Package ‘QTE.RD’

March 18, 2024

Title Quantile Treatment Effects under the Regression Discontinuity Design

Version 1.0.0

Author Zhongjun Qu [aut, cph],
Jungmo Yoon [aut, cre, cph]

Maintainer Jungmo Yoon <jmyoon@hanyang.ac.kr>

Description Provides comprehensive methods for testing, estimating, and conducting uniform inference on quantile treatment effects (QTEs) in sharp regression discontinuity (RD) designs, incorporating covariates and implementing robust bias correction methods of Qu, Yoon, Perron (2024) <doi:10.1162/rest_a_01168>.

Encoding UTF-8

RoxygenNote 7.3.1

Imports quantreg, plotrix, stats

Suggests spelling

Language en-US

NeedsCompilation no

Repository CRAN

Date/Publication 2024-03-18 18:10:05 UTC

R topics documented:

`QTE.RD-package` *QTE.RD: Quantile Treatment Effects under the Regression Discontinuity Design*

Description

Provides comprehensive methods for testing, estimating, and conducting uniform inference on quantile treatment effects (QTEs) in sharp regression discontinuity (RD) designs, incorporating covariates and implementing robust bias correction methods of Qu, Yoon, Perron (2024) doi:10.1162/rest_a_01168.

Details

The package `QTE.RD` includes four main functions:

- `rd.qte` estimates QTEs and provides uniform confidence bands, with or without covariates, and with or without robust bias correction.
- `rdq.test` conducts tests for three hypotheses, related to the significance of treatment effects, homogeneous treatment effects, and uniformly positive or negative treatment effects.
- `rdq.bandwidth` implements two bandwidth selection rules: the cross-validation bandwidth and the MSE optimal bandwidth.
- `plot.qte` generates figures summarizing the treatment effects along with their confidence bands.

Author(s)

- Zhongjun Qu qu@bu.edu
- Jungmo Yoon jmyoon@hanyang.ac.kr

References

Zhongjun Qu, Jungmo Yoon, Pierre Perron (2024), "Inference on Conditional Quantile Processes in Partially Linear Models with Applications to the Impact of Unemployment Benefits," The Review of Economics and Statistics; https://doi.org/10.1162/rest_a_01168

Zhongjun Qu and Jungmo Yoon (2019), "Uniform Inference on Quantile Effects under Sharp Regression Discontinuity Designs," Journal of Business and Economic Statistics, 37(4), 625–647; https://doi.org/10.1080/07350015.2017.1407323

`cv.bandwidth` *Cross-validation bandwidth*

Description

`cv.bandwidth` implements the cross-validation bandwidth selection rule. The function `rdq.bandwidth` calls this function to obtain the CV bandwidth.

Usage

```
cv.bandwidth(y, x, z, dz, x0, val, xl, order, bdy)
```

Arguments

`y`	a numeric vector, the outcome variable.
`x`	the running variable.
`z`	additional covariates.
`dz`	the number of covariates z.
`x0`	the cutoff point.
`val`	a set of candidate values for the CV bandwidth.
`xl`	if *xl=0.5*, the CV bandwidth is computed using the 50% of observations closest to x_0.
`order`	either 1 or 2. When *p.order=1*, a local linear regression is used, and when *p.order=2*, a local quadratic regression is used.
`bdy`	either 0 or 1. When *bdy=1*, the CV bandwidth is computed by treating x as a boundary point. Otherwise, x is treated as an interior point.

Value

A list with elements:

h.cv the selected CV bandwidth values at the median.

cand the criterion function evaluated at each of candidate value.

References

Zhongjun Qu, Jungmo Yoon, Pierre Perron (2024), "Inference on Conditional Quantile Processes in Partially Linear Models with Applications to the Impact of Unemployment Benefits," The Review of Economics and Statistics; https://doi.org/10.1162/rest_a_01168

Zhongjun Qu and Jungmo Yoon (2019), "Uniform Inference on Quantile Effects under Sharp Regression Discontinuity Designs," Journal of Business and Economic Statistics, 37(4), 625–647; https://doi.org/10.1080/07350015.2017.1407323

See Also

`rdq.bandwidth()`

Examples

```
n = 500
x = runif(n,min=-4,max=4)
d = (x > 0)
y = x + 0.3*(x^2) - 0.1*(x^3) + 1.5*d + rnorm(n)
cv.bandwidth(y=y, x=x, z=NULL, dz=0, x0=0, val=c(1,2,3,4), xl=0.5, order=2, bdy=1)
cv.bandwidth(y=y, x=x, z=NULL, dz=0, x0=0, val=c(1,2,3,4), xl=0.5, order=1, bdy=1)
```

---

depa *Epanechnikov kernel*

Description

Epanechnikov kernel

Usage

```
depa(xx, loc = 0, scale = 1)
```

Arguments

`xx`	the evaluation points.
`loc`	the location parameter (normalized to be 0).
`scale`	the scale parameter (normalized to be 1).

Value

values.

Examples

```
depa(0)
depa(seq(-1,1,by=0.05))
```

make.band *Uniform confidencedevtools::submit_cran() bands*

Description

make.band constructs uniform confidence bands using the output of rdq.sim. The function rdq.band calls this function to generates uniform bands.

Usage

```
make.band(n.sam,Dc.p,Dc.m,Dr.p,Dr.m,dz,cov,taus,hh,Qy.p,Qy.m,
     bias.p,bias.m,alpha,n.sim)
```

Arguments

n.sam	the sample size.
Dc.p	simulated values from $D_{1,v}(x_0^+, z, \tau)$.
Dc.m	simulated values from $D_{1,v}(x_0^-, z, \tau)$.
Dr.p	simulated values from $D_{1,v}(x_0^+, z, \tau) - D_{2,v}(x_0^+, z, \tau)$.
Dr.m	simulated values from $D_{1,v}(x_0^-, z, \tau) - D_{2,v}(x_0^-, z, \tau)$.
dz	the number of covariates
cov	either 0 or 1. Set *cov=1* if covariates are present in the model; otherwise set *cov=0*.
taus	a vector of quantiles of interest.
hh	the bandwidth values.
Qy.p	estimated conditional quantiles at (x_0^+, z).
Qy.m	estimated conditional quantiles at (x_0^-, z).
bias.p	estimated bias terms at (x_0^+, z).
bias.m	estimated bias terms at (x_0^-, z).
alpha	a number between 0 and 1, the desired significance level.
n.sim	the number of simulation repetitions.

Value

A list with elements:

qte QTE estimates without bias correction.

qte.r QTE estimates with bias correction.

uband uniform confidence band for QTE without bias correction.

uband.r uniform confidence band for QTE with robust bias correction.

sig standard errors for the bias-uncorrected QTE estimates.

sig.r standard errors for the bias-corrected QTE estimates. The values reflect the impact of the bias correction on the estimation precision.

See Also

rdq.band()

Examples

```
n = 500
x = runif(n,min=-4,max=4)
d = (x > 0)
y = x + 0.3*(x^2) - 0.1*(x^3) + 1.5*d + rnorm(n)
tlevel = seq(0.1,0.9,by=0.1)
tlevel2 = c(0.05,tlevel,0.95)
hh = rep(2,length(tlevel))
hh2 = rep(2,length(tlevel2))
sel = tlevel2 %in% tlevel

ab = rdq(y=y,x=x,d=d,x0=0,z0=NULL,tau=tlevel2,h.tau=hh2,cov=0)
delta = c(0.05,0.09,0.14,0.17,0.19,0.17,0.14,0.09,0.05)
fp = rdq.condf(x=x,Q=ab$qp.est,bcoe=ab$bcoe.p,taus=tlevel,taul=tlevel2,delta=delta,cov=0)
fm = rdq.condf(x=x,Q=ab$qm.est,bcoe=ab$bcoe.m,taus=tlevel,taul=tlevel2,delta=delta,cov=0)
bp = rdq.bias(y[d==1],x[d==1],dz=0,x0=0,z0=NULL,taus=tlevel,hh,hh,fx=fp$ff[(d==1),],cov=0)
bm = rdq.bias(y[d==0],x[d==0],dz=0,x0=0,z0=NULL,taus=tlevel,hh,hh,fx=fm$ff[(d==0),],cov=0)

sa = rdq.sim(x=x,d=d,x0=0,z0=NULL,dz=0,cov=0,tt=tlevel,hh,hh,fxp=fp$ff,fxm=fm$ff,n.sim=200)
ba = make.band(n,Dc.p=sa$dcp,Dc.m=sa$dcm,Dr.p=sa$drp,Dr.m=sa$drm,dz=0,cov=0,
taus=tlevel,hh,Qy.p=as.matrix(ab$qp.est[sel,]),Qy.m=as.matrix(ab$qm.est[sel,]),
bias.p=bp$bias,bias.m=bm$bias,alpha=0.1,n.sim=200)
```

---

make.band.cq *Uniform confidence bands for conditional quantile processes*

Description

make.band.cq constructs uniform confidence bands for conditional quantile processes as functions of tau for each side of the cutoff. See make.band as well. The function rdq.band calls this function to generates uniform bands for conditional quantiles.

Usage

```
make.band.cq(n.sam,Dc.p,Dc.m,Dr.p,Dr.m,dz,cov,taus,hh,Qy.p,Qy.m,
     bias.p,bias.m,alpha,n.sim)
```

Arguments

n.sam	the sample size.
Dc.p	simulated values from $D_{1,v}(x_0^+, z, \tau)$.
Dc.m	simulated values from $D_{1,v}(x_0^-, z, \tau)$.
Dr.p	simulated values from $D_{1,v}(x_0^+, z, \tau) - D_{2,v}(x_0^+, z, \tau)$.

`Dr.m`	simulated values from $D_{1,v}(x_0^-, z, \tau) - D_{2,v}(x_0^-, z, \tau)$.
`dz`	the number of covariates.
`cov`	either 0 or 1. Set *cov=1* if covariates are present in the model; otherwise set *cov=0*.
`taus`	a vector of quantiles of interest.
`hh`	the bandwidth values.
`Qy.p`	estimated conditional quantiles at (x_0^+, z).
`Qy.m`	estimated conditional quantiles at (x_0^-, z).
`bias.p`	estimated bias terms at (x_0^+, z).
`bias.m`	estimated bias terms at (x_0^-, z).
`alpha`	a number between 0 and 1, the desired significance level.
`n.sim`	the number of simulation repetitions.

Value

A list with elements:

qp conditional quantile estimates at x_0^+ (i.e., above the cutoff) without bias correction.

qp.r bias corrected conditional quantile estimates at x_0^+.

qm conditional quantile estimates at x_0^- (i.e., below the cutoff) without bias correction.

qm.r bias corrected conditional quantile estimates at x_0^-.

ubandp uniform confidence band for conditional quantiles at x_0^+ without bias correction.

ubandp.r uniform confidence band for conditional quantiles at x_0^+ with robust bias correction.

ubandm uniform confidence band for conditional quantiles at x_0^- without bias correction.

ubandm.r uniform confidence band for conditional quantiles at x_0^- with robust bias correction.

sp standard errors of the conditional quantile estimates without bias correction at x_0^+.

sp.r standard errors of the conditional quantile estimates with robust bias correction at x_0^+.

sm standard errors of the conditional quantile estimates without bias correction at x_0^-.

sm.r standard errors of the conditional quantile estimates with robust bias correction at x_0^-.

See Also

`make.band()`

Examples

```
n = 500
x = runif(n,min=-4,max=4)
d = (x > 0)
y = x + 0.3*(x^2) - 0.1*(x^3) + 1.5*d + rnorm(n)
tlevel = seq(0.1,0.9,by=0.1)
tlevel2 = c(0.05,tlevel,0.95)
hh = rep(2,length(tlevel))
hh2 = rep(2,length(tlevel2))
```

```
sel = tlevel2 %in% tlevel

ab = rdq(y=y,x=x,d=d,x0=0,z0=NULL,tau=tlevel2,h.tau=hh2,cov=0)
delta = c(0.05,0.09,0.14,0.17,0.19,0.17,0.14,0.09,0.05)
fp = rdq.condf(x=x,Q=ab$qp.est,bcoe=ab$bcoe.p,taus=tlevel,taul=tlevel2,delta,cov=0)
fm = rdq.condf(x=x,Q=ab$qm.est,bcoe=ab$bcoe.m,taus=tlevel,taul=tlevel2,delta,cov=0)
bp = rdq.bias(y[d==1],x[d==1],dz=0,x0=0,z0=NULL,taus=tlevel,hh,hh,fx=fp$ff[(d==1),],cov=0)
bm = rdq.bias(y[d==0],x[d==0],dz=0,x0=0,z0=NULL,taus=tlevel,hh,hh,fx=fm$ff[(d==0),],cov=0)

sa = rdq.sim(x=x,d=d,x0=0,z0=NULL,dz=0,cov=0,tt=tlevel,hh,hh,fxp=fp$ff,fxm=fm$ff,n.sim=200)
ba.cq = make.band.cq(n,Dc.p=sa$dcp,Dc.m=sa$dcm,Dr.p=sa$drp,Dr.m=sa$drm,dz=0,cov=0,
taus=tlevel,hh,Qy.p=as.matrix(ab$qp.est[sel,]),Qy.m=as.matrix(ab$qm.est[sel,]),
bias.p=bp$bias,bias.m=bm$bias,alpha=0.1,n.sim=200)
```

---

plot.qte *QTE plots*

Description

plot.qte generates plots summarizing the QTE estimates and their uniform confidence bands, helping users visualize the results. It also makes plots for conditional quantile processes for each side of the cutoff.

Usage

```
## S3 method for class 'qte'
plot(x, ptype = 1, ytext = NULL, mtext = NULL, subtext = NULL, ...)
```

Arguments

x	an object of class "qte" as produce by rd.qte.
ptype	either 1 or 2. Set *ptype=1* for the QTE plots, and *ptype=2* for the conditional quantile plots. The default value is 1.
ytext	the y-axis label.
mtext	the title of the plot.
subtext	the subtitles (used for the conditional quantile plots only).
...	optional arguments to plot

Value

plot(s) of the QTE estimates and uniform confidence bands.

Examples

```
# Without covariate
n = 500
x = runif(n,min=-4,max=4)
d = (x > 0)
y = x + 0.3*(x^2) - 0.1*(x^3) + 1.5*d + rnorm(n)
tlevel = seq(0.1,0.9,by=0.1)
A = rd.qte(y=y,x=x,d=d,x0=0,z0=NULL,tau=tlevel,bdw=2,cov=0,bias=1,cband=1,alpha=0.1)
plot(A)

y.text = "test scores"
m.text = "QTE and Uniform band"
plot(A,ytext=y.text,mtext=m.text)

z = sample(c(0,1),n,replace=TRUE)
y = x + 0.3*(x^2) - 0.1*(x^3) + 1.5*d + d*z + rnorm(n)
A = rd.qte(y=y,x=cbind(x,z),d=d,x0=0,z0=c(0,1),tau=tlevel,bdw=2,cov=1,bias=1,cband=1,alpha=0.1)

y.text = "test scores"
m.text = c("D=0","D=1")
plot(A,ytext=y.text,mtext=m.text)

# conditional quantile plots
n = 500
x = runif(n,min=-4,max=4)
d = (x > 0)
y = x + 0.3*(x^2) - 0.1*(x^3) + 1.5*d + rnorm(n)
tlevel = seq(0.1,0.9,by=0.1)
A = rd.qte(y=y,x=x,d=d,x0=0,z0=NULL,tau=tlevel,bdw=2,cov=0,bias=1,cband=1,alpha=0.1)

plot(A,ptype=2)

y.text = "test scores"
m.text = "Conditional quantile functions"
sub.text = c("D=0 group","D=1 group")
plot(A,ptype=2,ytext=y.text,mtext=m.text,subtext=sub.text)
```

`rd.qte` *QTE and its uniform confidence band.*

Description

`rd.qte` is the main function of the QTE.RD package. If *cov=1*, it estimates QTE for each subgroup defined by covariates. If *cov=0*, it estimate QTE without covariates. If *bias=1*, it corrects the bias in QTE estimates and obtains the robust confidence band and if *bias=0*, no bias correction is implemented. If *cband=1*, it provides a (1-alpha)100% uniform confidence bands, and if *cband=0*, it presents point estimates without confidence band.

Usage

```
rd.qte(y, x, d, x0, z0=NULL, tau, bdw, cov, bias, cband, alpha=NULL, print.qte=1)
```

Arguments

`y`	a numeric vector, the outcome variable.
`x`	a vector (or a matrix) of covariates, the first column is the running variable.
`d`	a numeric vector, the treatment status.
`x0`	the cutoff point.
`z0`	the value of the covariates at which to evaluate the effects. For example, if a female dummy z is included, z0 = 1 may indicate the female subgroup.
`tau`	a vector of quantiles of interest.
`bdw`	the bandwidth value(s). If 'bdw' is a scalar, it is interpreted as the bandwidth for the median. See the function `rdq.bandwidth` for how to select this bandwidth. The bandwidths for the rest of the quantiles are computed automatically using the formula of Yu and Jones (1998). If it is a vector with the same dimension as 'tau', the function will use these values for the respective quantiles accordingly.
`cov`	either 0 or 1. Set cov=1 when covariates are present in the model; otherwise set cov=0.
`bias`	either 0 or 1. If bias=1, the QTE estimate is bias corrected and the robust confidence band in Qu, Yoon, and Perron (2024) is produced. If bias=0, no bias correction is implemented.
`cband`	either 0 or 1. If cband=1, a uniform band is obtained; if cband=0, a point estimate is reported without confidence band.
`alpha`	a number between 0 and 1, the desired significance level. For example, when alpha=0.1, one will get a 90% uniform band.
`print.qte`	a logical flag specifying whether to print an outcome table.

Value

A list with elements:

qte QTE estimates.

uband uniform confidence band for QTE. If *bias=1*, the band is robust capturing the effect of the bias correction. If If *bias=0*, no bias correction is implemented.

sigma standard errors for each quantile level. If *bias=1*, its value captures the effect of the bias correction. If *bias=0*, no bias correction is implemented.

qp.est conditional quantile estimates on the right side of x_0 (or for the $D = 1$ group).

qm.est conditional quantile estimates on the left side of x_0 (or for the $D = 0$ group).

uband.p uniform confidence band for conditional quantiles on the right side of x_0.

uband.m uniform confidence band for conditional quantiles on the left side of x_0.

References

Zhongjun Qu, Jungmo Yoon, Pierre Perron (2024), "Inference on Conditional Quantile Processes in Partially Linear Models with Applications to the Impact of Unemployment Benefits," The Review of Economics and Statistics; https://doi.org/10.1162/rest_a_01168

Zhongjun Qu and Jungmo Yoon (2019), "Uniform Inference on Quantile Effects under Sharp Regression Discontinuity Designs," Journal of Business and Economic Statistics, 37(4), 625–647; https://doi.org/10.1080/07350015.2017.1407323

Keming Yu and M. C. Jones (1998), "Local Linear Quantile Regression," Journal of the American Statistical Association, 93(441), 228–237; https://doi.org/10.2307/2669619

Examples

```
# Without covariate
n = 500
x = runif(n,min=-4,max=4)
d = (x > 0)
y = x + 0.3*(x^2) - 0.1*(x^3) + 1.5*d + rnorm(n)
tlevel = seq(0.1,0.9,by=0.1)
A = rd.qte(y=y,x=x,d=d,x0=0,z0=NULL,tau=tlevel,bdw=2,cov=0,bias=1,cband=1,alpha=0.1)

# (continued) With covariates
z = sample(c(0,1),n,replace=TRUE)
y = x + 0.3*(x^2) - 0.1*(x^3) + 1.5*d + d*z + rnorm(n)
A = rd.qte(y=y,x=cbind(x,z),d=d,x0=0,z0=c(0,1),tau=tlevel,bdw=2,cov=1,bias=1,cband=1,alpha=0.1)
```

---

rdq — *Estimate the QTE under the RDD*

Description

`rdq` estimates QTE under the RDD with or without covariates. This function is used by `rd.qte` to generate QTE estimates.

Usage

```
rdq(y, x, d, x0, z0 = NULL, tau, h.tau, cov)
```

Arguments

`y`	a numeric vector, the outcome variable.
`x`	a vector (or a matrix) of covariates, the first column is the running variable.
`d`	a numeric vector, the treatment status.
`x0`	the cutoff point.

z0 — the value of the covariates at which to evaluate the effects. For example, if a female dummy is included, z0 = 1 may indicate the female subgroup.

tau — a vector of quantiles of interest.

h.tau — the bandwidth values (specified for each quantile level).

cov — either 0 or 1. Set *cov=1* if covariates are present in the model; otherwise set *cov=0*.

Value

A list with elements:

qte QTE estimates.

qp.est conditional quantile estimates on the right side of x_0 (or for the D=1 group).

qm.est conditional quantile estimates on the left side of x_0 (or for the D=0 group).

bcoe.p quantile regression coefficients on the right side of x_0.

bcoe.m quantile regression coefficients on the left side of x_0.

Examples

```
# Without covariate
n = 500
x = runif(n,min=-4,max=4)
d = (x > 0)
y = x + 0.3*(x^2) - 0.1*(x^3) + 1.5*d + rnorm(n)
tlevel = seq(0.1,0.9,by=0.1)
hh = rep(2,length(tlevel))
rdq(y=y,x=x,d=d,x0=0,z0=NULL,tau=tlevel,h.tau=hh,cov=0)

# (continued) With covariates
z = sample(c(0,1),n,replace=TRUE)
y = x + 0.3*(x^2) - 0.1*(x^3) + 1.5*d + d*z + rnorm(n)
rdq(y=y,x=cbind(x,z),d=d,x0=0,z0=c(0,1),tau=tlevel,h.tau=hh,cov=1)
```

---

rdq.band — *Uniform confidence bands for QTE*

Description

rdq.band produces uniform confidence bands for QTEs with and without bias correction. This function is used by rd.qte to generate uniform bands.

Usage

```
rdq.band(y, x, d, x0, z0 = NULL, tau, bdw, cov, alpha, print.qte = 1)
```

Arguments

`y`	a numeric vector, the outcome variable.
`x`	a vector (or a matrix) of covariates, the first column is the running variable.
`d`	a numeric vector, the treatment status.
`x0`	the cutoff point.
`z0`	the value of the covariates at which to evaluate the effects. For example, if a female dummy is included, z0 = 1 may indicate the female subgroup.
`tau`	a vector of quantiles of interest.
`bdw`	the bandwidth value(s). If 'bdw' is a scalar, it is interpreted as the bandwidth for the median. The bandwidths for the rest of the quantiles are computed automatically using the formula in Yu and Jones (1998). If it is a vector with the same dimension as 'tau', the function will use these values for the respective quantiles accordingly.
`cov`	either 0 or 1. Set cov=1 when covariates are present in the model; otherwise set cov=0.
`alpha`	a number between 0 and 1, the desired significance level.
`print.qte`	a logical flag specifying whether to print an outcome table.

Value

qte QTE estimates without bias correction.

qte.cor bias corrected QTE estimates.

uband uniform confidence band for QTE without bias correction.

uband.robust uniform confidence band for QTE with robust bias correction.

sig standard errors for each quantile level for estimates without bias correction.

sig.r standard errors for each quantile level for estimates with robust bias correction.

uband.p uniform confidence band for the conditional quantile estimates on the right side of the cutoff, without bias correction.

uband.robust.p uniform confidence band for the conditional quantile estimates on the right side of the cutoff, robust to the bias correction.

uband.m uniform confidence band for the conditional quantile estimates on the left side of the cutoff, without bias correction.

uband.robust.m uniform confidence band for the conditional quantile estimates on the left side of the cutoff, robust to the bias correction.

References

Zhongjun Qu, Jungmo Yoon, Pierre Perron (2024), "Inference on Conditional Quantile Processes in Partially Linear Models with Applications to the Impact of Unemployment Benefits," The Review of Economics and Statistics; https://doi.org/10.1162/rest_a_01168

Zhongjun Qu and Jungmo Yoon (2019), "Uniform Inference on Quantile Effects under Sharp Regression Discontinuity Designs," Journal of Business and Economic Statistics, 37(4), 625–647; https://doi.org/10.1080/07350015.2017.1407323

Keming Yu and M. C. Jones (1998), "Local Linear Quantile Regression," Journal of the American Statistical Association, 93(441), 228–237; https://doi.org/10.2307/2669619

See Also

rd.qte()

Examples

```
# Without covariate
n = 500
x = runif(n,min=-4,max=4)
d = (x > 0)
y = x + 0.3*(x^2) - 0.1*(x^3) + 1.5*d + rnorm(n)
tlevel = seq(0.1,0.9,by=0.1)
D = rdq.band(y=y,x=x,d=d,x0=0,z0=NULL,tau=tlevel,bdw=2,cov=0,alpha=0.1)

# (continued) With covariates
z = sample(c(0,1),n,replace=TRUE)
y = x + 0.3*(x^2) - 0.1*(x^3) + 1.5*d + d*z + rnorm(n)
D = rdq.band(y=y,x=cbind(x,z),d=d,x0=0,z0=c(0,1),tau=tlevel,bdw=2,cov=1,alpha=0.1)
```

---

rdq.bandwidth *Bandwidth estimation*

Description

rdq.bandwidth implements two bandwidth selection rules and obtains the cross-validation (CV) bandwidth and the MSE optimal bandwidth.

Usage

```
rdq.bandwidth(y, x, d, x0, z0=NULL, cov, cv, val,hp=NULL,pm.each=1,
     bdy=1,p.order=1,xl=0.5,print.qte=1)
```

Arguments

y	a numeric vector, the outcome variable.
x	a vector (or a matrix) of covariates, the first column is the running variable.
d	a numeric vector, the treatment status.
x0	the cutoff point.
z0	the value of the covariates at which to evaluate the effects.
cov	either 0 or 1. Set *cov=1* when covariates are present in the model; otherwise set *cov=0*.
cv	either 0 or 1. When *cv=1*, both the CV and MSE optimal bandwidths are produced. When *cv=0*, the MSE optimal bandwidth is produced.
val	a set of candidate values for the CV bandwidth.

hp	a pilot bandwidth to estimate nuisance parameters for the MSE optimal bandwidth. It will be used only if *cv=0*. If *cv=1*, the CV bandwidth will be used as the pilot bandwidth to compute the MSE optimal bandwidth.
pm.each	either 0 or 1. When *pm.each=1*, the CV bandwidths for each side of the cutoff will be obtained separately.
bdy	either 0 or 1. When *bdy=1*, the CV bandwidth uses the boundary point procdure.
p.order	either 1 or 2. When *p.order=1*, a local linear regression is used, and when *p.order=2*, a local quadratic regression is used.
xl	if *xl=0.5*, the CV bandwidth use the 50% of observations closest to x_0.
print.qte	a logical flag specifying whether to print an outcome table.

Value

A list with elements:

cv the selected CV bandwidth at the median.

opt.p the MSE optimal bandwidth at the median from the right side of x_0.

opt.m the MSE optimal bandwidth at the median from the left side of x_0.

References

Zhongjun Qu, Jungmo Yoon, Pierre Perron (2024), "Inference on Conditional Quantile Processes in Partially Linear Models with Applications to the Impact of Unemployment Benefits," The Review of Economics and Statistics; https://doi.org/10.1162/rest_a_01168

Zhongjun Qu and Jungmo Yoon (2019), "Uniform Inference on Quantile Effects under Sharp Regression Discontinuity Designs," Journal of Business and Economic Statistics, 37(4), 625–647; https://doi.org/10.1080/07350015.2017.1407323

Examples

```
# Without covariate
n = 500
x = runif(n,min=-4,max=4)
d = (x > 0)
y = x + 0.3*(x^2) - 0.1*(x^3) + 1.5*d + rnorm(n)
tlevel = seq(0.1,0.9,by=0.1)
rdq.bandwidth(y=y,x=x,d=d,x0=0,z0=NULL,cov=0,cv=1,val=(1:4))
rdq.bandwidth(y=y,x=x,d=d,x0=0,z0=NULL,cov=0,cv=0,val=(1:4),hp=2)

# (continued) With covariates
z = sample(c(0,1),n,replace=TRUE)
y = x + 0.3*(x^2) - 0.1*(x^3) + 1.5*d + d*z + rnorm(n)
rdq.bandwidth(y=y,x=cbind(x,z),d=d,x0=0,z0=c(0,1),cov=1,cv=1,val=(1:4),bdy=1,p.order=1)
```

`rdq.bias` *Bias estimation*

Description

`rdq.bias` estimates the bias terms using the local quadratic quantile regression.

Usage

```
rdq.bias(y, x, dz, x0, z0, taus, h.tau, h.tau2, fx, cov)
```

Arguments

`y`	a numeric vector, the outcome variable.
`x`	a vector (or a matrix) of covariates, the first column is the running variable.
`dz`	the number of covariates.
`x0`	the cutoff point.
`z0`	the value of the covariates at which to evaluate the effects.
`taus`	a vector of quantiles of interest.
`h.tau`	the bandwidth values (specified for each quantile level), for estimating conditional quantiles.
`h.tau2`	the bandwidth values for the local quadratic quantile regression, for estimating the bias terms.
`fx`	conditional density estimates.
`cov`	either 0 or 1. Set *cov=1* if covariates are present in the model; otherwise set *cov=0*.

Value

A list with elements:

bias the bias estimates.

b.hat the estimate of the $B_v(x, z, \tau)$ term. See Qu, Yoon, and Perron (2024).

References

Zhongjun Qu, Jungmo Yoon, Pierre Perron (2024), "Inference on Conditional Quantile Processes in Partially Linear Models with Applications to the Impact of Unemployment Benefits," The Review of Economics and Statistics; https://doi.org/10.1162/rest_a_01168

Examples

```
n = 500
x = runif(n,min=-4,max=4)
d = (x > 0)
y = x + 0.3*(x^2) - 0.1*(x^3) + 1.5*d + rnorm(n)
tlevel = seq(0.1,0.9,by=0.1)
tlevel2 = c(0.05,tlevel,0.95)
hh = rep(2,length(tlevel))
hh2 = rep(2,length(tlevel2))

ab = rdq(y=y,x=x,d=d,x0=0,z0=NULL,tau=tlevel2,h.tau=hh2,cov=0)
delta = c(0.05,0.09,0.14,0.17,0.19,0.17,0.14,0.09,0.05)
hh = rep(2,length(tlevel))
fe = rdq.condf(x,Q=ab$qp.est,bcoe=ab$bcoe.p,taus=tlevel,taul=tlevel2,delta=delta,cov=0)
be = rdq.bias(y[d==1],x[d==1],dz=0,x0=0,z0=NULL,taus=tlevel,hh,hh,fx=fe$ff[(d==1),],cov=0)
```

`rdq.condf` *Conditional density estimation*

Description

`rdq.condf` estimates conditional density functions by using the differencing method.

Usage

```
rdq.condf(x, Q, bcoe, taus, taul, delta, cov)
```

Arguments

`x`	a vector (or a matrix) of covariates.
`Q`	a vector of estimated conditional quantiles.
`bcoe`	quantile regression coefficient estimates.
`taus`	a vector of quantiles of interest.
`taul`	a vector of quantiles used for the conditional density estimation. It is needed to estimate the tail parts of conditional density functions more precisely.
`delta`	bandwidths for estimating the conditional density.
`cov`	either 0 or 1. Set *cov=1* if covariates are present in the model; otherwise set *cov=0*.

Value

conditional density function estimates

Examples

```
n = 500
x = runif(n,min=-4,max=4)
d = (x > 0)
y = x + 0.3*(x^2) - 0.1*(x^3) + 1.5*d + rnorm(n)
tlevel = seq(0.1,0.9,by=0.1)
hh = rep(2,length(tlevel))

ab = rdq(y=y,x=x,d=d,x0=0,z0=NULL,tau=tlevel,h.tau=hh,cov=0)
delta = 0.186
fe = rdq.condf(x=x,Q=ab$qp.est,bcoe=ab$bcoe.p,taus=0.5,taul=tlevel,delta=delta,cov=0)
```

`rdq.sim` *Simulation the asymptotic distributions*

Description

`rdq.sim` produces iid draws from the asymptotic distribution of the conditional quantile process estimate.

Usage

```
rdq.sim(x, d, x0, z0, dz, cov, tt, hh, hh2, fxp, fxm, n.sim)
```

Arguments

`x`	a vector (or a matrix) of covariates.
`d`	a numeric vector, the treatment status.
`x0`	the cutoff point.
`z0`	the value of the covariates at which to evaluate the effects.
`dz`	the number of covariates.
`cov`	either 0 or 1. Set *cov=1* if covariates are present in the model; otherwise set *cov=0*.
`tt`	a vector of quantiles.
`hh`	the bandwidth values (specified for each quantile level).
`hh2`	the bandwidth values for the local quadratic quantile regression.
`fxp`	conditional density estimates on the right side of x_0.
`fxm`	conditional density estimates on the left side of x_0.
`n.sim`	the number of simulation repetitions.

Value

A list with elements:

dcp realizations from the asymptotic distribution of the conditional quantile process, from the right side of x_0.

dcm realizations from the asymptotic distribution of the conditional quantile process, from the left side of x_0.

drp realizations from the asymptotic distribution of the bias corrected conditional quantile process, from the right side of x_0.

drm realizations from the asymptotic distribution of the bias corrected conditional quantile process, from the left side of x_0.

Examples

```
n = 500
x = runif(n,min=-4,max=4)
d = (x > 0)
y = x + 0.3*(x^2) - 0.1*(x^3) + 1.5*d + rnorm(n)
tlevel = seq(0.1,0.9,by=0.1)
tlevel2 = c(0.05,tlevel,0.95)
hh = rep(2,length(tlevel))
hh2 = rep(2,length(tlevel2))

ab = rdq(y=y,x=x,d=d,x0=0,z0=NULL,tau=tlevel2,h.tau=hh2,cov=0)
delta = c(0.05,0.09,0.14,0.17,0.19,0.17,0.14,0.09,0.05)
fp = rdq.condf(x=x,Q=ab$qp.est,bcoe=ab$bcoe.p,taus=tlevel,taul=tlevel2,delta,cov=0)
fm = rdq.condf(x=x,Q=ab$qm.est,bcoe=ab$bcoe.m,taus=tlevel,taul=tlevel2,delta,cov=0)
sa = rdq.sim(x=x,d=d,x0=0,z0=NULL,dz=0,cov=0,tt=tlevel,hh,hh,fxp=fp$ff,fxm=fm$ff,n.sim=200)
```

---

rdq.test *tests for QTE*

Description

`rdq.test` provides testing results for hypotheses on the treatment effects concerning (i) treatment significance, (ii) homogeneity of effects over quantiles, and (iii) positive or negative dominance hypothesis.

Usage

```
rdq.test(y,x,d,x0,z0=NULL,tau,bdw,cov,bias,alpha,type,std.opt=1,print.qte=1)
```

Arguments

`y`	a numeric vector, the outcome variable.
`x`	a vector (or a matrix) of covariates, the first column is the running variable.
`d`	a numeric vector, the treatment status.
`x0`	the cutoff point.
`z0`	the value of the covariates at which to evaluate the effects. For example, if a female dummy is included, z0 = 1 indicates the female subgroup.
`tau`	a vector of quantiles of interest.
`bdw`	the bandwidth value(s). If 'bdw' is a scalar, it is interpreted as the bandwidth for the median. The bandwidths for the rest of the quantiles are computed automatically using the formula in Yu and Jones (1998). If it is a vector with the same dimension as 'tau', the function will use these values for the respective quantiles accordingly.
`cov`	either 0 or 1. Set cov=1 when covariates are present in the model; otherwise set cov=0.
`bias`	either 0 or 1. If bias=1, the QTE estimate is bias corrected and the robust confidence band in Qu, Yoon, and Perron (2024) is produced. If bias=0, no bias correction is implemented.
`alpha`	a number between 0 and 1, the desired significance level. When alpha=0.1, one will get a 90% uniform band.
`type`	a value in 1–4. Set *type* to 1 to test the null hypothesis of a zero treatment effect against the alternative hypothesis of significant treatment effects; set *type* to 2 to test the null hypothesis of homogeneous treatment against heterogeneous treatment effects; set *type* to 3 to test the null hypothesis of uniformly non-negative treatment effects against the presence of negative effects; and set *type* to 4 to test the null hypothesis of uniformly non-positive treatment effects against the presence of positive effects at some quantiles.
`std.opt`	either 0 or 1. If *std.opt=1*, the test statistic is standardized so that the variance is equalized across quantiles; if *std.opt=0*, the test is not standardized.
`print.qte`	a logical flag specifying whether to print an outcome table.

Value

A list with elements:

te.st test statistics

cr.va critical values.

References

Zhongjun Qu, Jungmo Yoon, Pierre Perron (2024), "Inference on Conditional Quantile Processes in Partially Linear Models with Applications to the Impact of Unemployment Benefits," The Review of Economics and Statistics; https://doi.org/10.1162/rest_a_01168

Zhongjun Qu and Jungmo Yoon (2019), "Uniform Inference on Quantile Effects under Sharp Regression Discontinuity Designs," Journal of Business and Economic Statistics, 37(4), 625–647; https://doi.org/10.1080/07350015.2017.1407323

Examples

```
# Without covariate
n = 500
x = runif(n,min=-4,max=4)
d = (x > 0)
y = x + 0.3*(x^2) - 0.1*(x^3) + 1.5*d + rnorm(n)
tlevel = seq(0.1,0.9,by=0.1)
B = rdq.test(y=y,x=x,d=d,x0=0,z0=NULL,tau=tlevel,bdw=2,cov=0,bias=1,alpha=c(0.1,0.05),type=c(1,2,3))

# (continued) With covariates
z = sample(c(0,1),n,replace=TRUE)
y = x + 0.3*(x^2) - 0.1*(x^3) + 1.5*d + d*z + rnorm(n)
B = rdq.test(y=y,x=cbind(x,z),d=d,x0=0,z0=c(0,1),tau=tlevel,bdw=2,cov=1,bias=1,
alpha=c(0.1,0.05),type=c(3,4))
```

`run.test` *Run tests*

Description

`run.test` performs hypothesis testing. The function `rdq.test` calls this function to run tests.

Usage

```
run.test(n.sam,dz,taus,hh,Dc.p,Dc.m,Dr.p,Dr.m,Qy.p,Qy.m,bias.p,bias.m,
     cov,bias,alpha,n.sim,test.type,std.opt)
```

Arguments

`n.sam`	the sample size.
`dz`	the number of covariates.
`taus`	a vector of quantiles of interest.
`hh`	the bandwidth values.
`Dc.p`	simulated values from $D_{1,v}(x_0^+, z, \tau)$.
`Dc.m`	simulated values from $D_{1,v}(x_0^-, z, \tau)$.
`Dr.p`	simulated values from $D_{1,v}(x_0^+, z, \tau) - D_{2,v}(x_0^+, z, \tau)$.
`Dr.m`	simulated values from $D_{1,v}(x_0^-, z, \tau) - D_{2,v}(x_0^-, z, \tau)$.
`Qy.p`	estimated conditional quantiles at (x_0^+, z).
`Qy.m`	estimated conditional quantiles at (x_0^-, z).
`bias.p`	estimated bias terms at (x_0^+, z).
`bias.m`	estimated bias terms at (x_0^-, z).
`cov`	either 0 or 1. Set *cov=1* if covariates are present in the model; otherwise set *cov=0*.

`bias`	either 0 or 1. If *bias=1*, the QTE estimate is bias corrected and the robust confidence band in Qu, Yoon, and Perron (2024) is produced. If *bias=0*, no bias correction is implemented.
`alpha`	a number between 0 and 1, the desired significance level.
`n.sim`	the number of simulation repetitions.
`test.type`	a value in 1–4. Set *type* to 1 to test the null hypothesis of a zero treatment effect against the alternative hypothesis of significant treatment effects; set *type* to 2 to test the null hypothesis of homogeneous treatment against heterogeneous treatment effects; set *type* to 3 to test the null hypothesis of uniformly non-negative treatment effects against the presence of negative effects; and set *type* to 4 to test the null hypothesis of uniformly non-positive treatment effects against the presence of positive effects at some quantiles.
`std.opt`	either 0 or 1. If *std.opt=1*, the test statistic is standardized so that the variance is equalized across quantiles; if *std.opt=0*, the test is not standardized.

Value

A list with elements:

test.stat test statistics

cr.value critical values.

References

Zhongjun Qu, Jungmo Yoon, Pierre Perron (2024), "Inference on Conditional Quantile Processes in Partially Linear Models with Applications to the Impact of Unemployment Benefits," The Review of Economics and Statistics; https://doi.org/10.1162/rest_a_01168

See Also

`rdq.test()`

Examples

```
n = 500
x = runif(n,min=-4,max=4)
d = (x > 0)
y = x + 0.3*(x^2) - 0.1*(x^3) + 1.5*d + rnorm(n)
tlevel = seq(0.1,0.9,by=0.1)
tlevel2 = c(0.05,tlevel,0.95)
hh = rep(2,length(tlevel))
hh2 = rep(2,length(tlevel2))
sel = tlevel2 %in% tlevel

ab = rdq(y=y,x=x,d=d,x0=0,z0=NULL,tau=tlevel2,h.tau=hh2,cov=0)
delta = c(0.05,0.09,0.14,0.17,0.19,0.17,0.14,0.09,0.05)
fp = rdq.condf(x=x,Q=ab$qp.est,bcoe=ab$bcoe.p,taus=tlevel,taul=tlevel2,delta,cov=0)
fm = rdq.condf(x=x,Q=ab$qm.est,bcoe=ab$bcoe.m,taus=tlevel,taul=tlevel2,delta,cov=0)
bp = rdq.bias(y[d==1],x[d==1],dz=0,x0=0,z0=NULL,taus=tlevel,hh,hh,fx=fp$ff[(d==1),],cov=0)
bm = rdq.bias(y[d==0],x[d==0],dz=0,x0=0,z0=NULL,taus=tlevel,hh,hh,fx=fm$ff[(d==0),],cov=0)
```

```
sa = rdq.sim(x=x,d=d,x0=0,z0=NULL,dz=0,cov=0,tt=tlevel,hh,hh,fxp=fp$ff,fxm=fm$ff,n.sim=200)
bt <- run.test(n,dz=0,taus=tlevel,hh,Dc.p=sa$dcp,Dc.m=sa$dcm,Dr.p=sa$drp,Dr.m=sa$drm,
Qy.p=as.matrix(ab$qp.est[sel,]),Qy.m=as.matrix(ab$qm.est[sel,]),bias.p=bp$bias,bias.m=bm$bias,
cov=0,bias=1,alpha=0.1,n.sim=200,test.type=1,std.opt=1)
```

Index