Package ‘QR.break’

April 23, 2025

Title Structural Breaks in Quantile Regression

Version 1.0.2

Maintainer Zhongjun Qu <qu@bu.edu>

Description Methods for detecting structural breaks, determining the number of breaks, and estimating break locations in linear quantile regression, using one or multiple quantiles, based on Qu (2008) and Oka and Qu (2011). Applicable to both time series and repeated cross-sectional data. The main function is rq.break().
References for detailed theoretical and empirical explanations:
(1) Qu, Z. (2008). ``Testing for Structural Change in Regression Quantiles.'' Journal of Econometrics, 146(1), 170-184 <doi:10.1016/j.jeconom.2008.08.006>
(2) Oka, T., and Qu, Z. (2011). ``Estimating Structural Changes in Regression Quantiles.'' Journal of Econometrics, 162(2), 248-267 <doi:10.1016/j.jeconom.2011.01.005>.

License GPL (>= 3)

Depends R (>= 4.3.0)

Encoding UTF-8

NeedsCompilation no

RoxygenNote 7.3.2

Imports quantreg

Suggests SparseM

LazyData true

Author Zhongjun Qu [aut, cre],
Tatsushi Oka [aut],
Samuel Messer [ctb]

Repository CRAN

Date/Publication 2025-04-23 17:50:02 UTC

Contents

brdate . . . 2
ci.date.m . . . 3
dq . . . 5
dq.test.0vs1 . . . 7
dq.test.lvsl_1 . . . 8
driver . . . 9
gdp . . . 10
gen.long . . . 11
res.surface . . . 12
rq.break . . . 13
rq.est.full . . . 15
rq.est.regime . . . 16
sq . . . 17
sq.test.0vs1 . . . 18
sq.test.lvsl_1 . . . 19

Index **21**

`brdate` *Estimating Break Dates in a Quantile Regression*

Description

This function estimates break dates in a quantile regression model, allowing for up to `m` breaks. When `m = 1`, this function finds the optimal single-break partition within a segment by evaluating the objective function at each possible break point to determine the break date. When `m > 1`, a dynamic programming algorithm is used to efficiently determine the break dates.

Usage

```
brdate(y, x, n.size = 1, m, trim.size, vec.long)
```

Arguments

`y`	A numeric vector of dependent variables $(NT \times 1)$.
`x`	A numeric matrix of regressors $(NT \times p)$; a column of ones will be automatically added to x.
`n.size`	An integer specifying the number of cross-sectional units (N), equal to 1 for time series data.
`m`	An integer specifying the maximum number of breaks allowed.
`trim.size`	An integer representing the minimum length of a regime, which ensures break dates are not too short or too close to the sample's boundaries.
`vec.long`	A numeric vector/matrix used in dynamic programming, storing pre-computed objective function values for all possible segments of the sample for optimization. Produced by the function `gen.long()`.

Details

The function first determines the optimal one-break partition. If multiple breaks are allowed (`m > 1`), it applies a dynamic programming algorithm as in Bai and Perron (2003) to minimize the objective function. The algorithm finds break dates by iterating over all possible partitions, returning optimal break locations and associated objective function values.

Value

An upper triangular matrix (`m × m`) containing estimated break dates. The row index represents break dates, and the column index corresponds to the permitted number of breaks before the ending date.

References

Bai, J. and P. Perron (2003). Computation and analysis of multiple structural change models. *Journal of Applied Econometrics*, 18(1), 1-22.

Oka, T. and Z. Qu (2011). Estimating structural changes in regression quantiles, *Journal of Econometrics*, 162(2), 248-267.

Examples

```
# data
data(gdp)
y = gdp$gdp
x = gdp[,c("lag1", "lag2")]
n.size = 1
T.size = length(y) # number of time periods

# setting
vec.tau   = seq(0.20, 0.80, by = 0.150)
trim.e    = 0.2
trim.size = round(T.size * trim.e)  #minimum length of a regime
m.max     = 3

# compute the objective function under all possible partitions
out.long   = gen.long(y, x, vec.tau, n.size, trim.size)
vec.long.m = out.long$vec.long  ## for break estimation using multiple quantiles

# break date
mat.date = brdate(y, x, n.size, m.max, trim.size, vec.long.m)
print(mat.date)
```

`ci.date.m` *Confidence Intervals for Break Dates*

Description

This function constructs confidence intervals for break dates based on a single quantile or multiple quantiles (specified by the user).

Usage

```
ci.date.m(y, x, vec.tau, vec.date, n.size = 1, v.b = 2)
```

Arguments

`y`	A numeric vector of dependent variables ($NT \times 1$).
`x`	A numeric matrix of regressors ($NT \times p$).
`vec.tau`	A numeric vector of quantiles of interest.
`vec.date`	A numeric vector of estimated break dates.
`n.size`	An integer specifying the size of the cross section (N).
`v.b`	A numeric value specifying the confidence level: • `v.b = 1` for the 90% confidence interval. • `v.b = 2` for the 95% confidence interval.

Value

A numeric matrix where:

- The 1st column contains the break dates.
- The 2nd and 3rd columns contain the lower and upper bounds of the confidence intervals, respectively.

References

Oka, T. and Z. Qu (2011). Estimating Structural Changes in Regression Quantiles. *Journal of Econometrics*, 162(2), 248–267.

Examples

```
# data
data(gdp)
y = gdp$gdp
x = gdp[,c("lag1", "lag2")]

# quantiles
vec.tau  = 0.8

# break dates (point estimates)
vec.date = c(146, 200)

# Calculate confidence intervals for break dates
res = ci.date.m(y, x, vec.tau, vec.date, n.size = 1, v.b = 2)
print(res)
```

dq — *Sequential Determination of the Number of Breaks Using the DQ Test*

Description

This function determines the number of breaks in a model by sequentially applying the DQ($l|l+1$) test. It tests for additional breaks by comparing the test statistic to critical values at various significance levels.

Usage

```
dq(
  y,
  x,
  vec.tau,
  q.L,
  q.R,
  n.size = 1,
  m.max,
  trim.size,
  mat.date,
  d.Sym,
  table.cv
)
```

Arguments

`y`	A numeric vector of dependent variables ($NT \times 1$).
`x`	A numeric matrix of regressors ($NT \times p$).
`vec.tau`	A numeric vector of quantiles of interest.
`q.L`	A numeric value specifying the left-end quantile range for the DQ test.
`q.R`	A numeric value specifying the right-end quantile range for the DQ test.
`n.size`	An integer specifying the size of the cross-section (N).
`m.max`	An integer indicating the maximum number of breaks allowed.
`trim.size`	A numeric trimming value (the minimum length of a regime).
`mat.date`	A numeric matrix of break dates.
`d.Sym`	A logical value indicating whether the quantile range is symmetric satisfying $q.R = 1 - q.L$.
`table.cv`	A matrix of simulated critical values for cases not covered by the response surface.

Value

A list containing:

`test` A numeric vector of DQ test statistics.

`cv` A numeric matrix of critical values for the DQ test at 10, 5, and 1 percent significance levels.

`date` A numeric matrix of estimated break dates.

`nbreak` A numeric vector indicating the number of detected breaks at different significance levels.

Examples

```
# This example may take substantial time for automated package
# checks since it involves dynamic programming
data(gdp)
y = gdp$gdp
x = gdp[,c("lag1", "lag2")]
n.size = 1
T.size = length(y) # number of time periods

# setting
vec.tau   = seq(0.20, 0.80, by = 0.150)
trim.e    = 0.2
trim.size = round(T.size * trim.e)  #minimum length of a regime
m.max     = 3

# dynamic program algorithm to compute the objective function
out.long   = gen.long(y, x, vec.tau, n.size, trim.size)
mat.long.s = out.long$mat.long  ## for break estimation using individual quantile
vec.long.m = out.long$vec.long  ## for break estimation using multiple quantiles jointly

# break date
mat.date = brdate(y, x, n.size, m.max, trim.size, vec.long.m)

## qunatile ranges (left and right)
q.L   = 0.2
q.R   = 0.8
d.Sym = TRUE ## symmetric trimming of quantiles
table.cv = NULL ##covered by the response surface because d.Sym = TRUE

# determine the number of breaks
out.m = dq(y, x, vec.tau, q.L, q.R, n.size, m.max, trim.size, mat.date, d.Sym, table.cv)

# result
print(out.m)
```

`dq.test.0vs1` *Test for the Presence of a Break within a Range of Quantiles*

Description

This procedure computes test statistics for detecting a single structural break within a range of quantiles. The null hypothesis is that there is no break in any quantile in the specified range; the alternative is that at least one quantile in the range is affected by a break.

Usage

```
dq.test.0vs1(y, x, q.L, q.R, n.size = 1)
```

Arguments

`y`	A numeric vector of dependent variables ($NT \times 1$).
`x`	A numeric matrix of regressors ($NT \times p$), excluding the constant term.
`q.L`	A numeric value specifying the lower bound of the quantile range.
`q.R`	A numeric value specifying the upper bound of the quantile range.
`n.size`	An integer specifying the size of the cross-section (N).

Value

A numeric scalar representing the DQ test statistic.

References

Koenker, R. and Bassett Jr, G. (1978). Regression quantiles. *Econometrica*, *46*(1), 33–50.

Qu, Z. (2008). Testing for structural change in regression quantiles. *Journal of Econometrics*, *146*(1), 170–184.

Examples

```
## data
data(gdp)
y = gdp$gdp
x = gdp[,c("lag1", "lag2")]

## qunatile range (left and right limits)
q.L = 0.2
q.R = 0.8

## N
n.size = 1

# dq-test
result = dq.test.0vs1(y, x, q.L, q.R, n.size)
```

```
print(result)
```

dq.test.lvsl_1 *Sequential Test for Additional Breaks within a Range of Quantiles*

Description

This function performs a sequential test to determine whether the number of breaks in a quantile regression model should be increased from L to $L + 1$ using multiple quantiles.

Usage

```
dq.test.lvsl_1(y, x, q.L, q.R, n.size = 1, vec.date)
```

Arguments

y	A numeric vector of dependent variables ($NT \times 1$).
x	A numeric matrix of regressors ($NT \times p$).
q.L	A numeric value specifying the lower bound of the quantile range.
q.R	A numeric value specifying the upper bound of the quantile range.
n.size	An integer specifying the size of cross-sections (N).
vec.date	A numeric vector ($L \times 1$) of estimated break dates under the null hypothesis.

Details

This procedure tests for the existence of L breaks against $L + 1$ breaks based on multiple quantiles: $H_0 : L$ breaks vs. $H_1 : L + 1$ breaks.

Value

A numeric value representing the DQ test statistic.

References

Qu, Z. (2008). Testing for Structural Breaks in Regression Quantiles. *Journal of Econometrics*, 146(1), 170-184.

Examples

```
# Load data
data(gdp)
y = gdp$gdp
x = gdp[,c("lag1", "lag2")]

# Set quantile range (left and right limits)
q.L = 0.2
```

```
q.R = 0.8

# Set N parameter
n.size = 1

# Specify break date under H_0
vec.date = 146

# Run the test
result = dq.test.lvsl_1(y, x, q.L, q.R, n.size, vec.date)
print(result)
```

`driver` *The Dataset for Young Drivers*

Description

This is a repeated cross-sectional data set on young drivers (under 21 years old) involved in motor vehicle accidents in the state of California from 1983 to 2007 (quarterly data). The data are obtained from the National Highway Traffic Safety Administration (NHTSA), which include the blood alcohol concentration (BAC) of the driver, their age, gender, and whether the crash was fatal.

Usage

```
data(driver)
```

Format

A data frame with five variables:

- `yq`: Year and quarter ("Year Quarter" format, e.g., "1983 Q2").
- `bac`: The blood alcohol concentration (BAC) level of the driver.
- `age`: The driver's age.
- `gender`: A gender dummy, with 1 for male and 0 for female.
- `winter`: A dummy variable for the fourth quarter, with 1 for Q4 and 0 otherwise.

Details

Motor vehicle crashes are the leading cause of death among youth aged 15–20, with a high proportion involving drunk driving. The BAC level is an important measure of alcohol impairment. Oka and Qu (2011) used this data to examine whether and how young drivers' drinking behaviors have changed over time.

References

Oka, T. and Z. Qu (2011). Estimating Structural Changes in Regression Quantiles. *Journal of Econometrics*, 162(2), 248–267.

Examples

```
data(driver)
names(driver)
summary(driver)
```

---

gdp *US Real GDP Growth Data*

Description

This dataset contains quarterly real GDP growth rates in the US from 1947 Q4 to 2009 Q2. It is used in Oka and Qu (2011) to examine whether and where distributional breaks occur in US GDP.

Usage

```
data(gdp)
```

Format

A data frame with four variables:

- `yq` A character vector representing the year and quarter (e.g., "1947 Q2").
- `gdp` A numeric vector of real GDP growth rates (annualized by multiplying by 4).
- `lag1` A numeric vector representing the first-order lagged value of `gdp`.
- `lag2` A numeric vector representing the second-order lagged value of `gdp`.

References

Oka, T. and Z. Qu (2011). Estimating Structural Changes in Regression Quantiles. *Journal of Econometrics*, *162*(2), 248–267.

Examples

```
data(gdp)
names(gdp)
summary(gdp)
```

`gen.long` *Dynamic Programming Algorithm*

Description

This function computes the objective function values for all possible segments of the sample.

Usage

```
gen.long(y, x, vec.tau, n.size = 1, trim.size)
```

Arguments

`y`	A numeric vector of dependent variables (NT × 1).
`x`	A numeric matrix of regressors (NT × p).
`vec.tau`	A vector of quantiles of interest.
`n.size`	The size of the cross-section; default is set to 1.
`trim.size`	The minimum length of a regime (integer).

Value

A list containing:

mat.long A matrix of objective function values for separate quantiles.

vec.long A matrix of objective function values for combined quantiles.

References

Bai, J and P. Perron (2003). Computation and Analysis of Multiple Structural Change Models. *Journal of Applied Econometrics*, 18(1), 1-22.

Examples

```
# This example may take substantial time for automated package checks
data(gdp)
y = gdp$gdp
x = gdp[,c("lag1", "lag2")]
n.size = 1
T.size = length(y) # number of time periods

# setting
vec.tau   = seq(0.20, 0.80, by = 0.150)
trim.e    = 0.2
trim.size = round(T.size * trim.e)  #minimum length of a regime

out.long   = gen.long(y, x, vec.tau, n.size, trim.size)
```

`res.surface` *Compute Critical Values for the DQ test using a Response Surface*

Description

This function returns critical values obtained from a response surface analysis. Note that this procedure only applies when the trimming is symmetric, i.e., $q.R = 1 - q.L$.

Usage

```
res.surface(p, l, q.L, q.R, d.Sym)
```

Arguments

`p`	The number of parameters in the model.
`l`	The number of breaks under the null H_0 (i.e., $l + 1$ under H_1).
`q.L`	The lower bound of the quantile range.
`q.R`	The upper bound of the quantile range (not used in the function because $q.R = 1 - q.L$).
`d.Sym`	A logical value. Must be TRUE, as this method applies only to symmetric trimming ($q.R = 1 - q.L$).

Value

A numeric vector of length 3 containing critical values at the 10%, 5%, and 1% significance levels.

Examples

```
# The number of regerssors
p = 5
## The number of breaks under the null
l = 2

# qunatile range (left and right limits)
q.L = 0.2
q.R = 0.8

# symmetric quantile trimming is true
d.Sym = TRUE

## critical values from response surface
cvs = res.surface(p, l, q.L, q.R, d.Sym)

print(cvs)
```

rq.break — *Testing for Breaks and Estimating Break Dates and Sizes with Confidence Intervals*

Description

This is the main function of this package for testing breaks in quantile regression models and estimating break dates and break sizes with corresponding confidence intervals.

Usage

```
rq.break(y, x, vec.tau, N, trim.e, vec.time, m.max, v.a, v.b, verbose)
```

Arguments

`y`	a numeric vector, the outcome variable (NT x 1), the first N units are from the first period, the next N from the second period, and so forth.
`x`	A matrix of regressors (NT x p), structured in the same way as y, a column of ones will be automatically added to x.
`vec.tau`	a numeric vector, quantiles used for break estimation, for example `vec.tau = seq(0.20, 0.80, by = 0.10)`
`N`	a numeric vector, the number of cross-sectional units. Set to 1 for a time series quantile regression.
`trim.e`	a scalar between 0 and 1, the trimming proportion. For example, if `trim.e=0.1`, the minimum regime length is 0.1 times the data span.
`vec.time`	a vector of dates, needed for reporting the estimated break dates, in the format of (starting date...ending date); If set to NULL, the break dates will be reported as indices (e.g., 55 for the 55th observation in the sample).
`m.max`	the maximum number of breaks allowed.
`v.a`	the significance level used for determining the number of breaks; 1, 2 or 3 for 10%, 5% or 1%, respectively
`v.b`	the coverage level for constructing the confidence intervals of break dates; 1 or 2 for 90% and 95%, respectively.
`verbose`	Logical; set to TRUE to print estimates to the console. Default is FALSE.

Value

A list containing:

- `$s.out`: A list with break testing results, estimated break dates, confidence intervals, and coefficient estimates based on individual quantiles.
- `$m.out`: A list with break testing results, estimated break dates, confidence intervals, and coefficient estimates obtained by testing and estimating breaks using multiple quantiles simultaneously.

Each list (s.out or m.out) contains:

- test_tau: A matrix of test statistics and critical values for break detection at quantile tau.
- nbreak_tau: The number of detected breaks at quantile tau.
- br_est_tau: A matrix of estimated break dates and their confidence intervals at quantile tau.
- br_est_time_tau: The same as br_est_tau, but with break dates reported in calendar format (if vec.time is provided and is not NULL).
- coef_tau: Estimated regression coefficients for each regime at quantile tau.
- bsize_tau: Break size estimates for each transition between regimes at quantile tau.

References

Koenker, R. and G. Bassett Jr. (1978). Regression quantiles. *Econometrica*, 46(1), 33-50.

Oka, T. and Z. Qu (2011). Estimating Structural Changes in Regression Quantiles. *Journal of Econometrics*, 162(2), 248-267.

Qu, Z. (2008). Testing for Structural Change in Regression Quantiles. *Journal of Econometrics*, 146(1), 170-184.

Examples

```
## Example 1
## Time series example, using GDP data
## data
data(gdp)
y        = gdp$gdp
x        = gdp[,c("lag1", "lag2")]
vec.time = gdp$yq

## the maximum number of breaks allowed
m.max = 3

## the signifance level for sequenatial testing
## 1, 2 or 3 for 10%, 5% or 1%, respectively
v.a = 2

## the significance level for the confidence intervals of estimated break dates.
## 1 or 2 for 90% and 95%, respectively.
v.b = 2

## the size of the cross-section
N = 1

## the trimming proportion for estimating the break dates
## (represents the minimum length of a regime; used to exclude
## the boundaries of the sample)
trim.e = 0.15

## quantiles
vec.tau = seq(0.20, 0.80, by = 0.150)
```

```
verbose = FALSE #do not print

## main estimation
res = rq.break(y, x, vec.tau, N, trim.e, vec.time, m.max, v.a, v.b, verbose)
```

`rq.est.full` *Estimating Break Sizes and Confidence Intervals Given Break Dates*

Description

This procedure estimates a linear quantile regression given a set of break dates. It is structured to compute break sizes between adjacent regimes and their confidence intervals.

Usage

```
rq.est.full(y, x, v.tau, vec.date, n.size = 1)
```

Arguments

`y`	A numeric vector of dependent variables ($NT \times 1$).
`x`	A numeric matrix of regressors ($NT \times p$).
`v.tau`	A numeric value representing the quantile of interest.
`vec.date`	A numeric vector of break dates, specified by the user.
`n.size`	An integer specifying the size of the cross-section (N).

Value

An object from the quantile regression estimates, `rq()`, with structural breaks.

References

Koenker, R. and G. Bassett Jr. (1978). Regression Quantiles. *Econometrica*, 46(1), 33–50.

Oka, T. and Z. Qu (2011). Estimating Structural Changes in Regression Quantiles. *Journal of Econometrics*, 162(2), 248–267.

Examples

```
## data
data(gdp)
y = gdp$gdp
x = gdp[,c("lag1", "lag2")]

## quantile
v.tau = 0.8
```

```
## break date
vec.date = 146

# cross-sectional size
n.size = 1

## estimation
rq.est.full(y, x, v.tau, vec.date, n.size)
```

`rq.est.regime` *Regime-Specific Coefficients and Confidence Intervals Given Break Dates*

Description

This function estimates the coefficients for each regime, given the break dates.

Usage

```
rq.est.regime(y, x, v.tau, vec.date, n.size = 1)
```

Arguments

`y`	A vector of dependent variables ($NT \times 1$).
`x`	A matrix of regressors ($NT \times p$).
`v.tau`	The quantile of interest.
`vec.date`	A vector of estimated break dates.
`n.size`	The cross-sectional sample size (N).

Value

A list containing the estimated coefficients for each regime.

Examples

```
## data
data(gdp)
y        = gdp$gdp
x        = gdp[,c("lag1", "lag2")]

## quantile
v.tau = 0.8

## break date
```

```
vec.date = 146

# cross-sectional size
n.size = 1

## estimation
result = rq.est.regime(y, x, v.tau, vec.date, n.size)
print(result)
```

---

sq *Determine the Number of Breaks Using the SQ(l|l+1) Test*

Description

This procedure sequentially applies the SQ test to determine the number of breaks, based on a single quantile.

Usage

```
sq(y, x, v.tau, n.size = 1, m.max, trim.size, mat.date)
```

Arguments

y	A numeric vector of dependent variables ($NT \times 1$).
x	A numeric matrix of regressors ($NT \times p$).
v.tau	A numeric value representing the quantile of interest.
n.size	An integer specifying the size of the cross-section (N).
m.max	An integer specifying the maximum number of breaks allowed.
trim.size	A numeric value specifying the trimming size (the minimum length of a segment).
mat.date	A numeric matrix of break dates.

Value

A list with the following components:

`test` A numeric vector of SQ test statistics.

`cv` A numeric matrix of critical values for the SQ test, with the 1st, 2nd, and 3rd rows corresponding to the 10%, 5%, and 1% significance levels.

`date` A numeric matrix of break dates, with the 1st, 2nd, and 3rd rows corresponding to the 10%, 5%, and 1% significance levels.

`nbreak` A numeric vector indicating the number of breaks at the 10%, 5%, and 1% significance levels.

Examples

```
## data
data(gdp)
y = gdp$gdp
x = gdp[,c("lag1", "lag2")]

## quantile
v.tau = 0.8

# cross-sectional size
n.size = 1

# the maximum number of breaks
m.max = 3

## trim
T.size    = length(y)
trim.e    = 0.2
trim.size = round(T.size * trim.e)  #minimum length of a regime

# get.long
out.long   = gen.long(y, x, v.tau, n.size, trim.size)
mat.long.s = out.long$mat.long  ## for individual quantile

# mat.date
mat.date = brdate(y, x, n.size, m.max, trim.size, mat.long.s)

# sq
result = sq(y, x, v.tau, n.size, m.max, trim.size, mat.date)
print(result)
```

---

sq.test.0vs1 *Test for a Structural Break in a Conditional Quantile*

Description

The function implements a break test to evaluate whether a single structural break exists at a given quantile.

Usage

```
sq.test.0vs1(y, x, v.tau, n.size = 1)
```

Arguments

y	A numeric vector of dependent variables ($NT \times 1$).
x	A numeric matrix of regressors ($NT \times p$).

`v.tau`	A numeric value representing the quantile level.
`n.size`	An integer specifying the size of the cross-section (N).

Value

A numeric value representing the test statistic for the presence of a structural break.

References

Koenker, R. and G. Bassett Jr, (1978). Regression Quantiles. *Econometrica*, *46*(1), 33–50.

Qu, Z. (2008). Testing for Structural Change in Regression Quantiles. *Journal of Econometrics*, *146*(1), 170–184.

Examples

```
## data
data(gdp)
y = gdp$gdp
x = gdp[,c("lag1", "lag2")]

## quantile
v.tau = 0.8

# cross-sectional size
n.size = 1

# sq test: 0 vs 1
result = sq.test.0vs1(y, x, v.tau, n.size)
print(result)
```

`sq.test.lvsl_1` *Sequential Test for an Additional Break in a Conditional Quantile*

Description

This function tests the null hypothesis of L breaks against the alternative hypothesis of $L+1$ breaks in a single conditional quantile.

Usage

```
sq.test.lvsl_1(y, x, v.tau, n.size = 1, vec.date)
```

Arguments

`y`	A numeric vector of dependent variables ($NT \times 1$).
`x`	A numeric matrix of regressors ($NT \times p$).
`v.tau`	A numeric value representing the quantile of interest.
`n.size`	An integer specifying the size of the cross-section (N).
`vec.date`	A numeric vector of break dates estimated under the null hypothesis.

Details

The function sequentially tests for breaks by splitting the sample conditional on the break dates under the null hypothesis. At each step, it applies `sq.test.0vs1()` to compare the hypothesis of no additional break against one more break.

Value

A numeric value representing the test statistic.

References

Qu, Z. (2008). Testing for Structural Change in Regression Quantiles. *Journal of Econometrics*, 146(1), 170-184.

Oka, T. and Z. Qu (2011). Estimating Structural Changes in Regression Quantiles. *Journal of Econometrics*, 162(2), 248-267.

Examples

```
## data
data(gdp)
y = gdp$gdp
x = gdp[,c("lag1", "lag2")]

## quantile
v.tau = 0.8

# cross-sectional size
n.size = 1

## break date
vec.date = 146

## sq-test: 1 vs 2
result = sq.test.1vs1_1(y, x, v.tau, n.size, vec.date)
print(result)
```

Index

* **datasets**
    driver, 9
    gdp, 10

brdate, 2

ci.date.m, 3

dq, 5
dq.test.0vs1, 7
dq.test.lvsl_1, 8
driver, 9

gdp, 10
gen.long, 11

res.surface, 12
rq.break, 13
rq.est.full, 15
rq.est.regime, 16

sq, 17
sq.test.0vs1, 18
sq.test.lvsl_1, 19