%, and 99% quantiles.

References

Perron, P., and Qu, Z. (2006). Estimating restricted structural change models. *Journal of Econometrics*, 134(2), 373-399.

Examples

```
m <- 3
q <- 1
trm <- 0.10
S <- matrix(c(1, 0,
              0, 1,
              1, 0,
              0, 1), nrow = 4, byrow = TRUE)
s <- rep(0, 4)
null_model <- build_null_from_S(S, s, m, q)
rcv(Tstar = 40, rep = 5, q = q, m = m,
    S = S, s = s, S0 = null_model$S0, s0 = null_model$s0,
    trm = trm, Verbose = FALSE)
```

`rcv2` *Simulate Critical Values of Sup-F Test (Form B)*

Description

Simulates critical values for the restricted structural change sup-F test using Monte Carlo methods with Form B restrictions.

Usage

```
rcv2(Tstar, rep, q, m, R, r, R0, r0, trm, Verbose)
```

Arguments

`Tstar`	An integer of sample size for simulation (recommended: 500).
`rep`	An integer of number of Monte Carlo replications (recommended: 2000).
`q`	An integer of number of regressors in a single regime.
`m`	An integer of number of breaks under the alternative.
`R`	Numeric matrix defining linear restrictions for Form B, $R\delta = r$, where $\delta \in \mathbb{R}^{(m+1)q}$ stacks the q coefficients for each of the $m+1$ regimes. Each row of `R` represents one linear restriction. Must have full row rank. Do not use this form if there are no restrictions; use Form A instead.
`r`	Numeric vector of restriction constants in $R\delta = r$, with length equal to `nrow(R)`.
`R0`	Numeric matrix of restrictions under the null hypothesis, written in Form B as $R_0\delta = r_0$, generated by first running `null_model <- build_null_from_R(R, r, m, q, tol = 1e-10)` and then setting `R0 <- null_model$R0`. Do not use this form when there are no restrictions; use `pftest` instead.
`r0`	Numeric vector of restriction constants under the null hypothesis. Obtain via `r0 <- null_model$r0`.
`trm`	A numeric value of trimming parameter.
`Verbose`	Logical. If `TRUE`, print progress messages.

Value

A numeric vector of length 4 containing critical values at 90%, 95%, 97.5%, and 99% quantiles.

References

Perron, P., and Qu, Z. (2006). Estimating restricted structural change models. *Journal of Econometrics*, 134(2), 373-399.

Examples

```
m <- 3
q <- 1
trm <- 0.10
R <- matrix(c(1, 0, -1, 0,
              0, 1,  0, -1), nrow = 2, byrow = TRUE)
r <- c(0, 0)
null_model <- build_null_from_R(R, r, m, q)
rcv2(Tstar = 40, rep = 5, q = q, m = m,
     R = R, r = r, R0 = null_model$R0, r0 = null_model$r0,
     trm = trm, Verbose = FALSE)
```

Index