

Package ‘breaktest’

September 13, 2022

Version 0.0.3

Date 2022-09-13

Title Set of unit root and cointegration tests

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

Maintainer Vadim Zyamalov <zyamalov@ranepa.ru>

Depends R (>= 3.5.0)

Encoding UTF-8

Imports doSNOW, foreach, stringr, Rfast

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>

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.1

NeedsCompilation no

R topics documented:

ACF	3
ADF.test	3
ADF.test.S	5
AR	6
coint.conf.sets	7
coint.test.GH	8
determinants.KPSS.1.break	9
determinants.KPSS.2.breaks	10
determinants.KPSS.N.breaks	10
detrend.recursively	11
DOLS.1.break	12
DOLS.N.breaks	13
DOLS.vars.N.breaks	14

eos.break.test	15
get.cv.coint.conf.sets	16
get.cv.KPSS.1p	16
get.cv.KPSS.2p	17
get.p.values.SADF	17
GLS	18
GLS.bt	18
info.criterion	19
KP	20
KPSS	20
KPSS.1.break	21
KPSS.1.break.unknown	22
KPSS.2.breaks	23
KPSS.2.breaks.unknown	24
KPSS.HLT	25
KPSS.N.breaks	25
KPSS.N.breaks.bootstrap	27
lagn	28
lr.var	29
MDF.CHLT	30
MDF.multiple	31
MDF.single	32
MZ.statistic	32
NW.estimation	33
NW.loocv	34
NW.volatility	34
OLS	35
print.sadf	36
PY.sequential	36
PY.single	37
reindex.CT	38
rescale.CPST	38
robust.tests.multiple	39
robust.tests.single	40
SADF.bootstrap.test	40
SADF.test	41
sb.GSADF.test	42
seasonal.dummies	43
segments.GLS	44
segments.OLS.1.break	45
segments.OLS.2.breaks	45
segments.OLS.N.breaks	46
select.lead.lag	47
SSR.matrix	48
SSR.recursive	48
STADF.test	49
supBZ.statistic	51
supSBADF.statistic	52
VECM.test	52
weighted.SADF.test	53

ACF

*Calculating ACF values***Description**

A simple auxiliary function providing the estimates of autocorrelations of orders from 0 to $N - 1$.

Usage

```
ACF(y)
```

Arguments

`y` An input time series of interest.

Details

The function is not intended to be used directly so it's not exported.

Value

The vector of ACF values of orders from 0 to $N - 1$. Due to the R's way of indexing the 1-st element is the autocorrelation of order 0, and the N-th value is the autocorrelation of order $N - 1$.

ADF.test

*A simple implementation of ADF test***Description**

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).

Usage

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

Arguments

<code>y</code>	The input time series of interest.
<code>const, trend</code>	Whether a constant and trend are to be included.
<code>max.lag</code>	Maximum lag number
<code>criterion</code>	A criterion used to select number of lags. If lag selection is not needed keep this NULL.
<code>modified.criterion</code>	Whether the unit-root test modification is needed.
<code>rescale.criterion</code>	Whether the rescaling informational criterion is needed. Designed to cope with heteroscedasticity in residuals.

Details

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**

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 nuisance parameters eliminated.

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	A series of interest.
const, trend	Whether the constant and trend are to be included.
c	A filtration parameter used to construct an autocorrelation coefficient.
gamma	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	A 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 modification is needed.
iter	The number of bootstrap iterations.

Details

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 MacKinnon-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>.

Elliott, Graham, Thomas J. Rothenberg, and James H. Stock. "Efficient Tests for an Autoregressive Unit Root." *Econometrica* 64, no. 4 (1996): 813–36. <https://doi.org/10.2307/2171846>.

AR	<i>Custom AR with extra information</i>
----	---

Description

Custom AR with extra information

Usage

```
AR(y, x, max.lag, criterion = "aic")
```

Arguments

y	Dependent variable.
x	Exogenous explanatory variables.
max.lag	The maximum number of lags.
criterion	A criterion for lag number estimation.

Details

The function is not intended to be used directly so it's not exported.

Value

A list of:

- beta: estimates of coefficients,
- residuals: estimated residuals,
- predict: forecasted values,
- t.beta: *t*-statistics for beta,
- lag: estimated number of lags.

coint.conf.sets*Confidence sets for the break date in cointegrating regressions*

Description

Confidence sets for the break date in cointegrating regressions

Usage

```
coint.conf.sets(  
  y,  
  trend = FALSE,  
  zb = NULL,  
  zf = NULL,  
  z.lead = NULL,  
  z.lag = NULL,  
  conf.level = 0.9,  
  trim = 0.05,  
  criterion = "bic"  
)
```

Arguments

y	A LHS variable of interest.
trend	Whether the trend is to be included.
zb	I(1) regressors with break.
zf	I(1) regressors without break.
z.lead, z.lag	Number of leads and lags of z regressors. If any is NULL then both are estimated using informational criterion.
conf.level	Confidence level to obtain appropriate critical values.
trim	The trimming parameter to find the lower and upper bounds of possible break date.
criterion	A criterion for lead and lag number estimation.

Value

A list of confidence sets.

References

Kurozumi, Eiji, and Anton Skrobotov. "Confidence Sets for the Break Date in Cointegrating Regressions." *Oxford Bulletin of Economics and Statistics* 80, no. 3 (2018): 514–35. <https://doi.org/10.1111/obes.12223>.

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.

Value

An object of type cointGH. It's a list of

- shift: shift type,
- Za: MZ_{α} statistic and c.v.,
- Zt: MZ_t statistic and c.v.,
- ADF: ADF statistic and c.v..

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\(99\)00168-7](https://doi.org/10.1016/0304-4076(99)00168-7).

determinants.KPSS.1.break

Construct determinant variables for [KPSS.1.break](#)

Description

Construct determinant variables for [KPSS.1.break](#)

Usage

```
determinants.KPSS.1.break(model, n.obs, break.point)
```

Arguments

model	A scalar equal to <ul style="list-style-type: none"> • 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.obs	Number of observations.
break.point	Break point.

Details

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

The function is not intended to be used directly so it’s not exported.

Value

Matrix of determinant variables.

determinants.KPSS.2.breaks

Construct determinant variables for [KPSS.2.breaks](#)

Description

Construct determinant variables for [KPSS.2.breaks](#)

Usage

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

Arguments

model	<p>A scalar equal to</p> <ul style="list-style-type: none"> • 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.obs	Number of observations.
break.point	Positions for the first and second structural breaks (relative to the origin which is 1).

Details

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

The function is not intended to be used directly so it's not exported.

Value

Matrix of deterministic terms.

determinants.KPSS.N.breaks

Deterministic terms for [KPSS.N.breaks](#)

Description

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

Usage

```

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

```

Arguments

<code>model</code>	A scalar or vector of <ul style="list-style-type: none"> • 1: for the break in const, • 2: for the break in trend, • 3: for the break in const and trend.
<code>n.obs</code>	Number of observations.
<code>break.point</code>	Array of structural breaks.
<code>const, trend</code>	Include constant and 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.

<code>detrend.recursively</code>	<i>Detrending the data recursively</i>
----------------------------------	--

Description

This procedure is aimed to provide a recursively detrended series. More or less classical approach of full-sample detrending may lead to the regressors correlated with the error term.

Usage

```
detrend.recursively(y, x, c, gamma, trim)
```

Arguments

<code>y</code>	The dependent (LHS) variable.
<code>x</code>	The matrix of explanatory (RHS) variables.
<code>c</code>	A filtration parameter used to construct an autocorrelation coefficient.
<code>gamma</code>	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 $1 + c^\gamma T^{-\gamma}$.
<code>trim</code>	The trimming parameter. It's used to find the minimum size of subsamples while calculating recursive estimates. The ending point of the subsample for the t is $\max(t, \text{trim} \times T)$.

Details

Elliott et al (1996) recommend using $c = -7$ for the model with only an intercept, and $c = -13.5$ for the model with a linear trend.

The function is not intended to be used directly so it's not exported.

Value

A detrended series.

References

Elliott, Graham, Thomas J. Rothenberg, and James H. Stock. "Efficient Tests for an Autoregressive Unit Root." *Econometrica* 64, no. 4 (1996): 813–36. <https://doi.org/10.2307/2171846>.

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>.

DOLS.1.break	<i>Estimating DOLS regression for multiple known break points</i>
--------------	---

Description

Estimating DOLS regression for multiple known break points

Usage

```
DOLS.1.break(y, x, model, break.point, k.lags, k.leads)
```

Arguments

- | | |
|-----------------|---|
| y | A dependent (LHS) variable. |
| x | A matrix of explanatory (RHS) variables. |
| model | See Carrion-i-Silvestre and Sansó (2006) <ul style="list-style-type: none">• 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 | A position of the break point. |
| k.lags, k.leads | A number of lags and leads in DOLS regression. |

Details

The function is not intended to be used directly so it's not exported.

Value

A list of:

- Estimates of coefficients,
- Estimates of residuals,
- A value of BIC,
- *t*-statistics for the estimates of coefficients.

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>.

DOLS.N.breaks

Estimating DOLS regression for multiple known break points

Description

Estimating DOLS regression for multiple known break points

Usage

```
DOLS.N.breaks(
  y,
  x,
  model,
  break.point,
  const = FALSE,
  trend = FALSE,
  k.lags,
  k.leads
)
```

Arguments

<code>y</code>	A dependent (LHS) variable.
<code>x</code>	A matrix of explanatory (RHS) variables.
<code>model</code>	A scalar or vector of break types: <ul style="list-style-type: none"> • 1: for the break in const. • 2: for the break in trend. • 3: for the break in const and trend.
<code>break.point</code>	An array of moments of structural breaks.
<code>const, trend</code>	Whether a constant or trend are to be included.
<code>k.lags, k.leads</code>	A number of lags and leads in DOLS regression.

Details

The function is not intended to be used directly so it's not exported.

Value

A list of:

- Estimates of coefficients,
- Estimates of residuals,
- A set of informational criterions values,
- t -statistics for the estimates of coefficients.

DOLS.vars.N.breaks	<i>Preparing variables for DOLS regression with multiple known break points</i>
--------------------	---

Description

Preparing variables for DOLS regression with multiple known break points

Usage

```
DOLS.vars.N.breaks(
  y,
  x,
  model,
  break.point,
  const = FALSE,
  trend = FALSE,
  k.lags,
  k.leads
)
```

Arguments

<code>y</code>	A dependent (LHS) variable.
<code>x</code>	A matrix of explanatory (RHS) variables.
<code>model</code>	A scalar or vector of <ul style="list-style-type: none"> • 1: for the break in const. • 2: for the break in trend. • 3: for the break in const and trend.
<code>break.point</code>	An array of moments of structural breaks.
<code>const, trend</code>	Whether a constant or trend are to be included.
<code>k.lags, k.leads</code>	A number of lags and leads in DOLS regression.

Details

The function is not intended to be used directly so it's not exported.

Value

A list of LHS and RHS variables.

eos.break.test	<i>Andrews-Kim (2006) test</i>
----------------	--------------------------------

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

A list of

- 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>

```
get.cv.coint.conf.sets
```

Critical values for break point confidence intervals

Description

Auxiliary function returning pre-calculated critical values for [coint.conf.sets](#)

Usage

```
get.cv.coint.conf.sets(lambda, trend, conf.level, p_zb, p_zf)
```

Arguments

<code>lambda</code>	Relative break point position.
<code>trend</code>	Whether there is to be included.
<code>conf.level</code>	Confidence level.
<code>p_zb</code>	Number of variables with breaks.
<code>p_zf</code>	Number of variables without breaks.

Details

The function is not intended to be used directly so it's not exported.

```
get.cv.KPSS.1p
```

Critical values for KPSS test with 1 break

Description

Auxiliary function returning pre-calculated critical values for [KPSS.1.break](#)

Usage

```
get.cv.KPSS.1p(model, break.point, n.obs, k)
```

Arguments

<code>model</code>	A scalar equal to <ul style="list-style-type: none"> • 1: for model An, • 2: for model A, • 3: for model B, • 4: for model C, • 5: for model D, • 6: for model E.
<code>break.point</code>	Position of the break point.
<code>n.obs</code>	Number of observations.
<code>k</code>	Number of RHS variables.

Details

The function is not intended to be used directly so it's not exported.

get.cv.KPSS.2p	<i>Critical values for KPSS test with 2 breaks</i>
----------------	--

Description

Auxiliary function returning pre-calculated critical values for [KPSS.2.breaks](#)

Usage

```
get.cv.KPSS.2p(model, break.point, n.obs)
```

Arguments

model	A scalar equal to <ul style="list-style-type: none"> • 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.
break.point	Position of the break point.
n.obs	Number of observations.

Details

The function is not intended to be used directly so it's not exported.

get.p.values.SADF	<i>Critical values for SADF-type tests</i>
-------------------	--

Description

Interpolating p-value for intermediate observation numbers for SADF-type tests.

Usage

```
get.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.

Details

The function is not intended to be used directly so it's not exported.

GLS	<i>Custom GLS with extra information</i>
-----	--

Description

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

Usage

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

Arguments

y	Dependent variable.
z	Explanatory variables.
c	Coefficient for ρ calculation.

Details

The function is not intended to be used directly so it's not exported.

Value

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

GLS.bt	<i>GLS filtering</i>
--------	----------------------

Description

GLS filtering

Usage

```
GLS.bt(y, lambda, c)
```

Arguments

y	Series of interest.
lambda	Break relative position.
c	Filtering parameter.

info.criterion	<i>Information criteria</i>
----------------	-----------------------------

Description

Information criteria

Usage

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

Arguments

resids	Input residuals needed for estimating the values of information criteria.
extra	Number of extra parameters needed for estimating the punishment term.
modification	Whether the unit-root test modification 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 criteria:

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

Value

The list of information criteria values.

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	<i>Kejrival-Perron procedure of breaks number detection</i>
----	---

Description

Kejrival-Perron procedure of breaks number detection

Usage

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

Arguments

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

Value

The estimated optimal break point.

References

Kejrival, 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	<i>Auxiliary function returning KPSS statistic value.</i>
------	---

Description

Auxiliary function returning KPSS statistic value.

Usage

KPSS(resids, variance)

Arguments

resids	The series of residuals.
variance	The value of the long-run variance.

Details

The function is not intended to be used directly so it's not exported.

KPSS.1.break	<i>KPSS-test with known structural break</i>
--------------	--

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	An input (LHS) time series of interest.
x	A matrix of (RHS) explanatory stochastic regressors.
model	A scalar equal to <ul style="list-style-type: none"> • 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 <ul style="list-style-type: none"> • 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

A list of:

- 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.

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>.

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	A scalar equal to <ul style="list-style-type: none"> • 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 <ul style="list-style-type: none"> • 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).

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>.

 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	An input (LHS) time series of interest.
model	A scalar equal to <ul style="list-style-type: none"> • 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.
break.point	Positions for the first and second structural breaks (relative to the origin which is 1).
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 <ul style="list-style-type: none"> • 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

A list of:

- beta: DOLS estimates of the coefficients regressors,
- tests: SC test (coinKPSS-test),
- resid: Residuals of the model,
- t.beta: *t*-statistics for beta,
- break_point: Break points.

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>.

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	A scalar equal to <ul style="list-style-type: none"> • 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 <ul style="list-style-type: none"> • 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.

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>.

KPSS.HLT	<i>Unit root testing procedure under a single structural break.</i>
----------	---

Description

Unit root testing procedure under a single structural break.

Usage

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

Arguments

y	A 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.

Value

The value of test statistic.

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	<i>KPSS-test with multiple known structural breaks</i>
---------------	--

Description

Procedure to compute the KPSS test with multiple known 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

<code>y</code>	An input (LHS) time series of interest.
<code>x</code>	A matrix of (RHS) explanatory stochastic regressors.
<code>model</code>	A scalar or vector of <ul style="list-style-type: none"> • 1: for the break in const, • 2: for the break in trend, • 3: for the break in const and trend.
<code>break.point</code>	Array of structural breaks.
<code>const, trend</code>	Whether a constant or trend should be included.
<code>weakly.exog</code>	Boolean where we specify whether the stochastic regressors are exogenous or not <ul style="list-style-type: none"> • TRUE: if the regressors are weakly exogenous, • FALSE: if the regressors are not weakly exogenous (DOLS is used in this case).
<code>lags.init, leads.init</code>	Scalars defining the initial number of lags and leads for DOLS.
<code>max.lag</code>	scalar, with the maximum order of the parametric correction. The final order of the parametric correction is selected using the BIC information criterion.
<code>kernel</code>	Kernel for calculating long-run variance <ul style="list-style-type: none"> • <code>bartlett</code>: for Bartlett kernel, • <code>quadratic</code>: for Quadratic Spectral kernel, • <code>NULL</code> for the Kurozumi's proposal, using Bartlett kernel.
<code>criterion</code>	Information criterion for DOLS lags and leads selection: <code>aic</code> , <code>bic</code> , <code>hq</code> , or <code>lwz</code> .

Value

A list of

- `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.

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>.
- 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>.

 KPSS.N.breaks.bootstrap

KPSS-test with multiple unknown structural breaks

Description

Procedure to compute the KPSS test with multiple unknown 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

<code>y</code>	An input (LHS) time series of interest.
<code>x</code>	A matrix of (RHS) explanatory stochastic regressors.
<code>model</code>	A scalar or vector of <ul style="list-style-type: none"> • 1: for the break in const, • 2: for the break in trend, • 3: for the break in const and trend.
<code>break.point</code>	Array of structural breaks.
<code>const</code>	Include constant if TRUE .
<code>trend</code>	Include trend if TRUE .
<code>weakly.exog</code>	Boolean where we specify whether the stochastic regressors are exogenous or not <ul style="list-style-type: none"> • TRUE: if the regressors are weakly exogenous, • FALSE: if the regressors are not weakly exogenous (DOLS is used in this case).
<code>lags.init, leads.init</code>	Scalars defininig the initial number of lags and leads for DOLS.
<code>max.lag</code>	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 <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • "sample": sampling from residuals with replacement, • "Cavalieri-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.

Value

A list of:

- test: The value of KPSS test statistic,
- p.value: The estimates p-value,
- bootstrapped: Bootstrapped auxiliary statistics.

lagn	<i>Produce a vector lagged backward of forward</i>
------	--

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).
na	Value to fill missing observations, NA by default.

Details

The function is not intended to be used directly so it's not exported.

Value

Lagged or leaded vector.

lr.var

*Calculating long-run variance or covariance matrix***Description**

Calculating long-run variance or covariance matrix

Usage

```

lr.var(
  y,
  demean = TRUE,
  kernel = "bartlett",
  limit.lags = FALSE,
  limit.selector = "kpss-q",
  upper.rho.limit = 0.97,
  upper.lag.limit = 0.8,
  recolor = FALSE,
  max.lag = 0,
  criterion = "bic"
)

lr.var.bartlett(y)

lr.var.quadratic(y)

lr.var.bartlett.AK(y)

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

```

Arguments

y	A series of interest.
demean	Whether the demeaning is needed.
kernel	A kernel to be used: <ul style="list-style-type: none"> • truncated: $\begin{cases} 1 & x \leq 1 \\ 0 & \text{otherwise} \end{cases}$ • bartlett: $\begin{cases} 1 - x & x \leq 1 \\ 0 & \text{otherwise} \end{cases}$ • parzen: $\begin{cases} 1 - 6x^2 + 6 x ^3 & x \leq 1/2 \\ 2(1 - x)^3 & 1/2 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases}$ • tukey-hanning: $\begin{cases} (1 + \cos(\pi x))/2 & x \leq 1 \\ 0 & \text{otherwise} \end{cases}$ • quadratic: $\frac{25}{12\pi^2 x^2} \left(\frac{\sin(6\pi x/5)}{6\pi x/5} - \cos(6\pi x/5) \right)$
limit.lags	Whether all lags should be used in formulae.
limit.selector	Way of limit selection: <ul style="list-style-type: none"> • kpss-q: $4(T/100)^{1/4}$.

	<ul style="list-style-type: none"> • <code>kps-m</code>: $12(T/100)^{1/4}$. • <code>Andrews</code>: kernel-specific formula from Andrews (1991). • <code>Kurozumi</code>: kernel-specific formula from Andrews (1991) with Kurozumi (2002) proposal.
<code>upper.rho.limit</code>	The upper limit for the value or AR-coefficient.
<code>upper.lag.limit</code>	The value used to calculate the upper limit for Kurozumi (2002) proposal.
<code>recolor</code>	Whether the correction by Sul et al. (2005) should be used. This option resets <code>limit.lags</code> to TRUE, and <code>limit.selector</code> to Andrews.
<code>max.lag</code>	Maximum number of lags used in AR regression during recolorization. Otherwise ignored.
<code>criterion</code>	The information criterion: <code>bic</code> , <code>aic</code> or <code>lwz</code> .

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](https://doi.org/10.1016/S0304-4076(01)00106-3).
- Sul, Donggyu, Peter C. B. Phillips, and Chi-Young Choi. “Prewhitening Bias in HAC Estimation.” *Oxford Bulletin of Economics and Statistics* 67, no. 4 (August 2005): 517–46. <https://doi.org/10.1111/j.1468-0084.2005.00130.x>.

MDF.CHLT

MDF test for a single break and possible heteroscedasticity

Description

MDF test for a single break and possible heteroscedasticity

Usage

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

Arguments

<code>y</code>	A series of interest
<code>max.lag</code>	The maximum possible lag.
<code>trim</code>	Trimming parameter for lag selection
<code>iter</code>	Number of bootstrap iterations.

Value

An object of type `mdfCHLT`. It's a list of four sublists each containing:

- The value of MZ_α , MSB , MZ_t , or ADF ,
- The asymptotic c.v.,
- The bootstrapped c.v.

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	<i>MDF procedure for multiple unknown breaks.</i>
--------------	---

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

Value

A list of sublists each containing

- The value of statistic: $MDF - GLS$, $MDF - OLS$,
- The asymptotic critical values. UR values are included as well.

MDF.single	<i>MDF procedure for a single unknown break.</i>
------------	--

Description

MDF procedure for a single unknown break.

Usage

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

Arguments

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

Value

A list of sublists each containing

- The value of statistic: $MDF - GLS$, $MDF - OLS$,
- The asymptotic critical values. UR values are included as well.

MZ.statistic	<i>Calculating M-statistics by Stock (1990) and Perron and Ng (1996).</i>
--------------	---

Description

Calculating M-statistics by Stock (1990) and Perron and Ng (1996).

Usage

```
MZ.statistic(y, l, const = FALSE, trend = FALSE)
```

Arguments

y	A series of interest.
l	Number of lags for inner ADF test.
const, trend	Whether a constant and trend are to be included.

Details

The function is not intended to be used directly so it's not exported.

Value

List of values of MZ_α , MZ_t and MSB statistics.

References

Perron, Pierre, and Serena Ng. "Useful Modifications to Some Unit Root Tests with Dependent Errors and Their Local Asymptotic Properties." *The Review of Economic Studies* 63, no. 3 (July 1, 1996): 435–63. <https://doi.org/10.2307/2297890>.

Stock, James H. "A Class of Tests for Integration and Cointegration." Kennedy School of Government, Harvard University, 1990.

NW.estimation

Nadaraya–Watson kernel regression.

Description

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.

Details

The function is not intended to be used directly so it's not exported.

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	<i>LOO-CV for h in Nadaraya–Watson kernel regression.</i>
----------	---

Description

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.

Details

The function is not intended to be used directly so it's not exported.

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	<i>NW.volatility - Nadaraya–Watson kernel volatility estimation</i>
---------------	---

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.

Details

The function is not intended to be used directly so it's not exported.

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>.

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.

OLS	<i>Custom OLS with extra information</i>
-----	--

Description

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

Usage

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

Arguments

y	Dependent variable.
x	Explanatory variables.

Details

The function is not intended to be used directly so it's not exported.

Value

The list of:

- beta: estimates of coefficients,
- resid: estimated residuals,
- predict: forecasted values,
- t.beta: *t*-statistics for beta.

<code>print.sadf</code>	<i>Custom functions for printing results in a nice way.</i>
-------------------------	---

Description

Custom functions for printing results in a nice way.

Usage

```
## S3 method for class 'sadf'
print(x, ...)

## S3 method for class 'mdfHLT'
print(x, ...)

## S3 method for class 'mdfHLTN'
print(x, ...)

## S3 method for class 'mdfCHLT'
print(x, ...)

## S3 method for class 'cointGH'
print(x, ...)
```

Arguments

<code>x</code>	Object containing results.
<code>...</code>	Any additional arguments for print function.

<code>PY.sequential</code>	<i>Sequential Perron-Yabu (2009) statistic for breaks at unknown date.</i>
----------------------------	--

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.

Value

The estimated Wald statistic.

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, trend	Allowing the break in constant or 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.

Value

A list of the estimated Wald statistic as well as its c.v.

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.CT

A function that makes reindexing

Description

The function is aimed to calculate the sequence of indices providing a new "time transformed" time series as in Cavaliere and Taylor (2008).

Usage

```
reindex.CT(u)
```

Arguments

u The residuals series for reindexing.

Details

The function is not intended to be used directly so it's not exported.

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 Bubbles under Non-Stationary Volatility." *Journal of Financial Econometrics*, April 23, 2022. <https://doi.org/10.1093/jjfinec/nbac004>.

rescale.CPST

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

Description

This rescaling procedure is needed to cope with possible heteroscedasticity in the data. Simply it's achieved by taking a cumulative sum of the first difference normalized by the non-parametric local estimate of the variance.

Usage

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

Arguments

<code>d.y</code>	A series of first differences.
<code>x</code>	A matrix of ADF RHS variables.
<code>deter</code>	A matrix of deterministic variables for detrending.
<code>adf.lag</code>	A lag of the corresponding ADF model.
<code>max.lag</code>	The maximum possible lag.

Details

The function is not intended to be used directly so it's not exported.

Value

A rescaled series.

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>.

`robust.tests.multiple` A wrapping function around *KP* and *MDF.multiple*.

Description

A wrapping function around *KP* and *MDF.multiple*.

Usage

```
robust.tests.multiple(
  y,
  const = FALSE,
  season = FALSE,
  breaks = 2,
  trim = 0.15
)
```

Arguments

<code>y</code>	A series of interest.
<code>const</code>	Whether the constant term should be included.
<code>season</code>	Whether the seasonal adjustment is needed.
<code>breaks</code>	Number of breaks.
<code>trim</code>	Trimming value for a possible break date bounds.

`robust.tests.single` *A wrapping function around [MDF.single](#).*

Description

A wrapping function around [MDF.single](#).

Usage

```
robust.tests.single(
  y,
  const = FALSE,
  trend = FALSE,
  season = FALSE,
  trim = 0.15
)
```

Arguments

<code>y</code>	A series of interest.
<code>const</code>	Whether the constant term should be included.
<code>trend</code>	Whether the trend term should be included.
<code>season</code>	Whether the seasonal adjustment is needed.
<code>trim</code>	Trimming value for a possible break date bounds.

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

Description

`SADF.bootstrap.test` is a wild bootstrapping procedure for estimating critical and p -values for [SADF.test](#).

`GSADF.bootstrap.test` is the same procedure but for [GSADF.test](#).

Usage

```
SADF.bootstrap.test(
  y,
  trim = 0.01 + 1.8/sqrt(length(y)),
  const = TRUE,
  alpha = 0.05,
  iter = 999,
  seed = round(10^4 * sd(y))
)
```

```
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	An 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.

Value

An object of type `sadf`. It's a list of:

- y,
- trim,
- const,
- alpha,
- iter,
- seed,
- vector of t -values,
- the value of the corresponding test statistic,
- series of bootstrapped test statistics,
- bootstrapped critical values,
- p -value.

References

Kurozumi, Eiji, Anton Skrobotov, and Alexey Tsarev. "Time-Transformed Test for Bubbles under Non-Stationary Volatility." *Journal of Financial Econometrics*, April 23, 2022. <https://doi.org/10.1093/jfinec/nbac004>.

SADF.test

Supremum ADF tests

Description

SADF.test is a test statistic equal to the minimum value of [ADF.test](#) for subsamples starting at $t = 1$.

GSADF.test is a generalized version of SADF.test. Subsamples are allowed to start at any point between 1 and $T(1 - trim)$.

Usage

```

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

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.

Value

An object of type `sadf`. It's a list of:

- y,
- trim,
- const,
- vector of *t*-values,
- the value of the corresponding test statistic,
- *p*-value if it was asked for.

References

Kurozumi, Eiji, Anton Skrobotov, and Alexey Tsarev. "Time-Transformed Test for Bubbles under Non-Stationary Volatility." *Journal of Financial Econometrics*, April 23, 2022. <https://doi.org/10.1093/jfinec/nbac004>.

sb.GSADF.test

Sign-based SADF test

Description

Sign-based SADF test

Usage

```
sb.GSADF.test(
  y,
  trim = 0.01 + 1.8/sqrt(length(y)),
  const = TRUE,
  alpha = 0.05,
  iter = 999,
  urs = TRUE,
  seed = round(10^4 * sd(y))
)
```

Arguments

y	A series of interest.
trim	Trimming parameter to determine the lower and upper bounds.
const	Whether the constant needs to be included.
alpha	Needed level of significance.
iter	Number of bootstrapping iterations.
urs	Use union of rejections strategy if TRUE.
seed	The seed parameter for the random number generator.

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>.

seasonal.dummies	<i>Generating monthly seasonal dummy variables</i>
------------------	--

Description

Generating monthly seasonal dummy variables

Usage

```
seasonal.dummies(n.obs)
```

Arguments

n.obs	number of observations.
-------	-------------------------

Details

The function is not intended to be used directly so it's not exported.

Value

The matrix of values of 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.

Details

The function is not intended to be used directly so it's not exported.

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.1.break *Procedure to minimize the SSR for 1 break point*

Description

Procedure to minimize the SSR for 1 break point

Usage

```
segments.OLS.1.break(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

The function is not intended to be used directly so it's not exported.

Value

A list of:

- 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>.

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

Description

Procedure to minimize the SSR for 2 break points

Usage

```
segments.OLS.2.breaks(y, model)
```

Arguments

y	(Tx1)-vector of time series
model	A scalar equal to <ul style="list-style-type: none"> • 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

The function is not intended to be used directly so it's not exported.

Value

A list of

- 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.N.breaks *Find $m + 1$ optimal partitions*

Description

Find $m + 1$ optimal partitions

Usage

```
segments.OLS.N.breaks(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.

Details

The function is not intended to be used directly so it's not exported.

Value

A list of:

- optimal SSR,
- 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>.

select.lead.lag

Estimating optimal number of leads and lags

Description

Estimating optimal number of leads and lags

Usage

```
select.lead.lag(
  y,
  trend = TRUE,
  zb = NULL,
  zf = NULL,
  trim = 0.05,
  criterion = "bic"
)
```

Arguments

y	LHS dependent variable.
trend	Whether the trend is to be included.
zb	I(1) regressors with break.
zf	I(1) regressors without break.
trim	The trimming parameter to find the lower and upper bounds of possible break date.
criterion	A criterion for lead and lag number estimation.

Details

The function is not intended to be used directly so it's not exported.

Value

A list of estimated values of leads and lags.

References

Kurozumi, Eiji, and Anton Skrobotov. “Confidence Sets for the Break Date in Cointegrating Regressions.” *Oxford Bulletin of Economics and Statistics* 80, no. 3 (2018): 514–35. <https://doi.org/10.1111/obes.12223>.

SSR.matrix	<i>Pre-calculate matrix of recursive SSR values.</i>
------------	--

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.

Details

The function is not intended to be used directly so it's not exported.

Value

The matrix of recursive SSR values.

SSR.recursive	<i>Calculate SSR recursively</i>
---------------	----------------------------------

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, end	The start and the end of SSR calculating period.
width	Minimum spacing between the breaks.

Details

The function is not intended to be used directly so it's not exported.

Value

The vector of calculated recursive SSR.

References

Brown, R. L., J. Durbin, and J. M. Evans. "Techniques for Testing the Constancy of Regression Relationships over Time." *Journal of the Royal Statistical Society. Series B (Methodological)* 37, no. 2 (1975): 149–92.

STADF.test

Supremum ADF tests with time transformation

Description

See [SADF.test](#). Tests with time transformation are the modified versions of the ordinary SADF and GSADF tests using Nadaraya-Watson residuals and reindexing procedure by Cavaliere-Taylor (2008).

Usage

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

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

Arguments

y	An 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.

<code>ksi.input</code>	The value of the truncation parameter. Can be either auto or the explicit numerical value. In the former case the numeric value is estimated.
<code>hc</code>	The scaling parameter for Nadaraya-Watson bandwidth.
<code>pc</code>	The scaling parameter for the estimated truncation parameter value.
<code>add.p.value</code>	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.

Value

An object of type `sadf`. It's a list of:

- `y`,
- `N`: Number of observations,
- `trim`,
- `const`,
- `omega.est`,
- `truncated`,
- `is.reindex`,
- `new.index`: the vector of new indices,
- `ksi.input`,
- `hc`,
- `h.est`,
- `u.hat`,
- `pc`,
- `w.sq`,
- `t.values`: vector of t -values,
- the value of the corresponding test statistic,
- `u.hat.truncated`: truncated residuals if truncation was asked for,
- `ksi, sigma`: estimated values of the truncation parameter and resulting s.e. if `ksi.input` equals `auto`,
- `eta.hat`: the values of reindexing function if reindexing was asked for,
- p -value if it was asked for.

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 Bubbles under Non-Stationary Volatility." *Journal of Financial Econometrics*, April 23, 2022. <https://doi.org/10.1093/jfinec/nbac004>.

supBZ.statistic	<i>Calculate supBZ statistic</i>
-----------------	----------------------------------

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.
generalized	Whether to calculate generalized statistic value.

Details

The function is not intended to be used directly so it's not exported.

Value

A list of:

- y,
- trim,
- sigma.sq,
- BZ.values: a series of BZ-statistic,
- supBZ.value: the maximum of supBZ.values,
- h.est: the estimated value of bandwidth if sigma.sq is NULL.

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/10.1080/07474938.2019.1644444>

supSBADF.statistic	<i>Calculate superior sign-based SADF statistic.</i>
--------------------	--

Description

Calculate superior sign-based SADF statistic.

Usage

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

Arguments

y	The series of interest.
trim	Trimming parameter to determine the lower and upper bounds.
generalized	Whether to calculate generalized statistic value.

Details

The function is not intended to be used directly so it's not exported.

Value

A list of

- y,
- trim,
- C.t: the cumulative sum of "signs" (1 or -1) of the first difference of y,
- SBADF.values: series of sign-based ADF statistics,
- supSBADF.value: the maximum of SBADF.values.

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>.

VECM.test	<i>Test of the co-integration rank with a possible break in trend</i>
-----------	---

Description

This procedure is aimed on the problem of testing for the co-integration rank of a vector autoregressive process in the case where a trend break may potentially be present in the data.

The test is based on estimating the quasi log likelihood for two situations, with break, and without it. The one with the smallest value is considered to be the result.

Usage

```
VECM.test(y, r, max.lag, trim = 0.15)
```

Arguments

<code>y</code>	The matrix of n VAR variables.
<code>r</code>	The co-integration rank tested against the alternative of n .
<code>max.lag</code>	The maximum number of lags.
<code>trim</code>	The trimming parameter to determine the lower and upper bounds.

Value

A list of:

- the indicator of the rejection of null.
- the estimated break point.
- the estimated lag number.

References

Harris, David, Stephen J. Leybourne, and A. M. Robert Taylor. "Tests of the Co-Integration Rank in VAR Models in the Presence of a Possible Break in Trend at an Unknown Point." *Journal of Econometrics, Innovations in Multiple Time Series Analysis*, 192, no. 2 (June 1, 2016): 451–67. <https://doi.org/10.1016/j.jeconom.2016.02.010>.

<code>weighted.SADF.test</code>	<i>Weighted supremum ADF test</i>
---------------------------------	-----------------------------------

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))
)

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

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

Value

An object of type `sadf`. It's a list of:

- `y`,
- `trim`,
- `const`,
- `alpha`,
- `iter`,
- `urs`,
- `seed`,
- `sigma.sq`: the estimated variance,
- `BZ.values`: a series of BZ-statistic,
- `supBZ.value`: the maximum of `supBZ.values`,
- `supBZ.bootstrap.values`: bootstrapped supremum BZ values,
- `supBZ.cr.value`: supremum BZ α critical value,
- `p.value`,
- `is.explosive`: 1 if `supBZ.value` is greater than `supBZ.cr.value`.

if `urs` is TRUE the following items are also included:

- vector of t -values,
- the value of the SADF test statistic,
- `SADF.bootstrap.values`: bootstrapped SADF values,
- `U.value`: union test statistic value,
- `U.bootstrap.values`: bootstrapped series of `U.value`,
- `U.cr.value`: critical value of `U.value`.

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/10.1080/07474938.2019.1644444>
- Kurozumi, Eiji, Anton Skrobotov, and Alexey Tsarev. "Time-Transformed Test for Bubbles under Non-Stationary Volatility." *Journal of Financial Econometrics*, April 23, 2022. <https://doi.org/10.1093/jfinec/nbac004>.

Index

ACF, [3](#)
ADF.test, [3](#), [41](#)
ADF.test.S, [5](#)
AR, [6](#)

coint.conf.sets, [7](#), [16](#)
coint.test.GH, [8](#)

determinants.KPSS.1.break, [9](#)
determinants.KPSS.2.breaks, [10](#)
determinants.KPSS.N.breaks, [10](#)
detrend.recursively, [11](#)
DOLS.1.break, [12](#)
DOLS.N.breaks, [13](#)
DOLS.vars.N.breaks, [14](#)

eos.break.test, [15](#)

get.cv.coint.conf.sets, [16](#)
get.cv.KPSS.1p, [16](#)
get.cv.KPSS.2p, [17](#)
get.p.values.SADF, [17](#)
GLS, [18](#)
GLS.bt, [18](#)
GSADF.bootstrap.test
 (SADF.bootstrap.test), [40](#)
GSADF.test, [40](#)
GSADF.test (SADF.test), [41](#)
GSTADF.test (STADF.test), [49](#)

info.criterion, [19](#)

KP, [20](#), [39](#)
KPSS, [20](#)
KPSS.1.break, [9](#), [16](#), [21](#)
KPSS.1.break.unknown, [22](#)
KPSS.2.breaks, [10](#), [17](#), [23](#)
KPSS.2.breaks.unknown, [24](#)
KPSS.HLT, [25](#)
KPSS.N.breaks, [10](#), [25](#)
KPSS.N.breaks.bootstrap, [27](#)

lagn, [28](#)
lr.var, [29](#)

MDF.CHLT, [30](#)
MDF.multiple, [31](#), [39](#)
MDF.single, [32](#), [40](#)
MZ.statistic, [32](#)

NW.estimation, [33](#)
NW.loocv, [34](#)
NW.volatility, [34](#)

OLS, [35](#)

print, [36](#)
print.cointGH(print.sadf), [36](#)
print.mdfCHLT(print.sadf), [36](#)
print.mdfHLT(print.sadf), [36](#)
print.mdfHLTN(print.sadf), [36](#)
print.sadf, [36](#)
PY.sequential, [36](#)
PY.single, [37](#)

reindex.CT, [38](#)
rescale.CPST, [38](#)
robust.tests.multiple, [39](#)
robust.tests.single, [40](#)

SADF.bootstrap.test, [40](#)
SADF.test, [40](#), [41](#), [49](#)
sb.GSADF.test, [42](#)
seasonal.dummies, [43](#)
segments.GLS, [44](#)
segments.OLS.1.break, [45](#)
segments.OLS.2.breaks, [45](#)
segments.OLS.N.breaks, [46](#)
select.lead.lag, [47](#)
SSR.matrix, [48](#)
SSR.recursive, [48](#)
STADF.test, [49](#)
supBZ.statistic, [51](#)
supSBADF.statistic, [52](#)

VECM.test, [52](#)

weighted.GSADF.test
 (weighted.SADF.test), [53](#)
weighted.SADF.test, [53](#)