Package ‘mlrv’

July 30, 2024

Type Package

Title Long-Run Variance Estimation in Time Series Regression

Version 0.1.2

Description Plug-in and difference-based long-run covariance matrix estimation for time series regression. Two applications of hypothesis testing are also provided. The first one is for testing for structural stability in coefficient functions. The second one is aimed at detecting long memory in time series regression. Lujia Bai and Weichi Wu (2024)<doi:10.3150/23-BEJ1680> Zhou Zhou and Wei Biao Wu(2010)<doi:10.1111/j.1467-9868.2010.00743.x> Jianqing Fan and Wenyang Zhang<doi:10.1214/aos/1017939139> Lujia Bai and Weichi Wu(2024)<doi:10.1093/biomet/asae013> Dimitris N. Politis, Joseph P. Romano, Michael Wolf(1999)<doi:10.1007/978-1-4612-1554-7> Weichi Wu and Zhou Zhou(2018)<doi:10.1214/17-AOS1582>.

License MIT + file LICENSE

Depends R (>= 3.6.0)

Encoding UTF-8

LazyData true

Imports Rcpp, numDeriv, magrittr, foreach, doParallel, RcppArmadillo, mathjaxr, xtable, stats

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.3.2

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

VignetteBuilder knitr

RdMacros mathjaxr

Config/testthat/edition 3

Language en-US

NeedsCompilation yes

Author Lujia Bai [aut, cre],
Weichi Wu [ctb]

Maintainer Lujia Bai <bailujia98@gmail.com>

Repository CRAN

Date/Publication 2024-07-30 14:20:02 UTC

Contents

`bregress2` *Simulate data from time-varying time series regression model with change points*

Description

Simulate data from time-varying time series regression model with change points

Usage

```
bregress2(nn, cp = 0, delta = 0, type = "norm")
```

Arguments

`nn`	sample size
`cp`	number of change points. If cp is between 0 and 1, it specifies the location of the single change point
`delta`	double, magnitude of the jump
`type`	type of distributions of the innovations, default normal. It can also be "t4", "t5" and "t6".

Value

a list of data, x covariates, y response and e error. n = 300 data = bregress2(n, 2, 1) # time series regression model with 2 changes points gcv_cov *Generalized Cross Validation*

Description

Given a bandwidth, compute its corresponding GCV value

Usage

```
gcv_cov(bw, t, y, X, verbose = 1L)
```

Arguments

bw	double, bandwidth
t	vector, scaled time $[0, 1]$
y	vector, response
X	matrix, covariates matrix
verbose	bool, whether to print the numerator and denominator in GCV value

Details

Generalized cross validation value is defined as

$$n^{-1}|Y - \hat{Y}|^2/[1 - \mathrm{tr}(Q(b))/n]^2$$

When computing $\mathrm{tr}(Q(b))$, we use the fact that the first derivative of coefficient function is zero at central point The ith diagonal value of $Q(b)$ is actually $x^T(t_i)S_n^{-1}x(t_i)$ where S_n^{-1} means the top left p-dimension square matrix of $S_n(t_i) = X^T W(t_i)X$, $W(t_i)$ is the kernel weighted matrix. Details on the computation of S_n could be found in `LocLinear` and its reference

Value

GCV value

Examples

```
param = list(d = -0.2, heter = 2, tvd = 0,
 tw = 0.8, rate = 0.1, cur = 1, center = 0.3,
  ma_rate =  0, cov_tw =  0.2, cov_rate = 0.1,
   cov_center = 0.1, all_tw  = 1, cov_trend = 0.7)
data = Qct_reg(1000, param)
value <- gcv_cov(0.2, (1:1000)/1000, data$y, data$x)
```

heter_covariate	*Long memory tests for non-stationary time series regression*

Description

Test for long memory of e_i in the time series regression

$$y_i = x_i\beta_i + e_i, 1 \leq i \leq n$$

where x_i is the multivariate covariate process with first component 1, β_i is the functional coefficient, e_i is the error term which can be long memory. In particular,covariates and the error term are allowed to be dependent.

Usage

```
heter_covariate(
  data,
 param = list(B = 2000, lrvmethod = 1, gcv = 1, neighbour = 1, lb = 3, ub = 11, tau_n =
    0.3, type = "KPSS"),
  mvselect = -1,
  bw = 0.2,
  shift = 1,
  verbose_dist = FALSE,
  hyper = FALSE
)
```

Arguments

`data`	a list with the vector y and the matrix x, for example, list(x=...,y=...).
`param`	a list of parameters, list(B =..., lrvmethod =...,gcv = ..., neighbour =..., lb = ..., ub = ..., tau_n = ..., type = ..., ind = ...)
`mvselect`	the value of moving window parameter m. In addition, mvselect=-1 provides data-driven smoothing parameters via Minimum Volatility of the long-run covariance estimator as proposed in Chapter 9 of Politis et al.(1999), while mvselect = -2 provides data-driven smoothing parameters via Minimum Volatility of the bootstrap statistics, see Bai and Wu (2024a).
`bw`	the bandwidth parameter in the local linear regression, default 0.2.
`shift`	modify bw by a factor, default 1.
`verbose_dist`	whether to print intermediate results, i.e., the bootstrap distribution and statistics, default FALSE.
`hyper`	whether to only print the selected values of the smoothing parameters,m and τ_n, default FALSE.

Details

param

- B, the number of bootstrap simulation, say 2000 *lrvmethod, the method of long-run variance estimation, lrvmethod = 0 uses the plug-in estimator in Zhou (2010), lrvmethod = 1 offers the debias difference-based estimator in Bai and Wu (2024b), lrvmethod = 2 provides the plug-in estimator using the $\breve{\beta}$, the pilot estimator proposed in Bai and Wu (2024b)
- gcv, 1 or 0, whether to use Generalized Cross Validation for the selection of b, the bandwidth parameter in the local linear regression
- neighbour, the number of neighbours in the extended minimum volatility, for example 1,2 or 3
- lb, the lower bound of the range of m in the extended minimum volatility Selection
- ub, the upper bound of the range of m in the extended minimum volatility Selection
- bw_set, the proposed grid of the range of bandwidth selection. if not presented, a rule of thumb method will be used for the data-driven range
- tau_n, the value of τ when no data-driven selection is used. if τ is set to 0, the rule of thumb $n^{-2/15}$ will be used
- type, c("KPSS","RS","VS","KS") type of tests, see Bai and Wu (2024a).
- ind, types of kernels
- 1 Triangular $1-|u|$, $u \le 1$
- 2 Epanechnikov kernel $3/4(1-u^2)$, $u \le 1$
- 3 Quartic $15/16(1-u^2)^2$, $u \le 1$
- 4 Triweight $35/32(1-u^2)^3$, $u \le 1$
- 5 Tricube $70/81(1-|u|^3)^3$, $u \le 1$

Value

p-value of the long memory test

mlrv functions

Heter_LRV, heter_covariate, heter_gradient, gcv_cov, MV_critical

References

Bai, L., & Wu, W. (2024a). Detecting long-range dependence for time-varying linear models. Bernoulli, 30(3), 2450-2474.

Bai, L., & Wu, W. (2024b). Difference-based covariance matrix estimation in time series nonparametric regression with application to specification tests. Biometrika, asae013.

Zhou, Z. and Wu, W. B. (2010). Simultaneous inference of linear models with time varying coefficients.J. R. Stat. Soc. Ser. B. Stat. Methodol., 72(4):513–531.

Politis, D. N., Romano, J. P., and Wolf, M. (1999). Subsampling. Springer Science & Business Media.

Examples

```
param = list(d = -0.2, heter = 2, tvd = 0,
 tw = 0.8, rate = 0.1, cur = 1,
 center = 0.3, ma_rate =  0, cov_tw =  0.2,
 cov_rate = 0.1, cov_center = 0.1, all_tw  = 1, cov_trend = 0.7)
data = Qct_reg(1000, param)
### KPSS test B
heter_covariate(data, list(B=20, lrvmethod = 1,
gcv = 1, neighbour = 1, lb = 3, ub = 11, type = "KPSS"), mvselect = -2, verbose_dist = TRUE)
```

`heter_gradient` *Structural stability tests for non-stationary time series regression*

Description

Test for long memory of e_i in the time series regression

$$y_i = x_i\beta_i + e_i, 1 \leq i \leq n$$

where x_i is the multivariate covariate process with first component 1, β_i is the coefficient, e_i is the error term which can be long memory. The goal is to test whether the null hypothesis

$$\beta_1 = \ldots = \beta_n = \beta$$

holds. The alternative hypothesis is that the coefficient function β_i is time-varying. Covariates and the error term are allowed to be dependent.

Usage

```
heter_gradient(data, param, mvselect = -1, verbose_dist = FALSE, hyper = FALSE)
```

Arguments

`data`	a list with the vector y (response) and the matrix x (covariates), for example, list(x=...,y=...).
`param`	a list of parameters, list(B =..., lrvmethod =...,gcv = ..., neighbour =..., lb = ..., ub = ..., tau_n = ..., type = ..., ind = ...)
`mvselect`	the value of moving window parameter m. In addition, mvselect=-1 provides data-driven smoothing parameters via Minimum Volatility of the long-run covariance estimator, while mvselect = -2 provides data-driven smoothing parameters via Minimum Volatility of the bootstrap statistics.
`verbose_dist`	whether to print intermediate results, i.e., the bootstrap distribution and statistics, default FALSE.
`hyper`	whether to only print the selected values of the smoothing parameters,m and τ_n, default FALSE.

Details

param

- B, the number of bootstrap simulation, say 2000
- lrvmethod the method of long-run variance estimation, lrvmethod = -1 uses the ols plug-in estimator as in Wu and Zhou (2018), lrvmethod = 0 uses the plug-in estimator in Zhou (2010), lrvmethod = 1 offers the debias difference-based estimator in Bai and Wu (2024), lrvmethod = 2 provides the plug-in estimator using the $\breve{\beta}$, the pilot estimator proposed in Bai and Wu (2024)
- gcv, 1 or 0, whether to use Generalized Cross Validation for the selection of b, the bandwidth parameter in the local linear regression, which will not be used when lrvmethod is -1, 1 or 2.
- neighbour, the number of neighbours in the extended minimum volatility, for example 1,2 or 3
- lb, the lower bound of the range of m in the extended minimum volatility Selection
- ub, the upper bound of the range of m in the extended minimum volatility Selection
- bw_set, the proposed grid of the range of bandwidth selection, which is only useful when lrvmethod = 1. if not presented, a rule of thumb method will be used for the data-driven range.
- tau_n, the value of τ when no data-driven selection is used. if tau is set to 0, the rule of thumb $n^{-1/5}$ will be used
- type, default 0, uses the residual-based statistic proposed in Wu and Zhou (2018). "type" can also be set to -1, using the coefficient-based statistic in Wu and Zhou (2018).
- ind, types of kernels
- 1 Triangular $1-|u|$, $u \leq 1$
- 2 Epanechnikov kernel $3/4(1-u^2)$, $u \leq 1$
- 3 Quartic $15/16(1-u^2)^2$, $u \leq 1$
- 4 Triweight $35/32(1-u^2)^3$, $u \leq 1$
- 5 Tricube $70/81(1-|u|^3)^3$, $u \leq 1$

Value

p-value of the structural stability test

References

Bai, L., & Wu, W. (2024). Difference-based covariance matrix estimation in time series nonparametric regression with application to specification tests. Biometrika, asae013.

Wu, W., and Zhou, Z. (2018). Gradient-based structural change detection for nonstationary time series M-estimation. The Annals of Statistics, 46(3), 1197-1224.

Politis, D. N., Romano, J. P., and Wolf, M. (1999). Subsampling. Springer Science & Business Media.

Examples

```
# choose a small B for tests
param = list(B = 50, bw_set = c(0.15, 0.25), gcv =1, neighbour = 1, lb = 10, ub = 20, type = 0)
n = 300
data = bregress2(n, 2, 1) # time series regression model with 2 changes points
param$lrvmethod = 0 # plug-in
heter_gradient(data, param, 4, 1)
param$lrvmethod = 1 # difference based
heter_gradient(data, param, 4, 1)
```

Heter_LRV *Long-run covariance matrix estimators*

Description

The function provides a wide range of estimators for the long-run covariance matrix estimation in non-stationary time series with covariates.

Usage

```
Heter_LRV(
  e,
  X,
  m,
  tau_n = 0,
  lrv_method = 1L,
  ind = 2L,
  print_deg = 0L,
  rescale = 0L,
  ncp = 0L
)
```

Arguments

e	vector, if the plug-in estimator is used, e should be the vector of residuals, OLS or nonparametric ones. If the difference-based debiased method is adopted, e should be the response time series, i.e., y. Specially, e should also be the response time series, i.e., y, if the plug-in estimator using the $\breve{\beta}$, the pilot estimator proposed in Bai and Wu (2024).
X	a matrix $n \times p$
m	integer, the window size.
tau_n	double, the smoothing parameter in the estimator. If tau_n is 0, a rule-of-thumb value will be automatically used.
lrv_method	the method of long-run variance estimation, lrvmethod = 0 uses the plug-in estimator in Zhou (2010), lrvmethod = 1 offers the debias difference-based estimator in Bai and Wu (2024), lrvmethod = 2 provides the plug-in estimator using the $\breve{\beta}$, the pilot estimator proposed in Bai and Wu (2024)

`ind`	types of kernels
`print_deg`	bool, whether to print information of non-positiveness, default $0n \times p$
`rescale`	bool, whether to use rescaling to correct the negative eigenvalues, default 0
`ncp`	1 no change points, 0 possible change points

- 1 Triangular $1 - |u|, u \leq 1$
- 2 Epanechnikov kernel $3/4(1 - u^2), u \leq 1$
- 3 Quartic $15/16(1 - u^2)^2, u \leq 1$
- 4 Triweight $35/32(1 - u^2)^3, u \leq 1$
- 5 Tricube $70/81(1 - |u|^3)^3, u \leq 1$

Value

a cube. The time-varying long-run covariance matrix $p \times p \times n$, where p is the dimension of the time series vector, and n is the sample size.

References

Bai, L., & Wu, W. (2024). Difference-based covariance matrix estimation in time series nonparametric regression with application to specification tests. Biometrika, asae013.

Zhou, Z. and Wu, W. B. (2010). Simultaneous inference of linear models with time varying coefficients.J. R. Stat. Soc. Ser. B. Stat. Methodol., 72(4):513–531.

Examples

```
param = list(d = -0.2, heter = 2, tvd = 0,
tw = 0.8, rate = 0.1, cur = 1, center = 0.3,
ma_rate =  0, cov_tw =  0.2, cov_rate = 0.1,
cov_center = 0.1, all_tw  = 1, cov_trend = 0.7)
data = Qct_reg(1000, param)
sigma = Heter_LRV(data$y, data$x, 3, 0.3, lrv_method = 1)
```

`hk_data` *This is data to be included in my package*

Description

This is data to be included in my package

Author(s)

T. S. Lau

References

Fan, J., and Zhang, W. (1999). Statistical estimation in varying coefficient models. The annals of Statistics, 27(5), 1491-1518.

`LocLinear` *Local linear Regression*

Description

Local linear estimates for time varying coefficients

Usage

```
LocLinear(bw, t, y, X, db_kernel = 0L, deriv2 = 0L, scb = 0L)
```

Arguments

`bw`	double, bandwidth
`t`	vector, time, 1:n/n
`y`	vector, response series to be tested for long memory in the next step
`X`	matrix, covariates matrix
`db_kernel`	bool, whether to use jackknife kernel, default 0
`deriv2`	bool,whether to return second-order derivative, default 0
`scb`	bool, whether to use the result for further calculation of simultaneous confidence bands.

Details

The time varying coefficients are estimated by

$$(\hat{\boldsymbol{\beta}}_{b_n}(t), \hat{\boldsymbol{\beta}}'_{b_n}(t)) = \mathbf{argmin}_{\eta_0,\eta_1}[\sum_{i=1}^{n} y_i - \mathbf{x}_i^{\mathrm{T}}\eta_0 - \mathbf{x}_i^{\mathrm{T}}\eta_1(t_i - t)^2 \boldsymbol{K}_{b_n}(t_i - t)]$$

where beta0 is $\hat{\boldsymbol{\beta}}_{b_n}(t)$, mu is $X^T\hat{\boldsymbol{\beta}}_{b_n}(t)$

Value

a list of results

- mu: the estimated trend
- beta0: time varying coefficient
- X_reg: a matrix whose j'th row is $x_j^T\hat{M}(t_j)$
- t: 1:n/n
- bw: bandwidth used
- X: covariates matrix
- y: response
- n: sample size
- p: dimension of covariates including the intercept
- invM: inversion of M matrix, when scb = 1

References

Zhou, Z., & Wu, W. B. (2010). Simultaneous inference of linear models with time varying coefficients. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 72(4), 513-531.

Examples

```
param = list(d = -0.2, heter = 2, tvd = 0,
 tw = 0.8, rate = 0.1, cur = 1, center = 0.3,
  ma_rate =  0, cov_tw =  0.2, cov_rate = 0.1,
   cov_center = 0.1, all_tw  = 1, cov_trend = 0.7)
n = 500
t = (1:n)/n
data = Qct_reg(n, param)
result = LocLinear(0.2, t, data$y, data$x)
```

`loc_constant` *Nonparametric smoothing*

Description

Nonparametric smoothing

Usage

```
loc_constant(bw, x, y, db_kernel = 0L)
```

Arguments

bw	double, bandwidth, between 0 and 1.
x	vector, covariates
y	matrix, response variables
db_kernel	bool, whether to use jackknife kernel, default 0

Value

a matrix of smoothed values

Examples

```
n <- 800
p <- 3
t <- (1:n)/n
V <-  matrix(rnorm(n * p), nrow = p)
V3 <- loc_constant(0.2, t, V,1)
```

`lrv_measure` *Comparing bias or mse of lrv estimators based on numerical methods*

Description

Comparing bias or mse of lrv estimators based on numerical methods

Usage

```
lrv_measure(
  data,
  param,
  lrvmethod,
  mvselect = -1,
  tau = 0,
  verbose_dist = FALSE,
  mode = "mse"
)
```

Arguments

data	a list of data
param	a list of parameters
lrvmethod	int, method of long-run variance estimation
mvselect	int, method of MV selection
tau	double, value of tau. If tau is 0, a rule-of-thunk value will be applied
verbose_dist	bool, whether to output distributional information
mode	default "mse", It can be set as "bias".

Value

empirical MSE of the estimator.

Examples

```
n = 300
param = list(gcv = 1, neighbour = 1,lb = 6, ub = 13, ind = 2)   # covariates heterskadecity
data = bregress2(n, 2, 1) # with 2 change pointa
lrv_measure(data, param, lrvmethod = -1, mvselect = -2) #ols plug-in
#debiased difference-based
lrv_measure(data, param, lrvmethod = 1, mvselect = -2)
```

MV_critical *Statistics-adapted values for extended minimum volatility selection.*

Description

Calculation of the variance of the bootstrap statistics for the extended minimum volatility selection.

Usage

```
MV_critical(
  y,
  data,
  R,
  gridm,
  gridtau,
  type = 1L,
  cvalue = 0.1,
  B = 100L,
  lrvmethod = 1L,
  ind = 2L,
  rescale = 0L
)
```

Arguments

y	vector, as used in the Heter_LRV
data	list, a list of data
R	a cube of standard.normal random variables.
gridm	vector, a grid of candidate m's.
gridtau	vector, a grid of candidate tau's.
type	integer, 1 KPSS 2 RS 3 VS 4 KS
cvalue	double, 1-quantile for the calculation of bootstrap variance, default 0.1.
B	integer, number of iterations for the calculation of bootstrap variance
lrvmethod	integer, see also Heter_LRV
ind	integer, the type of kernel, see also Heter_LRV
rescale	bool, whether to rescale when positiveness of the matrix is not obtained. default 0

Value

a matrix of critical values

References

Bai, L., & Wu, W. (2024). Difference-based covariance matrix estimation in time series nonparametric regression with application to specification tests. Biometrika, asae013.

See Also

Heter_LRV

Examples

```
###with Long memory parameter 0.2
param = list(d = -0.2, heter = 2,
 tvd = 0, tw = 0.8, rate = 0.1, cur = 1,
  center = 0.3, ma_rate =  0, cov_tw =  0.2,
  cov_rate = 0.1, cov_center = 0.1,
  all_tw  = 1, cov_trend = 0.7)
n = 1000
data = Qct_reg(n, param)
p = ncol(data$x)
t = (1:n)/n
B_c = 100 ##small value for testing
Rc = array(rnorm(n*p*B_c),dim = c(p,B_c,n))
result1 = LocLinear(0.2, t, data$y, data$x)
critical <- MV_critical(data$y, result1, Rc, c(3,4,5), c(0.2, 0.25, 0.3))
```

`MV_critical_cp` *Statistics-adapted values for extended minimum volatility selection.*

Description

Smoothing parameter selection for bootstrap tests for change point tests

Usage

```
MV_critical_cp(
  y,
  X,
  t,
  gridm,
  gridtau,
  cvalue = 0.1,
  B = 100L,
  lrvmethod = 1L,
  ind = 2L,
  rescale = 0L
)
```

Arguments

`y`	vector, as used in the Heter_LRV
`X`	matrix, covariates
`t`	vector, time points.
`gridm`	vector, a grid of candidate m's.
`gridtau`	vector, a grid of candidate tau's.
`cvalue`	double, 1-quantile for the calculation of bootstrap variance, default 0.1.
`B`	integer, number of iterations for the calculation of bootstrap variance
`lrvmethod`	integer, see also Heter_LRV
`ind`	integer, the type of kernel, see also Heter_LRV
`rescale`	bool, whether to rescale when positiveness of the matrix is not obtained. default 0

Value

a matrix of critical values

References

Bai, L., & Wu, W. (2024). Difference-based covariance matrix estimation in time series nonparametric regression with application to specification tests. Biometrika, asae013.

Examples

```
n = 300
t = (1:n)/n
data = bregress2(n, 2, 1) # time series regression model with 2 changes points
critical = MV_critical_cp(data$y, data$x,t,  c(3,4,5), c(0.2,0.25, 0.3))
```

`MV_ise_heter_critical` *MV method*

Description

Selection of smoothing parameters for bootstrap tests by choosing the index minimizing the volatility of bootstrap statistics or long-run variance estimators in the neighborhood computed before.

Usage

```
MV_ise_heter_critical(critical, neighbour)
```

Arguments

`critical`	a matrix of critical values
`neighbour`	integer, number of neighbours

Value

a list of results,

- minp: optimal row number
- minq: optimal column number
- min_ise: optimal value

References

Bai, L., & Wu, W. (2024). Difference-based covariance matrix estimation in time series nonparametric regression with application to specification tests. Biometrika, asae013.

Examples

```
param = list(d = -0.2, heter = 2,
 tvd = 0, tw = 0.8, rate = 0.1,
 cur = 1, center = 0.3, ma_rate =  0,
 cov_tw =  0.2, cov_rate = 0.1,
 cov_center = 0.1, all_tw  = 1, cov_trend = 0.7)
n = 1000
data = Qct_reg(n, param)
p = ncol(data$x)
t = (1:n)/n
B_c = 100 ##small value for testing
Rc = array(rnorm(n*p*B_c),dim = c(p,B_c,n))
result1 = LocLinear(0.2, t, data$y, data$x)
gridm = c(3,4,5)
gridtau = c(0.2, 0.25, 0.3)
critical <- MV_critical(data$y, result1, Rc, gridm, gridtau)
mv_result = MV_ise_heter_critical(critical,  1)
m = gridm[mv_result$minp + 1]
tau_n = gridtau[mv_result$minq + 1]
```

Qct_reg *Simulate data from time-varying time series regression model*

Description

Simulate data from time-varying time series regression model

Usage

```
Qct_reg(T_n, param, type = 1)
```

Arguments

T_n	int, sample size
param	list, a list of parameters
type	type = 1 means the long memory expansion begins from its infinite past, type = 2 means the long memory expansion begins from t = 0

Value

list, a list of data, covariates, response and errors.(before and after fractional difference)

Examples

```
param = list(d = -0.2, heter = 2, tvd = 0,
tw = 0.8, rate = 0.1, cur = 1, center = 0.3,
ma_rate =  0, cov_tw =  0.2, cov_rate = 0.1,
cov_center = 0.1, all_tw  = 1, cov_trend = 0.7)
n = 500
data = Qct_reg(n, param)
```

Qt_data *Simulate data from time-varying trend model*

Description

Simulate data from time-varying trend model

Usage

```
Qt_data(T_n, param)
```

Arguments

T_n integer, sample size

param a list of parameters

- tw double, squared root of variance of the innovations
- rate double, magnitude of non-stationarity
- center double, the center of the ar coefficient
- ma_rate double, ma coefficient

Value

a vector of non-stationary time series

Examples

```
param = list(d = -0.2,  tvd = 0, tw = 0.8, rate = 0.1, center = 0.3, ma_rate =  0, cur = 1)
data = Qt_data(300, param)
```

`rule_of_thumb` *rule of thumb interval for the selection of smoothing parameter b*

Description

The function will compute a data-driven interval for the Generalized Cross Validation performed later, see also Bai and Wu (2024) .

Usage

```
rule_of_thumb(y, x)
```

Arguments

`y`	a vector, the response variable.
`x`	a matrix of covariates. If the intercept should be includes, the elements of the first column should be 1.

Value

c(left, right), the vector with the left and right points of the interval

References

Bai, L., & Wu, W. (2024). Detecting long-range dependence for time-varying linear models. Bernoulli, 30(3), 2450-2474.

Examples

```
param = list(d = -0.2, heter = 2, tvd = 0,
tw = 0.8, rate = 0.1, cur = 1, center = 0.3,
 ma_rate =  0, cov_tw =  0.2, cov_rate = 0.1,
 cov_center = 0.1, all_tw  = 1, cov_trend = 0.7)
data = Qct_reg(1000, param)
rule_of_thumb(data$y, data$x)
```

`sim_T` *bootstrap distribution*

Description

bootstrap distribution of the gradient based structural stability test

Usage

```
sim_T(X, t, sigma, m, B, type = 0L)
```

Arguments

X	matrix of covariates
t	vector of time points
sigma	a cube of long-run covariance function.
m	int value of window size
B	int, number of iteration
type	type of tests, residual-based or coefficient-based

Value

a vector of bootstrap statistics

Examples

```
param = list(B = 50, bw_set = c(0.15, 0.25), gcv =1, neighbour = 1, lb = 10, ub = 20, type = 0)
n = 300
data = bregress2(n, 2, 1) # time series regression model with 2 changes points
sigma = Heter_LRV(data$y, data$x, 3, 0.3, lrv_method = 1)
bootstrap = sim_T(data$x, (1:n)/n, sigma, 3, 20) ### 20 iterations
```

Index