

Package ‘sysbreak’

October 31, 2026

Title Structural Breaks in a System of Equations

Version 1.1.0

Maintainer Zhongjun Qu <qu@bu.edu>

Description Methods for detecting, estimating, and conducting inference on multiple structural breaks in a linear system of equations, such as multivariate linear regressions, seemingly unrelated regressions, vector autoregressions, and network vector autoregressions. Breaks can occur in the regression coefficients, the covariance matrix of the errors, or both, and can be restricted to a subset of the coefficients. The main function, `sysbreak()`, returns the number of breaks detected and confidence intervals for the break dates and for the regression coefficients. A separate function `ordbreak()` estimates ordered breaks, in which a break in one subset of equations occurs before a break in another. Specialized functions for vector autoregressions `varbreak()` and network vector autoregressions `netbreak()` are also provided. Reference: Qu, Z., and Perron, P. (2007). 'Estimating and Testing Structural Changes in Multivariate Regressions.' *Econometrica*, 75(2), 459-502.

License GPL (>= 3)

Depends R (>= 4.3.0)

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Encoding UTF-8

NeedsCompilation no

RoxygenNote 7.3.2

Imports graphics, grDevices, MASS, stats, utils

Config/testthat/edition 3

LazyData true

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

Repository CRAN

Date/Publication 2026-10-31 11:40:02 UTC

Contents

brtype	2
data1	3
data2	4
dynobreak	4
dyspill	5
housing	7
inflation	8
netbreak	9
netsim	12
ordbreak	13
ordcheck	16
plot.sysbreak	17
summary.sysbreak	18
sysbreak	19
sysbreak-methods	22
varbreak	23
Index	27

brtype	<i>Methods to Examine What Changes at Each Break Date</i>
--------	---

Description

For each estimated break date, tests whether the volatilities, the correlations and the regression coefficients differ between the two regimes on either side of it.

Usage

brtype(fit)

Arguments

fit A result from [sysbreak](#), [varbreak](#) or [netbreak](#) with at least one break.

Details

The volatility statistic is a two-sample Hotelling (1931) statistic on the squared residuals. The correlation statistic is the Jennrich (1970) test for equal correlation matrices, which assumes normal errors. The coefficient statistic is a Wald test for equal coefficients; it covers only the coefficients allowed to change, and is NA when no coefficient is allowed to change. Each test compares the two regimes adjacent to a break and treats the break dates as known.

The coefficient test holds the error covariance constant across regimes, even when `fit` allows it to break (`brv = TRUE`). Estimating the covariance separately in each regime would make the test oversized in samples of typical length; the constant covariance tends to make it conservative when the volatility does change.

Value

A data frame with one row per break: breaks, the break date as a row of the data; the statistics volatility, correlation and coefficients; and their p-values volatility_p, correlation_p and coefficients_p. The degrees of freedom are n , $n(n-1)/2$ and the number of coefficients allowed to change.

References

Hotelling, H. (1931). The generalization of Student's ratio. *The Annals of Mathematical Statistics*, 2(3), 360-378.

Jennrich, R. I. (1970). An asymptotic chi-square test for the equality of two correlation matrices. *Journal of the American Statistical Association*, 65, 904-912.

Examples

```
set.seed(11)
bigt <- 120
x <- rnorm(bigt)
y1 <- 1 + 0.4 * x + rnorm(bigt, sd = 0.3)
y2 <- -0.5 + 0.2 * x + rnorm(bigt, sd = 0.3)
y1[61:120] <- y1[61:120] + 1.2 - 0.6 * x[61:120]
dat <- data.frame(y1 = y1, y2 = y2, x = x)

fit <- sysbreak(cbind(y1, y2) ~ x, data = dat, m = 2, trim = 0.2)
brtype(fit)
```

data1

GAUSS/MATLAB Example Data 1

Description

Example data used by the original GAUSS/MATLAB mbreak2 script. In this package the corresponding procedure is [sysbreak](#).

Usage

data1

Format

A numeric matrix with 150 rows and 5 columns. Columns 1 and 2 are dependent variables, and columns 3 through 5 are regressors.

Source

Qu, Z., and Perron, P. (2007). Estimating and testing structural changes in multivariate regressions. *Econometrica*, 75(2), 459-502.

data2	<i>GAUSS/MATLAB Example Data 2</i>
-------	------------------------------------

Description

Example data used by the original GAUSS/MATLAB obreak2 script for the ordered-break procedure.

Usage

data2

Format

A numeric matrix with 100 rows and 5 columns. Columns 1 and 2 are dependent variables, and columns 3 through 5 are regressors.

Source

Qu, Z., and Perron, P. (2007). Estimating and testing structural changes in multivariate regressions. *Econometrica*, 75(2), 459-502.

dynobreak	<i>Diebold-Yilmaz Spillover Measures Without Breaks</i>
-----------	---

Description

Fits a VAR by OLS on the whole sample and computes the generalized spillover measures of `dyspill()`, for comparison with the measures of a model with breaks.

Usage

`dynobreak(y, lags, h, intercept = TRUE)`

Arguments

<code>y</code>	Numeric matrix, one column per variable.
<code>lags</code>	Positive integer number of lags.
<code>h</code>	Positive integer forecast horizon.
<code>intercept</code>	Logical; include an intercept.

Value

A list with the row-normalized generalized fevd, the total spillover index `total`, and the directional vectors `from`, `to` and `net`. The covariance divides by the number of observations.

Examples

```

set.seed(1)
bigt <- 120
a <- matrix(c(0.2, 0.4, 0.4, 0.2), 2)
y <- matrix(0, bigt, 2)
colnames(y) <- c("y1", "y2")
for (tt in 2:bigt) {
  y[tt, ] <- a %*% y[tt - 1, ] + rnorm(2, sd = 0.3)
}

sp <- dynobreak(y, lags = 1, h = 10)
sp$total
sp$net

```

dyspill

*Diebold-Yilmaz Generalized Spillover Measures***Description**

Computes horizon-specific spillover measures from a fitted structural-break VAR model. The forecast-error variance decomposition uses generalized shocks and is therefore invariant to variable ordering.

Usage

```
dyspill(fit, h = 10, boot = 0, seed = NULL)
```

Arguments

fit	A fitted object returned by <code>varbreak()</code> or <code>netbreak()</code> .
h	Positive integer for forecast horizon.
boot	Number of bootstrap replications for standard errors and confidence intervals; 0 for none. Available for <code>varbreak()</code> and <code>netbreak()</code> fits.
seed	Optional seed for the bootstrapping. The seed of the calling session is restored on exit.

Details

For a regime-specific VAR,

$$y_t = c_r + \sum_{\ell=1}^p A_{\ell,r} y_{t-\ell} + u_t, \quad E(u_t u_t') = \Sigma_r,$$

let $\Psi_{0,r} = I$ and

$$\Psi_{s,r} = \sum_{\ell=1}^{\min(p,s)} A_{\ell,r} \Psi_{s-\ell,r}.$$

The generalized variance share from shock j to variable i is

$$\theta_{ij}^g(H) = \frac{\sigma_{jj}^{-1} \sum_{s=0}^{H-1} (e_i' \Psi_s \Sigma e_j)^2}{\sum_{s=0}^{H-1} e_i' \Psi_s \Sigma \Psi_s' e_i}.$$

Rows are normalized to sum to one before spillover statistics are formed. For variable i , from is the contribution received from all other shocks, to is the contribution transmitted to all other variables, and net = to - from; a positive net value identifies a net transmitter. The total spillover index is the average off-diagonal contribution, expressed in percent.

The benchmark puts every lag matrix to zero and keeps Σ_r . The measures then depend on the error correlations only; comparing them with the fitted ones shows how much of the spillover comes from the dynamics. With two variables the benchmark total is $100\rho^2/(1 + \rho^2)$.

The bootstrap is a recursive residual bootstrap within each regime: the residuals of the regime are resampled, the series is rebuilt from the fitted VAR of the regime, the model is refitted at the same break dates in the same way as fit, and the measures are recomputed. The break dates are treated as known. The intervals are percentile intervals, since the draws are skewed in small samples. Draws in which a refitted VAR is not stable are dropped.

Value

An object of class "dyspill" containing:

horizon The forecast horizon h .

model The source model type.

variables Names of the endogenous variables.

break_grid A data frame with one row per regime: the regime number; its first and last observation, start and end, counted in the estimation sample (which omits the first lags observations); and the coefficient and covariance regimes it belongs to, coef_regime and cov_regime.

regimes A list with one entry per regime. Each entry contains:

- input_rows: the first and last row of the data in the regime.
- spectral_radius: the largest modulus of the eigenvalues of the companion matrix of the VAR; the VAR is stable when it is below one.
- fevd: the generalized forecast-error variance decomposition, with rows normalized to sum to one.
- total: the total spillover index.
- from, to and net: the directional spillovers of each variable.
- net_pairwise: the matrix of net pairwise spillovers.
- benchmark_total and benchmark_net: total and net recomputed with all lag coefficients set to zero, so that they reflect the error correlations only (see Details).
- With boot > 0: the bootstrap standard errors total_se and net_se, and the percentile intervals total_ci and net_ci.

boot Returned only when boot > 0. A list with:

- replications: the number of bootstrap replications.
- level: the coverage probability of the intervals, taken from ci_level of the fit.

- **total**: the bootstrap draws of the total index, a matrix with one row per replication and one column per regime.
- **net**: the bootstrap draws of the net spillovers, an array of replications by variables by regimes.
- **differences**: a data frame with one row per pair of regimes, giving the difference in the total index (the later regime minus the earlier one), its bootstrap standard error se, and the percentile interval lower and upper.
- **dropped**: the number of draws dropped in each regime because the refitted VAR was not stable.

References

Diebold, F. X., and Yilmaz, K. (2012). Better to give than to receive: Predictive directional measurement of volatility spillovers. *International Journal of Forecasting*, 28(1), 57–66.

Examples

```
set.seed(1)
bigt <- 120
y <- matrix(0, bigt, 2)
colnames(y) <- c("y1", "y2")
for (tt in 2:bigt) {
  a <- if (tt > 60) matrix(c(0.2, 0.6, 0.6, 0.2), 2) else diag(0.2, 2)
  y[tt, ] <- a %*% y[tt - 1, ] + rnorm(2, sd = 0.3)
}

fit <- varbreak(y, lags = 1, m = 1, trim = 0.2)
dyspill(fit, h = 10, boot = 99, seed = 1)
```

housing

U.S. House Prices and the Macroeconomy

Description

Quarterly data for the house price application of Ahamada and Diaz Sanchez (2013), 1959:Q4 to 2009:Q3.

Usage

housing

Format

A data frame with 200 rows and 6 columns:

date Quarter, such as "1960:Q1".

dGDP Log difference of real GDP.

- dC** Log difference of real personal consumption expenditures.
- dHP** Log difference of the real home price index.
- FFR** Federal funds rate, in percent, first month of the quarter.
- dI** Change in FFR.

Source

FRED series GDPC1, PCEC, CPIAUCSL and FEDFUNDS, and the Shiller nominal home price index, seasonally adjusted by log-STL. Consumption and home prices are deflated by CPIAUCSL.

References

Ahamada, I., and Diaz Sanchez, J. L. (2013). A retrospective analysis of the house price macro-relationship in the United States. *International Journal of Central Banking*, 9(4), 153-174.

inflation	<i>International Inflation</i>
-----------	--------------------------------

Description

Monthly consumer price inflation for the G7 countries and the euro area, 1973:04 to 2007:12, from Bataa, Osborn, Sensier and van Dijk (2013).

Usage

inflation

Format

A data frame with 417 rows and 9 columns: date, such as "1973:04", followed by inflation for Canada, EuroArea, France, Germany, Italy, Japan, UK and US. Inflation is $100 \times \Delta \log$ of the consumer price index, adjusted for outliers and seasonality by the authors.

Source

The replication files of Bataa et al. (2013).

References

Bataa, E., Osborn, D. R., Sensier, M., and van Dijk, D. (2013). Structural breaks in the international dynamics of inflation. *Review of Economics and Statistics*, 95(2), 646-659.

netbreak

*Structural Breaks in a network VAR***Description**

Estimates structural breaks in a VAR whose coefficients are restricted by a fixed network matrix. With one lag, the model is

$$y_t = \mu_j + (\alpha_{1,j}W + \beta_{1,j}I)y_{t-1} + u_t,$$

for observations in regime j . With multiple lags, the model is

$$y_t = \mu_j + \sum_{\ell=1}^p (\alpha_{\ell,j}W + \beta_{\ell,j}I)y_{t-\ell} + u_t.$$

Usage

```
netbreak(y, W, lags, m, intercept = TRUE,
  coef_breaks = "all", row_normalize = FALSE,
  test_level = 0.05, ci_level = 0.95,
  trim = 0.15, verbose = FALSE, brv = FALSE,
  robust = c("none", "hc", "hac"), hetq = FALSE, prewhit = FALSE,
  restart = c("none", "brbp", "brbp_residual"),
  restart_B = 10, restart_verbose = FALSE, h = NULL)
```

Arguments

y	Numeric matrix/data frame with one column per network node.
W	Numeric square network matrix. Row i gives the weights used to form the network lag for equation i .
lags	Positive integer, number of lags.
m	The maximum number of breaks allowed.
intercept	Logical. If TRUE, include equation-specific intercepts μ_i .
coef_breaks	Parameters allowed to change across regimes. The parameters are one intercept per node, μ_i ; the network coefficients α_ℓ , named network_lag1, network_lag2, and so on; and the own-lag coefficients β_ℓ , named own_lag1, own_lag2, and so on. The default "all" lets every parameter change. "none" or NULL holds them all constant, which requires brv = TRUE so that the disturbance covariance matrix can change; otherwise the code will return an error message. For anything in between, supply a character vector of parameter names or of these groups: <ul style="list-style-type: none"> • "intercepts", or "mu": all μ_i. • "network": all α_ℓ. • "own": all β_ℓ. • "dynamic": every lag coefficient, network and own.

	<ul style="list-style-type: none"> • "lag1", "lag2", ...: α_ℓ and β_ℓ at that lag. • "alpha1", "beta1", ...: α_ℓ or β_ℓ on its own. <p>Entries are case-insensitive and may be combined, as in <code>c("intercepts", "network_lag1")</code>.</p>
row_normalize	Logical. If TRUE, row-normalize W ; rows with zero row sums are left as zero rows.
test_level	Significance level used for selecting the number of breaks. One of 0.10, 0.05, 0.025 or 0.01. WDmax decides whether there is any break; if it is not significant the procedure stops and no break is reported. Otherwise the sequential tests of l against $l + 1$ breaks are applied from one break upwards, stopping at the first insignificant test or at m .
ci_level	Coverage probability of the reported confidence intervals for the break dates. Either 0.95 or 0.90.
trim	Trimming parameter. The minimum segment length is $h = \text{round}(\text{trim} * \text{nrow}(y))$. We recommend $h \geq 3(n + k)$, where n is the number of equations and k the number of regressors per equation, and a larger value for larger systems. A warning is issued if $h < 3(n + k)$.
verbose	Logical. If TRUE, print additional progress information from the testing and estimation.
brv	Logical. If TRUE, allow the disturbance covariance matrix to change across regimes.
robust	Inference for the coefficient-break tests, the coefficient standard errors and the break-date confidence intervals. "none" gives Gaussian quasi-likelihood-ratio tests, valid when the errors are serially uncorrelated and conditionally homoskedastic. "hc" gives the robust SupF statistic of Qu and Perron (2007, eq. 20) with a heteroskedasticity-consistent covariance estimate without autocovariance terms, valid when the errors are a martingale difference sequence. "hac" uses instead the quadratic spectral kernel with the automatic bandwidth of Andrews (1991), allowing for serial correlation. With enough lags the VAR errors are serially uncorrelated, and "hc" suffices for the tests and standard errors; with <code>brv = TRUE</code> and conditionally heteroskedastic errors of the ARCH type, the confidence intervals require "hac".
hetq	Logical. If TRUE, allow regressor moments to vary across regimes when computing confidence intervals for break dates.
prewhit	Logical. If TRUE, prewhiten the scores by a VAR(1) before the HAC estimation (Andrews and Monahan, 1992). Used only when <code>robust = "hac"</code> .
restart	A bootstrap restarting procedure to check and improve convergence. It applies only when some coefficients are held constant; otherwise it is ignored with a warning. Set to "none" to turn it off, "brbp" for resampling bootstrap restarts, or "brbp_residual" for residual-based restarts.
restart_B	Integer, number of restart values used when <code>restart != "none"</code> .
restart_verbose	Logical. If TRUE, print restart progress.
h	Optional positive integer for forecast horizon. If supplied, attach Diebold–Yilmaz generalized spillover measures in <code>spillover</code> .

Details

The interpretation of `testing$statistic` depends on `robust`.

- `robust = "none"` gives the SupLR statistic. The code can test the parameters specified by `coef_breaks`, the disturbance covariance when `brv = TRUE`, or both.
- `robust = "hc"` or `"hac"` gives the robust SupF statistic, for the parameters only, because the covariance-break critical values are case dependent under heteroskedasticity or serial correlation; it assumes a constant error covariance. With `brv = TRUE`, the estimates, standard errors and confidence intervals account for the covariance breaks; under `"hc"` the intervals further require errors without ARCH-type dependence.

Value

An object of class `c("netbreak", "sysbreak")` containing:

`spec` A summary of the setup, including the sample size `bigt`, the number of equations `n`, the total number of NVAR parameters per regime `q`, the arguments `m`, `lags`, `intercept`, `row_normalize`, `trim`, `brv`, `robust`, `hetq`, `prewhit`, `test_level` and `ci_level`, the network matrix `W`, the equation and parameter names, the retained input rows `y_rows` after discarding the lags, and a record of the `coef_breaks` selector.

`testing` The tests, reported at `test_level`: statistic, testing no break against k breaks for every $k = 1, \dots, m$, with its `critical_values`; `statistic_type`, "SupLR" or "SupF", and NA when a requested robust test is unavailable; `wdmax` and `wdmax_critical_values`; the sequential tests of l against $l + 1$ breaks, each containing `l`, a statistic and its `critical_values`; and the number of `changing_parameters`, which includes the changing covariance parameters when `robust = "none"`.

`br_selected` The outcome of the break-number determination at `test_level`: the selected number of breaks, the WDmax statistic with its critical value and significance flag, and the sequential steps taken. `breaks` is 0 when WDmax is not significant, and NA when the tests are unavailable for the requested specification.

`estimation` The estimates related to the selected number of breaks: estimated breaks; their confidence intervals `ci_breaks`; coefficients by regimes and their confidence intervals; disturbance covariance matrices. This component is NULL when no break is detected, in which case only `testing` and `br_selected` are informative.

`spillover` When `h` is supplied and at least one break is selected, regime-specific Diebold–Yilmaz generalized spillover measures returned by [dyspill](#).

`y` The responses used in estimation, rows `spec$y_rows` of `y`.

Note that `m` is the *maximum* number of breaks allowed. Testing output is reported for every value from 1 through `m`, while estimation corresponds to the number of breaks selected by the procedure under `test_level`. Parameters excluded by `coef_breaks` are repeated across regime blocks. The optional spillover component is omitted when `h = NULL`. The fit has `print()`, [summary\(\)](#), [plot\(\)](#) and the accessors in [sysbreak-methods](#).

References

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

Andrews, D. W. K., and Monahan, J. C. (1992). An improved heteroskedasticity and autocorrelation consistent covariance matrix estimator. *Econometrica*, 60(4), 953–966.

Zhu, X., Pan, R., Li, G., Liu, Y., and Wang, H. (2017). Network vector autoregression. *The Annals of Statistics*, 45(3), 1096–1123.

Qu, Z., and Perron, P. (2007). Estimating and testing structural changes in multivariate regressions. *Econometrica*, 75(2), 459–502.

Examples

```
set.seed(5)
bigt <- 80
W <- matrix(c(0, 1, 1, 0), 2, 2, byrow = TRUE)
y <- matrix(0, bigt, 2)
colnames(y) <- c("y1", "y2")
for (tt in 2:bigt) {
  alpha <- if (tt <= 40) 0.05 else 0.75
  beta <- 0.1
  y[tt, ] <- alpha * as.numeric(W %*% y[tt - 1, ]) +
    beta * y[tt - 1, ] + rnorm(2, sd = 0.2)
}

fit <- netbreak(
  y = y, W = W, lags = 1, m = 1,
  coef_breaks = c("network", "own"),
  trim = 0.2
)
fit$estimation$breaks
```

netsim

Simulated Network VAR

Description

One draw from Scenario 1 of Han and Lee (2024): a network VAR for 20 nodes over 300 periods, with breaks after periods 100 and 200.

Usage

netsim

Format

A list with two components:

y A 300 by 20 matrix of observations, with columns node1 to node20.

W The 20 by 20 row-normalized network matrix. Entry (i, l) is $1/n_i$ if node i follows node l and 0 otherwise, where n_i is the number of nodes that i follows.

Details

In regime j ,

$$y_{it} = V_i' \gamma_j + \sum_{\ell} \alpha_{\ell,j} (W y_{t-\ell})_i + \sum_{\ell} \beta_{\ell,j} y_{i,t-\ell} + e_{it}, \quad e_{it} \sim N(0, 0.1^2).$$

Regime 1 has one lag, with $\alpha = -0.10$ and $\beta = 0.20$. Regimes 2 and 3 have two lags, with $\alpha = (0.20, -0.22)$, $\beta = (-0.12, 0.40)$ and $\alpha = (-0.12, 0.10)$, $\beta = (0.25, -0.40)$. The covariates V_i are fixed, so $V_i' \gamma_j$ is a node intercept that changes across regimes. The in-degrees of the network follow a power law, as in Zhu et al. (2017).

Source

Simulated with `set.seed(1)` following Han and Lee (2024).

References

- Han, Y., and Lee, T. C. M. (2024). Structural break detection in non-stationary network vector autoregression models. *IEEE Transactions on Network Science and Engineering*, 11(5), 4134-4145.
- Zhu, X., Pan, R., Li, G., Liu, Y., and Wang, H. (2017). Network vector autoregression. *The Annals of Statistics*, 45(3), 1096-1123.

ordbreak

Ordered Structural Breaks in a System of Equations

Description

Estimation of a multiple-equation system in which the coefficients of one group of equations change before those of another group. The two break dates are estimated jointly and are not restricted to coincide.

Usage

```
ordbreak(set1, set2, data,
  coef_breaks1 = "all", coef_breaks2 = "all",
  trim = 0.2, hetq = TRUE, max_break_gap = NULL,
  test_level = 0.05, ci_level = 0.95,
  confidence_intervals = FALSE,
  ci_grid_radius = 20, ci_replications = 5000, ci_step = 1000,
  ci_grid_step = 0.02, ci_seed = NULL, ci_verbose = FALSE,
  tol = 1e-8, max_iter = 1000)
```

Arguments

set1	The equations whose coefficients change at the first date. A formula such as $y1 \sim x1 + x2$ for a single equation, or <code>list(y1 ~ x1 + x2, y2 ~ x2 + x3)</code> for several. The number of regressors may differ across equations. An intercept is added to each equation unless the formula indicates otherwise.
set2	The equations whose coefficients change at the second date, written in the same way as set1. Each equation of the system belongs to exactly one of the two sets.
data	Data frame used to evaluate the formulas.
coef_breaks1	An expression for which coefficients of set1 are allowed to change at the first date. The default "all" allows all of them to change. Otherwise supply a list of formulas, one per equation of set1, where the left-hand side should specify the equation and the right-hand side selects the coefficients of that equation that may change: <ul style="list-style-type: none"> • $y1 \sim x1$: the intercept and the coefficient on $x1$ change. • $y1 \sim x1 - 1$: the coefficient on $x1$ only. • $y1 \sim 0$: no coefficient of that equation changes. Coefficients left out are held constant across both dates. At least one coefficient of set1 must be allowed to change.
coef_breaks2	The same, for set2 and the second date.
trim	Trimming parameter, which must lie in $(0, 0.5)$. The minimum segment length is <code>floor(trim * nrow(data))</code> .
hetq	Logical. If TRUE, allow regressor moments to vary across regimes when computing confidence intervals for break dates.
max_break_gap	Truncation used when computing the test statistic. If NULL, <code>floor(sqrt(T))</code> is used, as in the original GAUSS program.
test_level	Significance level at which the critical value is reported. One of 0.10, 0.05, 0.025 or 0.01.
ci_level	Coverage probability of the reported confidence intervals for the break dates. Either 0.95 or 0.90.
confidence_intervals	Logical. If TRUE, simulate confidence intervals for the two break dates. This is the expensive part of the procedure and is therefore off by default.
ci_grid_radius	Half-width of the grid on which the limiting distribution is simulated.
ci_replications	Number of simulation replications.
ci_step	Number of steps used to discretize each Brownian motion.
ci_grid_step	Spacing of the simulation grid.
ci_seed	Optional seed for the simulation. The seed of the calling session is restored on exit.
ci_verbose	Logical. If TRUE, print simulation progress.
tol	Convergence tolerance of the feasible GLS iterations.
max_iter	Maximum number of feasible GLS iterations.

Details

The coefficients are stacked equation by equation in the order the formulas are given, so those of set1 occupy the leading $q1$ positions and those of set2 the remaining $q2$. `ordcheck` reports this layout without estimating anything, which is worth doing before a long run.

The test is that of Qu and Perron (2007, Section 6.4) for locally ordered breaks. The test for globally ordered breaks in their Section 6.5, whose limiting distribution differs, is not implemented.

Value

An object of class "ordbreak" containing:

`spec` The setup: the sample size `bigt`, the number of equations `n`, the minimum segment length `h`, the number of coefficients in each set `q1` and `q2`, the number of equations in the first set `n1`, the names of the coefficients allowed to change at each date, `changing1` and `changing2`, the arguments `trim`, `hetq` and `trun`, the input formulas split into `set1` and `set2`, the equation and coefficient names, the retained data rows `y_rows` after discarding missing values, and the selection matrix `S`.

`testing` The SupLR statistic for ordered breaks, `sup_lr_ordered`, its `critical_values` at `test_level`, the pair `truncated_breaks` the statistic is computed at, and the log-likelihoods `H1like` and `H0like` it is built from. The critical value is indexed by the number of coefficients allowed to change, not by the total number of coefficients.

`estimation` The estimated breaks, given as rows of data; the coefficients `beta1` and `beta2` of the two regimes; the residuals of the model at the estimated breaks, one column per equation; and the coefficient shifts `delta1` and `delta2` occurring at the two dates; and `ci_breaks`, a two by two matrix when `confidence_intervals = TRUE` and `NULL` otherwise, also given as rows of data. Its rows are the two break dates and its columns the lower and upper bounds at `ci_level`.

Note that the ordering is imposed rather than tested. The equations in `set1` are taken to break first, so exchanging the two sets specifies a different model. Both specifications are tested against the same no-break null, and their values of `sup_lr_ordered` can be compared.

References

Qu, Z., and Perron, P. (2007). Estimating and testing structural changes in multivariate regressions. *Econometrica*, 75(2), 459–502.

Examples

```
set.seed(13)
bigt <- 60
x1 <- rnorm(bigt)
x2 <- rnorm(bigt)
y1 <- 1 + 0.5 * x1 + rnorm(bigt, sd = 0.3)
y2 <- 1 + 0.5 * x2 + rnorm(bigt, sd = 0.3)
y1[21:bigt] <- y1[21:bigt] + 1.5 * x1[21:bigt]
y2[36:bigt] <- y2[36:bigt] - 1.5 * x2[36:bigt]
dat <- data.frame(y1 = y1, y2 = y2, x1 = x1, x2 = x2)
```

```
fit <- ordbreak(y1 ~ x1, y2 ~ x2, data = dat, trim = 0.2)
fit$estimation$breaks
```

ordcheck

Coefficient Layout of an Ordered-Break Specification

Description

Reports the coefficients of an ordered-break model, which set of equations they belongs to, and which of them are allowed to change, without estimating anything.

Usage

```
ordcheck(set1, set2, data, coef_breaks1 = "all", coef_breaks2 = "all")
```

Arguments

set1	The equations whose coefficients change at the first date, written as in ordbreak .
set2	The equations whose coefficients change at the second date.
data	Data frame used to evaluate the formulas.
coef_breaks1	Which coefficients of set1 are allowed to change at the first date. See ordbreak .
coef_breaks2	The same, for set2 and the second date.

Value

The layout is printed: one row per coefficient, giving its position in the coefficient vector, its name, the set it belongs to, whether it is allowed to change, and the date at which it changes.

Examples

```
set.seed(13)
dat <- data.frame(y1 = rnorm(40), y2 = rnorm(40),
                  x1 = rnorm(40), x2 = rnorm(40))

ordcheck(y1 ~ x1 + x2, y2 ~ x1 + x2, data = dat,
         coef_breaks1 = list(y1 ~ x1))
```

plot.sysbreak	<i>Plot a Structural-Break Fit</i>
---------------	------------------------------------

Description

Plots each series with the estimated break dates (dashed lines), their confidence intervals (shaded) and, optionally, the fitted values of the regime-specific model.

Usage

```
## S3 method for class 'sysbreak'
plot(x, which = NULL, dates = NULL, fitted = TRUE, ...)
```

Arguments

x	A fit returned by sysbreak , varbreak or netbreak .
which	Series to plot, as indices or names. By default all series, or the first four when there are more.
dates	Optional vector with one entry per row of the data passed to the fitting function, including the initial lags of a VAR and any rows dropped for missing values; the observation in row <i>r</i> is placed at dates[<i>r</i>]. Numeric values or dates are used as the horizontal axis; character values, such as "1983:Q1", are used as labels for the axis.
fitted	Logical. If TRUE, add the fitted values.
...	Graphical parameters. type, col, lwd, lty and pch apply to the plotted series; main and ylab may be vectors with one entry per panel; the others, such as xlab, ylim or las, are passed to plot() when each panel is set up.

Value

x, invisibly.

See Also

[sysbreak-methods](#), [summary.sysbreak](#)

Examples

```
set.seed(11)
bigt <- 40
x <- rnorm(bigt)
y1 <- 1 + 0.4 * x + rnorm(bigt, sd = 0.3)
y2 <- -0.5 + 0.2 * x + rnorm(bigt, sd = 0.3)
y1[21:40] <- y1[21:40] + 1.2 - 0.6 * x[21:40]
y2[21:40] <- y2[21:40] - 0.8 + 0.4 * x[21:40]
dat <- data.frame(y1 = y1, y2 = y2, x = x)
fit <- sysbreak(cbind(y1, y2) ~ x, dat, m = 1, trim = 0.3)
```

```
plot(fit)
```

summary.sysbreak	<i>Summarize a Structural-Break Fit</i>
------------------	---

Description

A report of a fit returned by [sysbreak](#), [varbreak](#) or [netbreak](#): the break number, the break dates with their confidence intervals, and a coefficient table for each regime. `print()` of the fit gives the full output.

Usage

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

## S3 method for class 'summary.sysbreak'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

object	A fit returned by <code>sysbreak()</code> , <code>varbreak()</code> or <code>netbreak()</code> .
...	Not used.
x	An object returned by <code>summary()</code> .
digits	Number of significant digits in the coefficient tables.

Value

An object of class "summary.sysbreak", a list with the model type, the numbers of equations `n` and observations `bigt`, the tests for the break-number decision, the number of breaks selected, and, when at least one break is selected, the break dates with their confidence intervals (`breaks`), the rows of the data in each regime (`regimes`) and the coefficient tables (`coefficients`): estimate, standard error, z value and two-sided p -value, one table per regime, treating the break dates as known.

See Also

[sysbreak-methods](#), [plot.sysbreak](#)

Examples

```
set.seed(11)
bigt <- 40
x <- rnorm(bigt)
y1 <- 1 + 0.4 * x + rnorm(bigt, sd = 0.3)
y2 <- -0.5 + 0.2 * x + rnorm(bigt, sd = 0.3)
```

```

y1[21:40] <- y1[21:40] + 1.2 - 0.6 * x[21:40]
y2[21:40] <- y2[21:40] - 0.8 + 0.4 * x[21:40]
dat <- data.frame(y1 = y1, y2 = y2, x = x)
fit <- sysbreak(cbind(y1, y2) ~ x, dat, m = 1, trim = 0.3)
summary(fit)

```

sysbreak

Structural Breaks in a System of Equations

Description

Estimation of a multiple-equation system affected by structural breaks in the regression coefficients and/or the variance covariance matrix of the residuals.

Usage

```

sysbreak(model, data = NULL, m, coef_breaks = "all",
  test_level = 0.05, ci_level = 0.95,
  trim = 0.15, verbose = FALSE, brv = FALSE,
  robust = c("none", "hc", "hac"), hetq = FALSE, prewhit = FALSE,
  restart = c("none", "brbp", "brbp_residual"),
  restart_B = 10, restart_verbose = FALSE)

```

Arguments

- | | |
|-------------|--|
| model | A formula such as <code>cbind(y1, y2) ~ x1 + x2</code> for a system with common regressors across equations, or <code>list(y1 ~ x1 + x2, y2 ~ x2 + x3)</code> for a system allowing different regressors across equations. The number of regressors may differ across equations, e.g., <code>list(y1 ~ x1 + x2, y2 ~ x3)</code> . An intercept is added to each equation unless the formula indicates otherwise. |
| data | Data frame used to evaluate model. |
| m | Integer, the maximum number of breaks allowed when testing for breaks. |
| coef_breaks | <p>An expression for which regression coefficients are allowed to change across regimes. The default "all" allows all regression coefficients to change. "none" or NULL restrictions all coefficients to be constant. \ This case requires setting <code>brv = TRUE</code>, so that the disturbance covariance matrix\ can change; otherwise the code will return an error message. For in between, supply a list of formulas, one per equation, where left-hand side should specify the equation and the right-hand side selects the coefficients of that equation that may change:</p> <ul style="list-style-type: none"> • <code>y1 ~ x1, y2 ~ x2</code>: the coefficient on <code>x1</code> in the first equation and the coefficient on <code>x2</code> in the second equation can change. • <code>y1 ~ 1, y2 ~ x2 + x4</code>: the intercept in the first equation and the two slope coefficients in the second equation. • <code>y1 ~ 1 + x1, y2 ~ 0</code>: the intercept and <code>x1</code> in the first equation, and no coefficient in the second equation. |

test_level	Significance level used to select the number of breaks. One of 0.10, 0.05, 0.025 or 0.01. The double-maximum tests decide whether there is any break; if WDmax is not significant the procedure stops and no break is reported. Otherwise the sequential tests of l against $l + 1$ breaks are applied from one break upwards, stopping at the first insignificant test or at m .
ci_level	Coverage probability of the reported confidence intervals for the break dates. Either 0.95 or 0.90.
trim	Trimming parameter. The minimum segment length is $h = \text{round}(\text{trim} * \text{nrow}(y))$. We recommend $h \geq 3(n + k)$, where n is the number of equations and k the number of regressors per equation, and a larger value for larger systems. A warning is issued if $h < 3(n + k)$.
verbose	Logical. If TRUE, print additional information from the estimation.
brv	Logical. If TRUE, allow the covariance matrix of the residuals to change across regimes.
robust	Inference for the coefficient-break tests, the coefficient standard errors and the break-date confidence intervals. "none" gives Gaussian quasi-likelihood-ratio tests, valid when the errors are serially uncorrelated and conditionally homoskedastic. "hc" gives the robust SupF statistic of Qu and Perron (2007, eq. 20) with a heteroskedasticity-consistent covariance estimate without autocovariance terms, valid when the errors are a martingale difference sequence. "hac" uses instead the quadratic spectral kernel with the automatic bandwidth of Andrews (1991), allowing for serial correlation.
hetq	Logical. If TRUE, allow regressor moments to vary across regimes when computing confidence intervals for break dates.
prewhit	Logical. If TRUE, prewhiten the scores by a VAR(1) before the HAC estimation (Andrews and Monahan, 1992). Used only when robust = "hac".
restart	A bootstrap restarting procedure to check and improve convergence. It applies only when some coefficients are held constant; otherwise it is ignored with a warning. Set to "none" to turn it off, "brbp" for resampling bootstrap restarts, or "brbp_residual" for residual-based restarts.
restart_B	Number of restarting values used when restart != "none"; 10 is recommended.
restart_verbose	Logical. If TRUE, print restart progress.

Details

The interpretation of `testing$statistic` depends on `robust`.

- `robust = "none"` gives the SupLR statistic. The code can test the coefficients specified by `coef_breaks`, the disturbance covariance when `brv = TRUE`, or both.
- `robust = "hc"` or `"hac"` gives the robust SupF statistic, for the coefficients only, because the covariance-break critical values are case dependent under heteroskedasticity or serial correlation; it assumes a constant error covariance. With `brv = TRUE`, the estimates, standard errors and confidence intervals account for the covariance breaks; under `"hc"` the intervals further require errors without ARCH-type dependence.

Value

An object of class "sysbreak" containing:

spec A summary of the setup, including the sample size *bigt*, the number of equations *n*, the total number of regression coefficients per regime *q*, the arguments *m*, *trim*, *brv*, *robust*, *hetq*, *prewhit*, *test_level* and *ci_level*, the input formula or formulas, the equation and coefficient names, the retained data rows *y_rows* after discarding missing values, and a record of the *coef_breaks* selector.

testing The tests, reported at *test_level*: *statistic*, testing no break against *k* breaks for every $k = 1, \dots, m$, with its *critical_values*; *statistic_type*, "SupLR" or "SupF", and NA when a requested robust test is unavailable; *wdmax* and *wdmax_critical_values*; the sequential tests of *l* against *l + 1* breaks, each containing *l*, a statistic and its *critical_values*; and the number of *changing_parameters*, which includes the changing covariance parameters when *robust* = "none".

br_selected The outcome of the break-number determination at *test_level*: the selected number of breaks, the WDmax statistic with its critical value and significance flag, and the sequential steps taken. *breaks* is 0 when WDmax is not significant, and NA when the tests are unavailable for the requested specification.

estimation The estimates related to the selected number of breaks: estimated breaks; their confidence intervals *ci_breaks*; coefficients by regimes and their confidence intervals; disturbance covariance matrices. This component is NULL when no break is detected, in which case only testing and *br_selected* are informative.

y The responses used in estimation, rows *spec\$y_rows* of data.

Note that *m* is the *maximum* number of breaks allowed. Testing output is reported for every value from 1 through *m*, while estimation corresponds to the number of breaks selected by the procedure under *test_level*. The fit has `print()`, `summary()`, `plot()` and the accessors in [sysbreak-methods](#).

References

- Andrews, D. W. K. (1991). Heteroskedasticity and autocorrelation consistent covariance matrix estimation. *Econometrica*, 59(3), 817–858.
- Andrews, D. W. K., and Monahan, J. C. (1992). An improved heteroskedasticity and autocorrelation consistent covariance matrix estimator. *Econometrica*, 60(4), 953–966.
- Bai, J., and Perron, P. (1998). Estimating and testing linear models with multiple structural changes. *Econometrica*, 66(1), 47–78.
- Qu, Z., and Perron, P. (2007). Estimating and testing structural changes in multivariate regressions. *Econometrica*, 75(2), 459–502.

Examples

```
set.seed(11)
bigt <- 40
x <- rnorm(bigt)
y1 <- 1 + 0.4 * x + rnorm(bigt, sd = 0.3)
y2 <- -0.5 + 0.2 * x + rnorm(bigt, sd = 0.3)
y1[21:40] <- y1[21:40] + 1.2 - 0.6 * x[21:40]
```

```

y2[21:40] <- y2[21:40] - 0.8 + 0.4 * x[21:40]
dat <- data.frame(y1 = y1, y2 = y2, x = x)

fit <- sysbreak(cbind(y1, y2) ~ x, data = dat, m = 1, trim = 0.2)
fit$estimation$breaks

```

sysbreak-methods

Methods for Displaying Structural-Break Fits

Description

Fits returned by [sysbreak](#), [varbreak](#) and [netbreak](#). All refer to the estimates at the selected number of breaks.

Usage

```

## S3 method for class 'sysbreak'
coef(object, ...)

## S3 method for class 'sysbreak'
vcov(object, ...)

## S3 method for class 'sysbreak'
confint(object, parm, level = NULL, type = c("breaks", "coef"), ...)

## S3 method for class 'sysbreak'
residuals(object, ...)

## S3 method for class 'sysbreak'
fitted(object, ...)

## S3 method for class 'sysbreak'
nobs(object, ...)

```

Arguments

object	A fit returned by <code>sysbreak()</code> , <code>varbreak()</code> or <code>netbreak()</code> .
...	Not used.
parm	Breaks (type = "breaks") or coefficients (type = "coef") to report, as indices or row names.
level	Coverage probability. By default the <code>ci_level</code> of the fit, the level of the intervals shown by <code>print()</code> and <code>summary()</code> . For break dates either 0.95 or 0.90, the two levels computed at estimation; for coefficients any value in (0, 1).
type	"breaks" for the break dates or "coef" for the regression coefficients.

Value

coef A named vector of the regime coefficients: all coefficients of regime 1, then those of regime 2, and so on. Each name joins the regime and the coefficient label used by `summary()`, as in "regime1:y1:x" (equation y1, regressor x) or "regime2:mu:node3" for netbreak. Coefficients held constant by `coef_breaks` are repeated in every regime.

vcov The covariance matrix of `coef()`, with rows and columns in the same order and with the same names. When `coef_breaks` holds coefficients constant, the matrix is singular, since the repeated entries of a constant coefficient are perfectly correlated.

confint For `type = "breaks"`, a matrix with one row per break: the lower bound, the estimated break date and the upper bound. For `type = "coef"`, the lower and upper bounds of confidence intervals, estimate $\pm$ normal quantile times standard error, treating the break dates as known.

residuals, fitted Matrices with one column per equation and one row per observation; the row names are the rows of the data (`spec$y_rows`).

nobs The number of observations used.

When no break is selected the regime estimates are not computed, and all but `nobs` stop with an error.

See Also

[summary.sysbreak](#), [plot.sysbreak](#)

Examples

```
set.seed(11)
bigt <- 40
x <- rnorm(bigt)
y1 <- 1 + 0.4 * x + rnorm(bigt, sd = 0.3)
y2 <- -0.5 + 0.2 * x + rnorm(bigt, sd = 0.3)
y1[21:40] <- y1[21:40] + 1.2 - 0.6 * x[21:40]
y2[21:40] <- y2[21:40] - 0.8 + 0.4 * x[21:40]
dat <- data.frame(y1 = y1, y2 = y2, x = x)
fit <- sysbreak(cbind(y1, y2) ~ x, dat, m = 1, trim = 0.3)

coef(fit)
confint(fit)
confint(fit, level = 0.90)
confint(fit, type = "coef")
head(residuals(fit))
nobs(fit)
```

varbreak

Structural Breaks in a VAR

Description

Estimation of a VAR affected by structural breaks in the regression coefficients and/or the variance covariance matrix of the residuals.

Usage

```
varbreak(y, lags, m, intercept = TRUE, coef_breaks = "all",
  test_level = 0.05, ci_level = 0.95,
  trim = 0.15, verbose = FALSE, brv = FALSE,
  robust = c("none", "hc", "hac"), hetq = FALSE, prewhit = FALSE,
  restart = c("none", "brbp", "brbp_residual"),
  restart_B = 10, restart_verbose = FALSE, h = NULL)
```

Arguments

<code>y</code>	Numeric response matrix with one column per variable.
<code>lags</code>	Positive integer, number of lags.
<code>m</code>	Integer, maximum number of breaks allowed.
<code>intercept</code>	Logical. If TRUE, include an intercept in the regressors matrix.
<code>coef_breaks</code>	Coefficient-break selector, applied after the VAR lag matrix is built. The default "all" lets all VAR coefficients change. Use "none" or NULL for no coefficient changes, for which <code>brv = TRUE</code> is required so that the covariance matrix of the residuals can change. Use a list such as <code>list(y1 ~ ., y2 ~ 0)</code> to restrict breaks to selected regression coefficients.
<code>test_level</code>	Significance level used for selecting the number of breaks. One of 0.10, 0.05, 0.025 or 0.01. WDmax decides whether there is any break; if it is not significant the procedure stops and no break is reported. Otherwise the sequential tests of l against $l + 1$ breaks are applied from one break upwards, stopping at the first insignificant test or at m .
<code>ci_level</code>	Coverage probability of the reported confidence intervals for the break dates. Either 0.95 or 0.90.
<code>trim</code>	Trimming parameter. The minimum segment length is $h = \text{round}(\text{trim} * \text{nrow}(y))$. We recommend $h \geq 3(n + k)$, where n is the number of equations and k the number of regressors per equation, and a larger value for larger systems. A warning is issued if $h < 3(n + k)$.
<code>verbose</code>	Logical. If TRUE, print additional progress information from the testing and estimation.
<code>brv</code>	Logical. If TRUE, allow the disturbance covariance matrix to change across regimes.
<code>robust</code>	Inference for the coefficient-break tests, the coefficient standard errors and the break-date confidence intervals. "none" gives Gaussian quasi-likelihood-ratio tests, valid when the errors are serially uncorrelated and conditionally homoskedastic. "hc" gives the robust SupF statistic of Qu and Perron (2007, eq. 20) with a heteroskedasticity-consistent covariance estimate without autocovariance terms, valid when the errors are a martingale difference sequence. "hac" uses instead the quadratic spectral kernel with the automatic bandwidth of Andrews (1991), allowing for serial correlation. With enough lags the VAR errors are serially uncorrelated, and "hc" suffices for the tests and standard errors; with <code>brv = TRUE</code> and conditionally heteroskedastic errors of the ARCH type, the confidence intervals require "hac".

hetq	Logical. If TRUE, allow regressor moments to vary across regimes when computing confidence intervals for break dates.
prewhit	Logical. If TRUE, prewhiten the scores by a VAR(1) before the HAC estimation (Andrews and Monahan, 1992). Used only when robust = "hac".
restart	A bootstrap restarting procedure to check and improve convergence. It applies only when some coefficients are held constant; otherwise it is ignored with a warning. Set to "none" to turn it off, "brbp" for resampling bootstrap restarts, or "brbp_residual" for residual-based restarts.
restart_B	Integer, number of restart values used when restart != "none".
restart_verbose	Logical. If TRUE, print restart progress.
h	Optional positive integer for forecast horizon. If supplied, attach Diebold–Yilmaz generalized spillover measures in spillover.

Details

The interpretation of `testing$statistic` depends on `robust`.

- `robust = "none"` gives the SupLR statistic. The code can test the coefficients specified by `coef_breaks`, the disturbance covariance when `brv = TRUE`, or both.
- `robust = "hc" or "hac"` gives the robust SupF statistic, for the coefficients only, because the covariance-break critical values are case dependent under heteroskedasticity or serial correlation; it assumes a constant error covariance. With `brv = TRUE`, the estimates, standard errors and confidence intervals account for the covariance breaks; under "hc" the intervals further require errors without ARCH-type dependence.

Value

An object of class `c("varbreak", "sysbreak")` containing:

- `spec` A summary of the setup, including the sample size `bigt`, the number of equations `n`, the total number of coefficients per regime `q`, the arguments `m`, `lags`, `intercept`, `trim`, `brv`, `robust`, `hetq`, `prewhit`, `test_level` and `ci_level`, the equation and coefficient names, the retained input rows `y_rows` after discarding the lags, and the `coef_breaks` selector.
- `testing` The tests, reported at `test_level`: statistic, testing no break against k breaks for every $k = 1, \dots, m$, with its `critical_values`; `statistic_type`, "SupLR" or "SupF", and NA when a requested robust test is unavailable; `wdmax` and `wdmax_critical_values`; the sequential tests of l against $l + 1$ breaks, each containing `l`, a statistic and its `critical_values`; and the number of `changing_parameters`, which includes the changing covariance parameters when `robust = "none"`.
- `br_selected` The outcome of the break-number determination at `test_level`: the selected number of breaks, the WDmax statistic with its critical value and significance flag, and the sequential steps taken. `breaks` is 0 when WDmax is not significant, and NA when the tests are unavailable for the requested specification.
- `estimation` The estimates related to the selected number of breaks: estimated breaks; their confidence intervals `ci_breaks`; coefficients by regimes and their confidence intervals; disturbance covariance matrices. This component is NULL when no break is detected, in which case only `testing` and `br_selected` are informative.

spillover When `h` is supplied and at least one break is selected, regime-specific Diebold–Yilmaz generalized spillover measures returned by [dyspill](#).

y The responses used in estimation, rows `spec$y_rows` of `y`.

Note that `m` is the *maximum* number of breaks allowed. Testing output is reported for every value from 1 through `m`, while estimation corresponds to the number of breaks selected by the procedure under `test_level`. The optional spillover component is omitted when `h = NULL`. The fit has `print()`, [summary\(\)](#), [plot\(\)](#) and the accessors in [sysbreak-methods](#).

References

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

Andrews, D. W. K., and Monahan, J. C. (1992). An improved heteroskedasticity and autocorrelation consistent covariance matrix estimator. *Econometrica*, 60(4), 953–966.

Bai, J., and Perron, P. (1998). Estimating and testing linear models with multiple structural changes. *Econometrica*, 66(1), 47–78.

Qu, Z., and Perron, P. (2007). Estimating and testing structural changes in multivariate regressions. *Econometrica*, 75(2), 459–502.

Examples

```
set.seed(12)
bigt <- 45
y <- matrix(0, bigt, 2)
for (tt in 2:bigt) {
  shift <- if (tt > 24) c(0.8, -0.6) else c(0, 0)
  y[tt, ] <- shift +
    c(0.3 * y[tt - 1, 1], -0.2 * y[tt - 1, 2]) +
    rnorm(2, sd = 0.3)
}

fit <- varbreak(y, lags = 1, m = 1, trim = 0.2)
fit$estimation$breaks
```

Index

* datasets

- data1, [3](#)
- data2, [4](#)
- housing, [7](#)
- inflation, [8](#)
- netsim, [12](#)

brtype, [2](#)

coef.sysbreak (sysbreak-methods), [22](#)
confint.sysbreak (sysbreak-methods), [22](#)

data1, [3](#)
data2, [4](#)
dynobreak, [4](#)
dyspill, [5](#), [11](#), [26](#)

fitted.sysbreak (sysbreak-methods), [22](#)

housing, [7](#)

inflation, [8](#)

netbreak, [2](#), [9](#), [17](#), [18](#), [22](#)
netsim, [12](#)
nobs.sysbreak (sysbreak-methods), [22](#)

ordbreak, [13](#), [16](#)
ordcheck, [15](#), [16](#)

plot(), [11](#), [21](#), [26](#)
plot.sysbreak, [17](#), [18](#), [23](#)
print.summary.sysbreak
(summary.sysbreak), [18](#)

residuals.sysbreak (sysbreak-methods),
[22](#)

summary(), [11](#), [21](#), [26](#)
summary.sysbreak, [17](#), [18](#), [23](#)
sysbreak, [2](#), [3](#), [17](#), [18](#), [19](#), [22](#)
sysbreak-methods, [22](#)

varbreak, [2](#), [17](#), [18](#), [22](#), [23](#)

vcov.sysbreak (sysbreak-methods), [22](#)