Package ‘breaktest’

June 29, 2022

Version 0.0.2

Date 2022-04-28

Title My First Collection of Functions

Author Anton Skrobotov [aut, ctb],
Alexey Tsarev [aut, ctb],
Vadim Zyamalov [aut, cre, ctb]

Maintainer Vadim Zyamalov <zyamalov@ranepa.rut>

Depends R (>= 3.5.0)

Encoding UTF-8

Imports doSNOW, foreach, stringr, zeallot

Description This package contains a set of different unit root and
cointegration tests in the presence of structural breaks in the data.

License GPL (>= 2)

URL https://github.com/d9d6ka/breaktest

BugReports https://github.com/d9d6ka/breaktest/issues

RoxygenNote 7.2.0

NeedsCompilation no

R topics documented:

`ACF` *Calculating ACF values.*

Description

Calculating ACF values.

Usage

```
ACF(y)
```

Arguments

`y`	The input time series of interest.

Value

The vector of ACF values for s from 0 to $N - 1$.

`ADF.test` *ADF.test - simple implementation of ADF test*

Description

ADF.test - simple implementation of ADF test

Usage

```
ADF.test(
  y,
  const = TRUE,
  trend = FALSE,
  max.lag = 0,
  criterion = NULL,
  modified.criterion = FALSE,
  rescale.criterion = FALSE
)
```

Arguments

`y`	The input time series of interest.
`const`	Include const to the model if TRUE.
`trend`	Include trend to the model if TRUE.
`max.lag`	Maximum lag number
`criterion`	A criterion used to select number of lags. If lag selection is not needed keep this NULL.
`modified.criterion`	Whether the unit-root test modificaton is needed.
`rescale.criterion`	Whether the rescaling informational criterion is needed. Designed to cope with heteroscedasticity in residuals.

Details

A function for ADF test with the ability to select the number of lags. Lags are selected by informational criterions which can be modified as in Ng and Perron (2001) and Cavaliere et al. (2015).

Due to the Frisch-Waugh-Lovell theorem we first detrend ‘y‘ and then apply the test to the detrended series.

Value

List containing:

- y
- const
- trend
- residuals
- coefficient estimates
- t-statistic value
- critical value
- Number of lags
- indicator of stationarity

References

Cavaliere, Giuseppe, Peter C. B. Phillips, Stephan Smeekes, and A. M. Robert Taylor. "Lag Length Selection for Unit Root Tests in the Presence of Nonstationary Volatility." Econometric Reviews 34, no. 4 (April 21, 2015): 512–36. https://doi.org/10.1080/07474938.2013.808065.

Ng, Serena, and Pierre Perron. "Lag Length Selection and the Construction of Unit Root Tests with Good Size and Power." Econometrica 69, no. 6 (2001): 1519–54. https://doi.org/10.1111/1468-0262.00256.

`ADF.test.S` *Detrending bootstrap test by Smeekes (2013)*

Description

Detrending bootstrap test by Smeekes (2013)

Usage

```
ADF.test.S(
  y,
  const = TRUE,
  trend = FALSE,
  c = 0,
  gamma = 0,
  trim = 0.15,
  max.lag = 0,
  criterion = NULL,
  modified.criterion = FALSE,
  iter = 999
)
```

Arguments

`y`	The series of interest.
`const`	Whether the constant is to included.
`trend`	Whether the trend is to be included.
`c`	A filtration parameter.
`gamma`	A detrending type selection parameter. If 0 the OLS detrending is applied, if 1 the GLS detrending is applied, otherwise the autocorrelation coefficient is calculated as $1 + c^{\gamma}T^{-\gamma}$.
`trim`	The trimming parameter.
`max.lag`	The maximum lag for inner ADF testing.
`criterion`	A criterion used to select number of lags. If lag selection is not needed keep this NULL.
`modified.criterion`	Whether the unit-root test modificaton is needed.
`iter`	The number of bootstrap steps.

Details

This bootstrap test is based on the recursive detrending procedure of Taylor (2002). The main idea is to apply the standard ADF test to the series with nuissanse parameters eliminated.

Critical values are calculated via a bootstrapping using MacKinnon-like regressions. For each number of observations and each number of variables obtained were 1999 values of test statistics. After that 1st, 2.5-th, 5-th, 10-th, and 97.5-th percentiles were calculated and saved along with the corresponding number of observations. This step was repeated 5 times to cope with possible biases. After that MacKunnon-like regressions were estimated.

References

Taylor, A. M. Robert. "Regression-Based Unit Root Tests With Recursive Mean Adjustment for Seasonal and Nonseasonal Time Series." Journal of Business & Economic Statistics 20, no. 2 (April 2002): 269–81. https://doi.org/10.1198/073500102317352001.

MacKinnon, James G. "Critical Values for Cointegration Tests." Working Paper. Economics Department, Queen's University, January 2010. https://ideas.repec.org/p/qed/wpaper/1227.html.

Smeekes, Stephan. "Detrending Bootstrap Unit Root Tests." Econometric Reviews 32, no. 8 (July 2013): 869–91. https://doi.org/10.1080/07474938.2012.690693.

`coint.test.GH` *Gregory-Hansen test for the absense of cointegration*

Description

Gregory and Hansen (1996) test for the null hypothesis of no cointegration under a possible structural break at the unknown moment of time.

The authors proposed ADF- and Z-type tests, slightly modified to allow the presence of a possible regime shift. Three type of shifts are allowed:

- a shift in the constant,

- a shift in the constand with the trend included,
- and a shift in the constant and the cointegrating vector.

Critical values are calculated via the adopted MacKinnon procedure of estimating the model for the response surface.

Usage

```
coint.test.GH(
  ...,
  shift = "level",
  trim = 0.15,
  max.lag = 10,
  criterion = "aic",
  add.criticals = TRUE
)
```

Arguments

`...`	Variables of interest.
`shift`	Expected break type.
`trim`	The trimming parameter to calculate break moment bounds.
`max.lag`	The maximum number of lags for the internal ADF testing.
`criterion`	The criterion for lag selection.
`add.criticals`	Whether critical values are to be returned. This argument is needed to suppress the calculation of critical values during the precalculation of tables needed for the p-values estimating.

References

MacKinnon, James G. "Critical Values for Cointegration Tests." Working Paper. Working Paper. Economics Department, Queen's University, January 2010. https://ideas.repec.org/p/qed/wpaper/1227.html.

Gregory, Allan W., and Bruce E. Hansen. "Residual-Based Tests for Cointegration in Models with Regime Shifts." Journal of Econometrics 70, no. 1 (January 1, 1996): 99–126. https://doi.org/10.1016/0304-4076(69)41685-7.

`CPST.rescale` *Generating rescaled series as in Cavaliere et al. (2015).*

Description

Generating rescaled series as in Cavaliere et al. (2015).

Usage

```
CPST.rescale(d.y, x, deter, k, max.lag)
```

Arguments

`d.y`	The series of first differences.
`x`	The matrix of ADF RHS variables.
`deter`	The matrix of deterministic variables for detrending.
`k`	The lag of the corresponding ADF model.
`max.lag`	The maximum possible lag.

References

Cavaliere, Giuseppe, Peter C. B. Phillips, Stephan Smeekes, and A. M. Robert Taylor. “Lag Length Selection for Unit Root Tests in the Presence of Nonstationary Volatility.” Econometric Reviews 34, no. 4 (April 21, 2015): 512–36. https://doi.org/10.1080/07474938.2013.808065.

`determinants.KPSS.1.break` *Construct determinant variables.*

Description

Construct determinant variables.

Usage

```
determinants.KPSS.1.break(model, N, break.point)
```

Arguments

`model`	**1** Model with trend, break in const. **2** Model with const and trend, break in const. **3** Model with const and trend, break in trend. **4** Model with const and trend, break in const and trend.
`N`	Number of observations.
`break.point`	Break point.

Details

Procedure to compute deterministic terms for KPSS with 1 structural break.

Value

Matrix of determinant variables.

`determinants.KPSS.2.breaks`
Construct determinant variables.

Description

Construct determinant variables.

Usage

```
determinants.KPSS.2.breaks(model, N, break.point)
```

Arguments

`model`	**1** for the AA (without trend) model. **2** for the AA (with trend) model. **3** for the BB model. **4** for the CC model. **5** for the AC-CA model. **6** for the AC-CA model. **7** for the AC-CA model.
`N`	Number of observations.
`break.point`	Positions for the first and second structural breaks (respective to the origin which is 1).

Details

Procedure to compute deterministic terms for KPSS with 2 structural breaks.

Value

Matrix of deterministic terms.

`determinants.KPSS.N.breaks`
Deterministic terms for multi-break KPSS

Description

Procedure to compute deterministic terms for KPSS with m structural breaks.

Usage

```
determinants.KPSS.N.breaks(model, N, break.point, const = FALSE, trend = FALSE)
```

Arguments

`model`	A scalar or vector of **1** for the break in const. **2** for the break in trend. **3** for the break in const and trend.
`N`	Number of observations.
`break.point`	Array of structural breaks.
`const`	Include constant if **TRUE**.
`trend`	Include trend if **TRUE**.

Details

model should be either a scalar or a vector of the same size as the **break.point**. If scalar **model** will be repeated till the length of **break.point** is achieved.

Value

Matrix of deterministic terms.

`eos.break.test` *Andrews-Kim (2006) test*

Description

Test for structural break at the end of the sample.

Usage

```
eos.break.test(eq, m, dataset)
```

Arguments

`eq`	Base model formula. At the moment all the variables included should be defined explicitly, dynamic regressors (i.e. functions etc.) are not supported.
`m`	Post-break period length.
`dataset`	Source of the data.

Details

See Andrews and Kim (2006) for the detailed description.

Value

The list, containing

- **m**
- estimated values of P- and R-tests.
- sequences of auxiliary statistics P_j and R_j.
- the corresponding p-values.

References

Andrews, D. W. K. "End-of-Sample Instability Tests." Econometrica 71, no. 6 (2003): 1661–94. https://doi.org/10.1111/1468-0262.00466.

Andrews, Donald W. K, and Jae-Young Kim. "Tests for Cointegration Breakdown Over a Short Time Period." Journal of Business & Economic Statistics 24, no. 4 (2006): 379–94. https://doi.org/10.1198/07350010600

`GLS` *Custom GLS with extra information*

Description

Getting GLS estimates of betas, residuals, forecasted values and t-values.

Usage

```
GLS(y, z, c)
```

Arguments

`y`	Dependent variable.
`c`	Coefficient for ρ calculation.
`x`	Explanatory variables.

Value

The list of betas, residuals, forecasted values and t-values.

`GSADF.bootstrap.test` *Generalized supremum ADF test with wild bootstrap.*

Description

Generalized supremum ADF test with wild bootstrap.

Usage

```
GSADF.bootstrap.test(
  y,
  trim = 0.01 + 1.8/sqrt(length(y)),
  const = TRUE,
  alpha = 0.05,
  iter = 4 * 200,
  seed = round(10^4 * sd(y))
)
```

Arguments

`y`	The input time series of interest.
`trim`	Trimming parameter to determine the lower and upper bounds.
`const`	Whether the constant needs to be included.
`alpha`	The significance level of interest.
`iter`	The number of iterations/
`seed`	The seed parameter for the random number generator.

References

Kurozumi, Eiji, Anton Skrobotov, and Alexey Tsarev. "Time-Transformed Test for the Explosive Bubbles under Non-Stationary Volatility." arXiv, November 15, 2021. http://arxiv.org/abs/2012.13937.

`GSADF.test` *Generalized supremum ADF test.*

Description

Generalized supremum ADF test.

Usage

```
GSADF.test(
  y,
  trim = 0.01 + 1.8/sqrt(length(y)),
  const = TRUE,
  add.p.value = TRUE
)
```

Arguments

`y`	The input time series of interest.
`trim`	Trimming parameter to determine the lower and upper bounds.
`const`	Whether the constant needs to be included.
`add.p.value`	Whether the p-value is to be returned. This argument is needed to suppress the calculation of p-values during the precalculation of tables needed for the p-values estimating.

References

Kurozumi, Eiji, Anton Skrobotov, and Alexey Tsarev. "Time-Transformed Test for the Explosive Bubbles under Non-Stationary Volatility." arXiv, November 15, 2021. http://arxiv.org/abs/2012.13937.

GSTADF.test *Generalized supremum ADF test with time transformation.*

Description

Generalized supremum ADF test with time transformation.

Usage

```
GSTADF.test(
  y,
  trim = 0.01 + 1.8/sqrt(length(y)),
  const = FALSE,
  omega.est = TRUE,
  h = "auto_CV",
  truncated = TRUE,
  is.reindex = TRUE,
  ksi.input = "auto",
  hc = 1,
  pc = 1,
  add.p.value = TRUE
)
```

Arguments

`y`	The input time series of interest.
`trim`	Trimming parameter to determine the lower and upper bounds.
`const`	Whether the constant needs to be included.
`omega.est`	Whether the variance of Nadaraya-Watson residuals should be used.
`truncated`	Whether the truncation of Nadaraya-Watson residuals is needed.
`is.reindex`	Whether the Cavaliere and Taylor (2008) time transformation is needed.
`ksi.input`	The value of the truncation parameter. Can be either 'auto' or the explicit numerical value. In the former case the numeric value is estimated.
`hc`	The scaling parameter for Nadaraya-Watson bandwidth.
`pc`	The scaling parameter for the estimated truncation parameter value.
`add.p.value`	Whether the p-value is to be returned. This argument is needed to suppress the calculation of p-values during the precalculation of tables needed for the p-values estimating.

References

Cavaliere, Giuseppe, and A. M. Robert Taylor. "Time-Transformed Unit Root Tests for Models with Non-Stationary Volatility." Journal of Time Series Analysis 29, no. 2 (March 2008): 300–330. https://doi.org/10.1111/j.1467-9892.2007.00557.x.

Kurozumi, Eiji, Anton Skrobotov, and Alexey Tsarev. "Time-Transformed Test for the Explosive Bubbles under Non-Stationary Volatility." arXiv, November 15, 2021. http://arxiv.org/abs/2012.13937.

`info.criterion` *Information criterions*

Description

Information criterions

Usage

```
info.criterion(resid, extra, modification = FALSE, alpha = 0, y = NULL)
```

Arguments

`resid`	Input residuals needed for estimating the values of information criterions.
`extra`	Number of extra parameters needed for estimating the punishment term.
`modification`	Whether the unit-root test modificaton is needed. See Ng and Perron (2001) for further information.
`alpha`	The coefficient α of y_{t-1} in ADF model. Needed only for criterion modification purposes.
`y`	The vector of y_{t-1} in ADF model. Needed only for criterion modification purposes.

Details

Calculating the value of the following informational criterions:

- Akaike,
- Schwarz (Bayesian),
- Hannan-Quinn,
- Liu et al.

Value

The value of the selected informational criterion.

References

Ng, Serena, and Pierre Perron. “Lag Length Selection and the Construction of Unit Root Tests with Good Size and Power.” Econometrica 69, no. 6 (2001): 1519–54. https://doi.org/10.1111/1468-0262.00256.

KP *Kejrival-Perron procedure of breaks number detection.*

Description

Kejrival-Perron procedure of breaks number detection.

Usage

```
KP(y, const = FALSE, breaks = 1, criterion = "aic", trim = 0.15)
```

Arguments

`y`	The input series of interest.
`const`	Allowing the break in constant.
`breaks`	Number of breaks.
`criterion`	Needed information criterion: aic, bic, hq or lwz.
`trim`	A trimming value for a possible break date bounds.

References

Kejriwal, Mohitosh, and Pierre Perron. “A Sequential Procedure to Determine the Number of Breaks in Trend with an Integrated or Stationary Noise Component: Determination of Number of Breaks in Trend.” Journal of Time Series Analysis 31, no. 5 (September 2010): 305–28. https://doi.org/10.1111/j.1467-9892.2010.00666.x.

KPSS *Auxiliary function returning KPSS statistic value.*

Description

Auxiliary function returning KPSS statistic value.

Usage

```
KPSS(resid, variance)
```

Arguments

`resid`	The series of residuals.
`variance`	The value of the long-run variance.

`KPSS.1.break` *KPSS-test with known structural break*

Description

Computes the cointegration test with one known structural break.

Usage

```
KPSS.1.break(y, x, model, break.point, weakly.exog = TRUE, ll.init)
```

Arguments

`y`	(Tx1)-vector of the dependent variable.
`x`	(Txk)-matrix of explanatory stochastic regressors.
`model`	**1** for model An. **2** for model A. **3** for model B. **4** for model C. **5** for model D. **6** for model E.
`break.point`	Position of the break point.
`weakly.exog`	Exogeneity of the stochastic regressors **TRUE** if the regressors are weakly exogenous, **FALSE** if the regressors are not weakly exogenous (DOLS is used in this case).
`ll.init`	Scalar, defines the initial number of leads and lags for DOLS.

Details

The code provided is the original GAUSS code ported to R.

See Carrion-i-Silvestre and Sansó (2006) for further details.

Value

beta DOLS estimates of the coefficients regressors.

tests SC test (coinKPSS-test).

resid Residuals of the model.

t.beta Individual significance t-statistics.

break_point Break points.

`KPSS.1.break.unknown` *KPSS-test of cointegration*

Description

Procedure for testing the null of cointegration in the possible presence of structural breaks.

Usage

```
KPSS.1.break.unknown(y, x, model, weakly.exog, ll.init)
```

Arguments

`y`	(Tx1)-vector of the dependent variable
`x`	(Txk)-matrix of explanatory stochastic regressors
`model`	Scalar, denotes the deterministic component: **1** for model An. **2** for model A. **3** for model B. **4** for model C. **5** for model D. **6** for model E.
`weakly.exog`	Exogeneity of the stochastic regressors **TRUE** if the regressors are weakly exogenous, **FALSE** if the regressors are not weakly exogenous (DOLS is used in this case).
`ll.init`	Scalar, defines the initial number of leads and lags for DOLS.

Details

Computes the cointegration test with one unknown structural break where the break point is estimated either minimizing the value of the statistic or the sum of the squared residuals. The estimation of the cointegrating relationship bases on DOLS.

The code provided is the original GAUSS code ported to R.

See Carrion-i-Silvestre and Sansó (2006) for further details.

Value

(2x2)-matrix, where the first rows gives the value of the min(SC) test and the estimated break point; the second row gives the value of the SC statistic, where the break point is estimated as min(SSR).

KPSS.2.breaks *KPSS-test with 2 known structural breaks*

Description

Procedure to compute the KPSS test with two structural breaks

Usage

```
KPSS.2.breaks(y, model, break.point, max.lag, kernel)
```

Arguments

`y`	(Tx1)-vector of time series
`model`	**1** for the AA (without trend) model. **2** for the AA (with trend) model. **3** for the BB model. **4** for the CC model. **5** for the AC-CA model.
`max.lag`	scalar, with the maximum order of the parametric correction. The final order of the parametric correction is selected using the BIC information criterion.
`kernel`	Kernel for calculating long-run variance **bartlett** for Bartlett kernel. **quadratic** for Quadratic Spectral kernel. **NULL** for the Kurozumi's proposal, using Bartlett kernel.
`tb1`	The first break point.
`tb2`	The second break point.

Details

The break points are known

The code provided is the original GAUSS code ported to R.

See Carrion-i-Silvestre and Sansó (2007) for further details.

Value

beta DOLS estimates of the coefficients regressors.

tests SC test (coinKPSS-test).

resid Residuals of the model.

break_point Break points.

KPSS.2.breaks.unknown — *KPSS-test with 2 unknown structural breaks*

Description

Procedure to compute the KPSS test with two structural breaks

Usage

```
KPSS.2.breaks.unknown(y, model, max.lag = 0, kernel = "bartlett")
```

Arguments

`y`	(Tx1)-vector of time series.
`model`	**1** for the AA (without trend) model. **2** for the AA (with trend) model. **3** for the BB model. **4** for the CC model. **5** for the AC-CA model.
`max.lag`	scalar, with the maximum order of the parametric correction. The final order of the parametric correction is selected using the BIC information criterion.
`kernel`	Kernel for calculating long-run variance **bartlett** for Bartlett kernel. **quadratic** for Quadratic Spectral kernel. **NULL** for the Kurozumi's proposal, using Bartlett kernel.

Details

The break points are known

The code provided is the original GAUSS code ported to R.

See Carrion-i-Silvestre and Sansó (2007) for further details.

Value

Value of test statistic.

KPSS.HLT — *Unit root testing procedure under a single structural break.*

Description

Unit root testing procedure under a single structural break.

Usage

```
KPSS.HLT(y, const = FALSE, trim = 0.15)
```

Arguments

`y`	The series of interest.
`const`	Whether a constant should be included.
`trim`	The trimming parameter to find the lower and upper bounds of possible break dates.

References

Harvey, David I., Stephen J. Leybourne, and A. M. Robert Taylor. "Unit Root Testing under a Local Break in Trend." Journal of Econometrics 167, no. 1 (2012): 140–67.

`KPSS.N.breaks` *KPSS-test with 2 known structural breaks*

Description

Procedure to compute the KPSS test with two structural breaks

Usage

```
KPSS.N.breaks(
  y,
  x,
  model,
  break.point,
  const = FALSE,
  trend = FALSE,
  weakly.exog = TRUE,
  lags.init,
  leads.init,
  max.lag,
  kernel,
  criterion = "bic"
)
```

Arguments

`y`	(Tx1)-vector of time series.
`x`	(Txk)-matrix of explanatory stochastic regressors.
`model`	A scalar or vector of **1** for the break in const. **2** for the break in trend. **3** for the break in const and trend.
`break.point`	Array of structural breaks.
`trend`	Include trend if 'TRUE'.
`weakly.exog`	Boolean where we specify whether the stochastic regressors are exogenous or not **TRUE** if the regressors are weakly exogenous,

	FALSE if the regressors are not weakly exogenous (DOLS is used in this case).
`lags.init`	Scalar, defines the initial number of lags for DOLS.
`leads.init`	Scalar, defines the initial number of leads for DOLS.
`max.lag`	scalar, with the maximum order of the parametric correction. The final order of the parametric correction is selected using the BIC information criterion.
`kernel`	Kernel for calculating long-run variance **bartlett** for Bartlett kernel. **quadratic** for Quadratic Spectral kernel. **NULL** for the Kurozumi's proposal, using Bartlett kernel.
`criterion`	Information criterion for DOLS lags and leads selection: aic, bic or lwz.

Details

The break points are known.

The code provided is the original GAUSS code ported to R.

See Carrion-i-Silvestre and Sansó (2006) for further details.

Value

beta DOLS estimates of the coefficients regressors.

tests SC test (coinKPSS-test).

resid Residuals of the model.

t.beta t-statistics for ‘beta‘.

DOLS.lags The estimated number of lags and leads in DOLS.

break_point Break points.

`KPSS.N.breaks.bootstrap` *Calculating p-values using bootstrap*

Description

Procedure to compute the KPSS test with two structural breaks

Usage

```
KPSS.N.breaks.bootstrap(
  y,
  x,
  model,
  break.point,
  const = FALSE,
  trend = FALSE,
  weakly.exog = TRUE,
  lags.init,
  leads.init,
  max.lag,
```

```
    kernel,
    iter = 9999,
    bootstrap = "sample",
    criterion = "bic"
)
```

Arguments

`y`	(Tx1)-vector of time series.
`x`	(Txk)-matrix of explanatory stochastic regressors.
`model`	A scalar or vector of **1** for the break in const. **2** for the break in trend. **3** for the break in const and trend.
`break.point`	Array of structural breaks.
`const`	Include constant if **TRUE**.
`trend`	Include trend if **TRUE**.
`weakly.exog`	Exogeneity of the stochastic regressors **TRUE** if the regressors are weakly exogenous, **FALSE** if the regressors are not weakly exogenous (DOLS is used in this case).
`max.lag`	scalar, with the maximum order of the parametric correction. The final order of the parametric correction is selected using the BIC information criterion.
`kernel`	Kernel for calculating long-run variance **bartlett** for Bartlett kernel. **quadratic** for Quadratic Spectral kernel. **NULL** for the Kurozumi's proposal, using Bartlett kernel.
`iter`	Number of bootstrap iterations.
`bootstrap`	Type of bootstrapping: **sample** sampling from residuals with replacement. **Cavaliere-Taylor** multiplying residuals by $N(0,1)$-distributed variable. **Rademacher** multiplying residuals by Rademacher-distributed variable.
`criterion`	Information criterion for DOLS lags and leads selection: aic, bic or lwz.
`ll.init`	Scalar, defines the initial number of leads and lags for DOLS.

Details

The break points are known.

See Cavaliere and Taylor (2006) for further details.

Value

List of 3 elements:

test The value of KPSS test statistic.

p.value The estimates p-value.

bootstrapped Bootstrapped auxiliary statistics.

`lagn` *Produce a vector lagged backward of forward*

Description

Produce a vector lagged backward of forward

Usage

```
lagn(x, i, na = NA)
```

Arguments

`x`	Initial vector.
`i`	Size of lag (lead if negative).

Value

Lagged or leaded vector.

`lr.var.bartlett` *Calculating long-run variance*

Description

Calculating long-run variance

Usage

```
lr.var.bartlett(e, l = NULL)
```

Arguments

`e`	(Tx1) vector or residuals.

Value

Long-run variance.

`lr.var.bartlett.AK` *Calculating long-run variance*

Description

Procedure ALRVR to estimate the long-run variance as in Andrews (1991) and Kurozumi (2002).

Usage

```
lr.var.bartlett.AK(e)
```

Arguments

`e`	(Tx1) vector or residuals.

Value

Long-run variance.

References

Andrews, Donald W. K. “Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimation.” Econometrica 59, no. 3 (1991): 817–58. https://doi.org/10.2307/2938229.

Kurozumi, Eiji. “Testing for Stationarity with a Break.” Journal of Econometrics 108, no. 1 (May 1, 2002): 63–99. https://doi.org/10.1016/S0304-4076(01)00106-3.

`lr.var.quadratic` *Estimating heteroscedasticity and autocorrelation consistent variance*

Description

Estimating heteroscedasticity and autocorrelation consistent variance

Usage

```
lr.var.quadratic(y)
```

References

Andrews, Donald W. K. “Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimation.” Econometrica 59, no. 3 (1991): 817–58. https://doi.org/10.2307/2938229.

lr.var.SPC *Calculating long-run variance*

Description

Procedure ALRVR to estimate the long-run variance as in Sul, Phillips and Choi (2003).

Usage

```
lr.var.SPC(e, max.lag = 0, kernel = "bartlett", criterion = "bic")
```

Arguments

e	(Tx1) vector or residuals.
max.lag	Maximum number of lags. The exact number is selected by information criterions.
kernel	Kernel for calculating long-run variance **bartlett** for Bartlett kernel. **quadratic** for Quadratic Spectral kernel. **NULL** for the Kurozumi's proposal, using Bartlett kernel.
criterion	The information crietreion: bic, aic or lwz.

Details

Used are Quadratic Spectral and Bartlett kernels.

Value

Long-run variance.

MDF.CHLT *MDF test for a single break and possible heterscedasticity*

Description

MDF test for a single break and possible heterscedasticity

Usage

```
MDF.CHLT(y, max.lag = 10, trim = 0.15, iter = 499)
```

Arguments

y	The series of interest
max.lag	The maximum possible lag.
trim	The trimming parameter for lag selection

References

Cavaliere, Giuseppe, David I. Harvey, Stephen J. Leybourne, and A.M. Robert Taylor. “Testing for Unit Roots in the Presence of a Possible Break in Trend and Nonstationary Volatility.” Econometric Theory 27, no. 5 (October 2011): 957–91. https://doi.org/10.1017/S0266466610000605.

`MDF.multiple` *MDF procedure for multiple unknown breaks.*

Description

MDF procedure for multiple unknown breaks.

Usage

```
MDF.multiple(
  y,
  const = FALSE,
  breaks = 1,
  breaks.star = 1,
  trim = 0.15,
  ZA = FALSE
)
```

Arguments

`y`	The series of interest.
`const`	Whether the constant term should be included.
`breaks`	Number of breaks.
`breaks.star`	Number of breaks got from the Kejrival-Perron procedure.
`trim`	A trimming value for a possible break date bounds.
`ZA`	Whether ZA variant should be used.

`MDF.single` *MDF procedure for a single unknown break.*

Description

MDF procedure for a single unknown break.

Usage

```
MDF.single(y, const = FALSE, trend = FALSE, trim = 0.15)
```

Arguments

`y`	The series of interest.
`const`	Whether the constant term should be included.
`trend`	Whether the trend term should be included.
`trim`	A trimming value for a possible break date bounds.

`NW.estimation` *NW.estimation - Nadaraya–Watson kernel regression.*

Description

NW.estimation - Nadaraya–Watson kernel regression.

Usage

```
NW.estimation(y, x, h, kernel = "unif")
```

Arguments

`y`	LHS dependent variable.
`x`	RHS explanation variable.
`h`	Bandwidth.
`kernel`	Needed kernel, currently only ‘unif‘ and ‘gauss‘.

Value

A list of arguments as well as the estimated coefficient vector and residuals.

References

Harvey, David I., S. Leybourne, Stephen J., and Yang Zu. “Nonparametric Estimation of the Variance Function in an Explosive Autoregression Model.” School of Economics. University of Nottingham, 2022.

`NW.loocv` *NW.loocv - LOO-CV for h in Nadaraya–Watson kernel regression.*

Description

NW.loocv - LOO-CV for h in Nadaraya–Watson kernel regression.

Usage

```
NW.loocv(y, x, kernel = "unif")
```

Arguments

`y`	LHS dependent variable.
`x`	RHS explanation variable.
`kernel`	Needed kernel, currently only ‘unif‘ and ‘gauss‘.

Value

A list of arguments as well as the estimated bandwidth ‘h‘.

References

Harvey, David I., S. Leybourne, Stephen J., and Yang Zu. "Nonparametric Estimation of the Variance Function in an Explosive Autoregression Model." School of Economics. University of Nottingham, 2022.

`NW.volatility` *NW.volatility - Nadaraya–Watson kernel volatility estimation*

Description

NW.volatility - Nadaraya–Watson kernel volatility estimation

Usage

```
NW.volatility(e, h, kernel = "unif")
```

Arguments

`e`	The series of interest.
`h`	Bandwidth.
`kernel`	Needed kernel, currently only 'unif' and 'gauss'.

Value

A list of arguments as well as the estimated omega and s.e.

References

Cavaliere, Giuseppe, Peter C. B. Phillips, Stephan Smeekes, and A. M. Robert Taylor. "Lag Length Selection for Unit Root Tests in the Presence of Nonstationary Volatility." Econometric Reviews 34, no. 4 (April 21, 2015): 512–36. https://doi.org/10.1080/07474938.2013.808065.

`OLS` *Custom OLS with extra information*

Description

Getting OLS estimates of betas, residuals, forecasted values and t-values.

Usage

```
OLS(y, x)
```

Arguments

`y`	Dependent variable.
`x`	Explanatory variables.

Value

The list of betas, residuals, forecasted values and t-values.

p.values.SADF *Interpolating p-value for intermediate observation numbers.*

Description

Interpolating p-value for intermediate observation numbers.

Usage

```
p.values.SADF(statistic, N.obs, cr.values)
```

Arguments

statistic	The statistic value.
N.obs	The number of observations.
cr.values	The set of precalculated tables.

PY.sequential *Sequential Perron-Yabu (2009) statistic for breaks at unknown date.*

Description

Sequential Perron-Yabu (2009) statistic for breaks at unknown date.

Usage

```
PY.sequential(
  y,
  const = FALSE,
  breaks = 1,
  criterion = "aic",
  trim = 0.15,
  max.lag = 1
)
```

Arguments

y	The input series of interest.
const	Allowing the break in constant.
breaks	Number of breaks.
criterion	Needed information criterion: aic, bic, hq or lwz.
trim	A trimming value for a possible break date bounds.
max.lag	The maximum possible lag in the model.

References

Kejriwal, Mohitosh, and Pierre Perron. "A Sequential Procedure to Determine the Number of Breaks in Trend with an Integrated or Stationary Noise Component: Determination of Number of Breaks in Trend." Journal of Time Series Analysis 31, no. 5 (September 2010): 305–28. https://doi.org/10.1111/j.1467-9892.2010.00666.x.

PY.single	*Perron-Yabu (2009) statistic for break at unknown date.*

Description

Perron-Yabu (2009) statistic for break at unknown date.

Usage

```
PY.single(
  y,
  const = FALSE,
  trend = FALSE,
  criterion = "aic",
  trim = 0.15,
  max.lag
)
```

Arguments

y	The input series of interest.
const	Allowing the break in constant.
trend	Allowing the break in trend.
criterion	Needed information criterion: aic, bic, hq or lwz.
trim	A trimming value for a possible break date bounds.
max.lag	The maximum possible lag in the model.

References

Perron, Pierre, and Tomoyoshi Yabu. "Testing for Shifts in Trend With an Integrated or Stationary Noise Component." Journal of Business & Economic Statistics 27, no. 3 (July 2009): 369–96. https://doi.org/10.1198/jbes.2009.07268.

reindex	*reindex - function that makes reindexing.*

Description

reindex - function that makes reindexing.

Usage

```
reindex(u)
```

Arguments

u	The residuals series for reindexing.

References

Cavaliere, Giuseppe, and A. M. Robert Taylor. “Time-Transformed Unit Root Tests for Models with Non-Stationary Volatility.” Journal of Time Series Analysis 29, no. 2 (March 2008): 300–330. https://doi.org/10.1111/j.1467-9892.2007.00557.x.

Kurozumi, Eiji, Anton Skrobotov, and Alexey Tsarev. “Time-Transformed Test for the Explosive Bubbles under Non-Stationary Volatility.” arXiv, November 15, 2021. http://arxiv.org/abs/2012.13937.

SADF.bootstrap.test *Supremum ADF test with wild bootstrap.*

Description

Supremum ADF test with wild bootstrap.

Usage

```
SADF.bootstrap.test(
  y,
  trim = 0.01 + 1.8/sqrt(length(y)),
  const = TRUE,
  alpha = 0.05,
  iter = 4 * 200,
  seed = round(10^4 * sd(y))
)
```

Arguments

`y`	The input time series of interest.
`trim`	Trimming parameter to determine the lower and upper bounds.
`const`	Whether the constant needs to be included.
`alpha`	The significance level of interest.
`iter`	The number of iterations/
`seed`	The seed parameter for the random number generator.

References

Kurozumi, Eiji, Anton Skrobotov, and Alexey Tsarev. “Time-Transformed Test for the Explosive Bubbles under Non-Stationary Volatility.” arXiv, November 15, 2021. http://arxiv.org/abs/2012.13937.

`SADF.test` *Supremum ADF test.*

Description

Supremum ADF test.

Usage

```
SADF.test(
  y,
  trim = 0.01 + 1.8/sqrt(length(y)),
  const = TRUE,
  add.p.value = TRUE
)
```

Arguments

`y`	The input time series of interest.
`trim`	Trimming parameter to determine the lower and upper bounds.
`const`	Whether the constant needs to be included.
`add.p.value`	Whether the p-value is to be returned. This argument is needed to suppress the calculation of p-values during the precalculation of tables needed for the p-values estimating.

References

Kurozumi, Eiji, Anton Skrobotov, and Alexey Tsarev. “Time-Transformed Test for the Explosive Bubbles under Non-Stationary Volatility.” arXiv, November 15, 2021. http://arxiv.org/abs/2012.13937.

`sb.GSADF.test` *Sign-based SADF test (HLZ, 2019).*

Description

Sign-based SADF test (HLZ, 2019).

Usage

```
sb.GSADF.test(
  y,
  r0 = 0.01 + 1.8/sqrt(length(y)),
  const = TRUE,
  alpha = 0.05,
  iter = 4 * 200,
  urs = TRUE,
  seed = round(10^4 * sd(y))
)
```

seasonal.dummies *Generating monthly seasonal dummy variables.*

Description

Generating monthly seasonal dummy variables.

Usage

```
seasonal.dummies(N)
```

Arguments

`N`	number of observations.

Value

The matrix of values od seasonal dummies.

segments.GLS *Procedure to minimize the GLS-SSR for 1 break point*

Description

Procedure to minimize the GLS-SSR for 1 break point

Usage

```
segments.GLS(
  y,
  const = FALSE,
  trend = FALSE,
  breaks = 1,
  first.break = NULL,
  last.break = NULL,
  trim = 0.15
)
```

Arguments

`y`	Variable of interest.
`const`	Whether there is a break in the constant.
`trend`	Whether there is a break in the trend.
`breaks`	Number of breaks.
`first.break`	First possible break point.
`last.break`	Last possible break point.
`trim`	Trim value to calculate ‘first.break‘ and ‘last.break‘ if not provided.

Value

The point of possible break.

References

Skrobotov, Anton. "On Trend Breaks and Initial Condition in Unit Root Testing." Journal of Time Series Econometrics 10, no. 1 (2018): 1–15. https://doi.org/10.1515/jtse-2016-0014.

`segments.OLS` *Find $m + 1$ optimal partitions*

Description

Find $m + 1$ optimal partitions

Usage

```
segments.OLS(y, x, m = 1, width = 2, SSR.data = NULL)
```

Arguments

`y`	(Tx1)-vector of the dependent variable.
`x`	(Txk)-vector of the explanatory stochastic regressors.
`m`	Number of breaks.
`width`	Minimum spacing between the breaks.
`SSR.data`	Optional matrix of recursive SSR's.

Value

List of 2 elements: optimal SSR and the vector of break points.

References

Bai, Jushan, and Pierre Perron. "Computation and Analysis of Multiple Structural Change Models." Journal of Applied Econometrics 18, no. 1 (2003): 1–22. https://doi.org/10.1002/jae.659.

`segments.OLS.double` *Procedure to minimize the SSR for 2 break points*

Description

Procedure to minimize the SSR for 2 break points

Usage

```
segments.OLS.double(y, model)
```

Arguments

`y`	(Tx1)-vector of time series
`model`	**1** for the AA (without trend) model. **2** for the AA (with trend) model. **3** for the BB model. **4** for the CC model. **5** for the AC-CA model.

Details

See Carrion-i-Silvestre and Sansó (2007) for further details.

Value

List containing

resid (Tx1) vector of estimated OLS residuals.

tb1 The first break point.

tb2 The second break point.

References

Carrion-i-Silvestre, Josep Lluís, and Andreu Sansó. "The KPSS Test with Two Structural Breaks." Spanish Economic Review 9, no. 2 (May 16, 2007): 105–27. https://doi.org/10.1007/s10108-006-9017-8.

segments.OLS.single — *Procedure to minimize the SSR for 1 break point*

Description

Procedure to minimize the SSR for 1 break point

Usage

```
segments.OLS.single(beg, end, first.break, last.break, len, SSR.data)
```

Arguments

`beg`	Sample begin.
`end`	Sample end.
`first.break`	First possible break point.
`last.break`	Last possible break point.
`len`	Total number of observations.
`SSR.data`	The matrix of recursive SSR values.

Details

See Carrion-i-Silvestre and Sansó (2006) for further details.

Value

List containing

SSR Optimal SSR value.

break_point The point of possible break.

References

Carrion-i-Silvestre, Josep Lluís, and Andreu Sansó. "Testing the Null of Cointegration with Structural Breaks." Oxford Bulletin of Economics and Statistics 68, no. 5 (October 2006): 623–46. https://doi.org/10.1111/j.1468-0084.2006.00180.x.

SSR.matrix — *Pre-calculate matrix of recursive SSR values.*

Description

Pre-calculate matrix of recursive SSR values.

Usage

```
SSR.matrix(y, x, width = 2)
```

Arguments

`y`	Dependent variable.
`x`	Explanatory variables.
`width`	Minimum spacing between the breaks.

Value

The matrix of recursive SSR values.

`SSR.recursive` *Calculate SSR recursively*

Description

Calculate SSR recursively

Usage

```
SSR.recursive(y, x, beg, end, width = 2)
```

Arguments

`y`	(Tx1)-vector of the dependent variable.
`x`	(Txk)-vector of the explanatory stochastic regressors.
`beg`	The start of SSR calculating period.
`end`	The end of SSR calculating period.
`width`	Minimum spacing between the breaks.

Details

Based on Brown, Durbin and Evans (1975).

Value

The vector of calculated recursive SSR.

`STADF.test` *Supremum ADF test with time transformation.*

Description

Supremum ADF test with time transformation.

Usage

```
STADF.test(
  y,
  trim = 0.01 + 1.8/sqrt(length(y)),
  const = FALSE,
  omega.est = TRUE,
  h = "auto_CV",
  truncated = TRUE,
  is.reindex = TRUE,
  ksi.input = "auto",
  hc = 1,
  pc = 1,
  add.p.value = TRUE
)
```

Arguments

`y`	The input time series of interest.
`trim`	Trimming parameter to determine the lower and upper bounds.
`const`	Whether the constant needs to be included.
`omega.est`	Whether the variance of Nadaraya-Watson residuals should be used.
`truncated`	Whether the truncation of Nadaraya-Watson residuals is needed.
`is.reindex`	Whether the Cavaliere and Taylor (2008) time transformation is needed.
`ksi.input`	The value of the truncation parameter. Can be either 'auto' or the explicit numerical value. In the former case the numeric value is estimated.
`hc`	The scaling parameter for Nadaraya-Watson bandwidth.
`pc`	The scaling parameter for the estimated truncation parameter value.
`add.p.value`	Whether the p-value is to be returned. This argument is needed to suppress the calculation of p-values during the precalculation of tables needed for the p-values estimating.

References

Cavaliere, Giuseppe, and A. M. Robert Taylor. "Time-Transformed Unit Root Tests for Models with Non-Stationary Volatility." Journal of Time Series Analysis 29, no. 2 (March 2008): 300–330. https://doi.org/10.1111/j.1467-9892.2007.00557.x.

Kurozumi, Eiji, Anton Skrobotov, and Alexey Tsarev. "Time-Transformed Test for the Explosive Bubbles under Non-Stationary Volatility." arXiv, November 15, 2021. http://arxiv.org/abs/2012.13937.

supBZ.statistic *Calculate supBZ statistic*

Description

Calculate supBZ statistic

Usage

```
supBZ.statistic(
  y,
  trim = 0.01 + 1.8/sqrt(length(y)),
  sigma.sq = NULL,
  generalized = FALSE
)
```

Arguments

y	The series of interest.
trim	Trimming parameter to determine the lower and upper bounds.
sigma.sq	Local non-parametric estimates of variance. If 'NULL' they will be estimated via Nadaraya-Watson procedure.

References

Harvey, David I., Stephen J. Leybourne, and Yang Zu. "Testing Explosive Bubbles with Time-Varying Volatility." Econometric Reviews 38, no. 10 (November 26, 2019): 1131–51. https://doi.org/10.1080/07474938.2

supSBADF.statistic *Calculate superior sign-based SADF statistic.*

Description

Calculate superior sign-based SADF statistic.

Usage

```
supSBADF.statistic(y, trim = 0.01 + 1.8/sqrt(length(y)), generalized = FALSE)
```

References

Harvey, David I., Stephen J. Leybourne, and Yang Zu. "Sign-Based Unit Root Tests for Explosive Financial Bubbles in the Presence of Deterministically Time-Varying Volatility." Econometric Theory 36, no. 1 (February 2020): 122–69. https://doi.org/10.1017/S0266466619000057.

`weighted.GSADF.test` *Weighted generalized supremum ADF test.*

Description

Weighted generalized supremum ADF test.

Usage

```
weighted.GSADF.test(
  y,
  trim = 0.01 + 1.8/sqrt(length(y)),
  const = TRUE,
  alpha = 0.05,
  iter = 4 * 200,
  urs = TRUE,
  seed = round(10^4 * sd(y))
)
```

Arguments

`y`	The input time series of interest.
`trim`	Trimming parameter to determine the lower and upper bounds.
`const`	Whether the constant needs to be included.
`alpha`	The significance level of interest.
`iter`	The number of iterations.
`urs`	Use ‘union of rejections‘ strategy.
`seed`	The seed parameter for the random number generator.

References

Harvey, David I., Stephen J. Leybourne, and Yang Zu. “Testing Explosive Bubbles with Time-Varying Volatility.” Econometric Reviews 38, no. 10 (November 26, 2019): 1131–51. https://doi.org/10.1080/07474938.2

Kurozumi, Eiji, Anton Skrobotov, and Alexey Tsarev. “Time-Transformed Test for the Explosive Bubbles under Non-Stationary Volatility.” arXiv, November 15, 2021. http://arxiv.org/abs/2012.13937.

`weighted.SADF.test` *Weighted supremum ADF test.*

Description

Weighted supremum ADF test.

Usage

```
weighted.SADF.test(
  y,
  trim = 0.01 + 1.8/sqrt(length(y)),
  const = TRUE,
  alpha = 0.05,
  iter = 4 * 200,
  urs = TRUE,
  seed = round(10^4 * sd(y))
)
```

Arguments

`y`	The input time series of interest.
`trim`	Trimming parameter to determine the lower and upper bounds.
`const`	Whether the constant needs to be included.
`alpha`	The significance level of interest.
`iter`	The number of iterations.
`urs`	Use 'union of rejections' strategy.
`seed`	The seed parameter for the random number generator.

References

Harvey, David I., Stephen J. Leybourne, and Yang Zu. "Testing Explosive Bubbles with Time-Varying Volatility." Econometric Reviews 38, no. 10 (November 26, 2019): 1131–51. https://doi.org/10.1080/07474938.2

Index