Package ‘EXRQ’

January 20, 2025

Type Package

Title Extreme Regression of Quantiles

Version 1.0

Date 2016-07-06

Author Huixia Judy Wang

Maintainer Huixia Judy Wang <judywang@gwu.edu>

Description Estimation for high conditional quantiles based on quantile regression.

Depends quantreg, mnormt

License GPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2016-07-06 23:48:41

Contents

`est.gamma.func` *Estimation of the Extreme Value Index on the Original Scale*

Description

This function estimates the extreme value index on the original scale based on the estimated intermediate conditional quantiles on the transformed scale

Usage

```
est.gamma.func(taus, Lam.Q, lam, a = 0, tol)
```

Arguments

`taus`	a grid of intermediate high quantile levels
`Lam.Q`	a vector of the same length as taus, representing the estimated intermediate conditional quantiles of Y (at taus) on the transformed scale
`lam`	the power-transformation parameter
`a`	location shift parameter in the power transformation (introduced to avoid negative y values)
`tol`	the tolerance level for checking quantile crossing issue

Value

A list is returned with the following components.

gamma.x: the estimated EVI. If quantile crossing is too severe, which suggests that the estimated intermediate conditional quantiles are unstable, then NA is returned.

Q: the estimated conditional quantile of Y on the original scale

---

`Estc.func` *Estimation of the C vector*

Description

This function estimates the C vector involved in the function test.EVI for testing the constancy of EVI

Usage

```
Estc.func(y, x, tau = 0.99, M = 2)
```

Arguments

y	a vector of n untransformed responses
x	a n x p matrix of n observations and p predictors
tau	an upper quantile level close to one
M	a constant larger than one that is used for estimating the c vector and thus K(x) function. The default is two

Value

A p-dimensional vector is returned.

EVI.CFG.func — *Hill Estimator of the Extreme Value Index*

Description

Hill Estimator of the Extreme Value Index

Usage

```
EVI.CFG.func(x, tol = 1e-04, min.prop = 0.3, taus)
```

Arguments

x	the estimated quantiles at intermediate quantile levels
tol	the tolerance level used for checking quantile crossing
min.prop	the minimum proportion of quantiles that are estimated higher than the adjacent lower quantiles
taus	the corresponding quantile levels

Details

The function estimates the extreme value index using Hill estimator based on the estimated intermediate quantiles.

Value

The estimated extreme value index is returned. If the proportion of cases with quantile crossing is too high, an NA is returned.

References

Chernozhukov, C., Fernandez-Val, I., and Galichon, A. (2010). Quantile and probability curves without crossing. Econometrica, 78, 1093-1125.

PowT.1tau.func *Estimation for Quantile Power Transformation Model*

Description

This function estimates the power transformation parameter at a single given quantile level

Usage

```
PowT.1tau.func(y, x, tau, lams = seq(-2, 2, 0.1), a)
```

Arguments

y	a vector of length n representing the response
x	a n x p matrix of n observations and p predictors
tau	the quantile level of interest
lams	a set of transformation parameters for grid search
a	the location shift

Details

This function estimates the transformation parameter lam following the estimation method in Mu and He (2007) such that the conditional quantile of the transformed response is linear in covariates. The transformed response is defined as

$$\Lambda(y) = (y + a)^{\lambda} - 1\lambda, if\lambda \neq 0; = log(y + a)if\lambda = 0.$$

Value

A list is returned with the following components

lam: the estimated transformation parameter

coef: the estimated quantile coefficient from the power-transformed linear quantile regression

References

Mu, Y. and He, X. (2007). Power transformation toward a linear regression quantile. Journal of the American Statistical Association, 102, 269-279.

`qpareto` *Quantile of the Pareto Distribution*

Description

Quantile of the Pareto Distribution

Usage

```
qpareto(p, gamma)
```

Arguments

`p`	the quantile level
`gamma`	the shape parameter

Value

the pth quantile

`rpareto` *Random Generation for the Pareto Distribution*

Description

Random Generation for the Pareto Distribution

Usage

```
rpareto(n, gamma)
```

Arguments

`n`	number of observations
`gamma`	the shape parameter

Value

a vector of n i.i.d. random variables from the Pareto distribution `select.k.func` *Selection of the Tuning Parameter k*

Description

This function selects the tuning parameter k, the number of upper order statistics involved in Hill estimator of EVI among a grid of points following the method described in Section 3.3 of Wang and Li (2013). The method selects k as the value that minimizes the discrepancy between the estimated x-dependent EVI on the transformed scale and lam times the estimated x-dependent EVI on the original scale

Usage

```
select.k.func(y, x, Lam.y, lam, a, max.tau, grid.k, n)
```

Arguments

`y`	a vector of n untransformed responses
`x`	a n x p matrix of n observations and p predictors
`Lam.y`	a vector of n power-transformed responses
`lam`	the power-transformation parameter
`a`	location shift parameter in the power transformation (introduced to avoid negative y values)
`max.tau`	the upper bound of the intermediate quantile levels
`grid.k`	the grid for the number of upper order statistics involved in Hill estimator
`n`	the number of observations

Value

the selected k is returned

References

Wang, H. and Li, D. (2013). Estimation of conditional high quantiles through power transformation. Journal of the American Statistical Association, 108, 1062-1074.

testC.EVI *Testing the Constancy of EVI Over Covariates*

Description

This function tests whether the extreme value index of Y, gamma(x), is constant or varying across the covariate x by using the test procedure described in Section 3.4 of Wang and Li (2013).

Usage

```
testC.EVI(y, x, grid.lam = seq(-2, 2, 0.1), grid.k, tau.lam = 0.9,
  u.x = 0, a = 0, M = 2, tol = 1e-04)
```

Arguments

`y`	a vector of n untransformed responses
`x`	a n x p matrix of n observations and p predictors
`grid.lam`	a grid of points for power-transformation parameter
`grid.k`	a grid of points for k, the number of upper order statistics involved in Hill estimator
`tau.lam`	the quantile level used for estimating the transformation parameter
`u.x`	the proportion to be trimmed in the x direction
`a`	location shift parameter in the power transformation (introduced to avoid negative y values)
`M`	a constant larger than one that is used for estimating the c vector and thus K(x) function. The default is two
`tol`	the tolerance level for checking quantile crossing issue

Value

A list is returned with the following components

lam: the estimated power-transformation parameter

k: the selected tuning parameter k, the number of upper order statistics involved in Hill estimator

Tm: the proposed test statistic

scaledTm: the standardized test statistic

pval.iid: the p-value based on iid assumption, that is, assuming that K(x)=1

pval.nid: the p-value based on estimated K(x)=(X'C)^(1/EVI)

gamma.bar: the pooled EVI estimator

hat.gamma: a N-dimensional vector consisting of the estimated x-dependent EVI at x=xstar

xstar: a N x p matrix of N observations and p predictors

References

Wang, H. and Li, D. (2013). Estimation of conditional high quantiles through power transformation. Journal of the American Statistical Association, 108, 1062-1074.

Examples

```
library(EXRQ)
n=500
tau.e = c(0.99, 0.993, 0.995)
set.seed(12368819)
x1 = runif(n, -1, 1)
x2 = runif(n, -1, 1)
sqrty = 2 + x1 + x2 + (1+0.8*x1)*rpareto(n, 0.5)
x = as.matrix(cbind(x1, x2))
y = sqrty^2
out = testC.EVI(y, x, grid.lam=seq(-0.5, 1.5, 0.1), grid.k=50, tau.lam=0.9)
(Tval = out$scaledTm)
(pval.iid = out$pval.iid)
(pval.nid = out$pval.nid)
```

ThreeStage *Three-Stage Extreme Conditional Quantile Estimator*

Description

Provides the estimation of extreme conditional quantile using the three-stage estimation method in Wang and Li (2013). Specifically the function estimates the tau.e-th conditional quantile of Y given x=xstar based on the power-transformed quantile regression model and extreme value theory. The method is based on Hill estimator for the extreme value index and works for heavy-tailed distributions (on the original scale).

Usage

```
ThreeStage(y, x, xstar, tau.e, grid.lam = seq(-2, 2, 0.1), grid.k, tau.lam,
  a = 0, tol = 1e-04)
```

Arguments

y	a vector of n responses
x	a n x p matrix of n observations and p predictors
xstar	a m x p matrix of m observations and p predictors
tau.e	the extreme quantile level of interest
grid.lam	the set of lambda (transformation parameter) values for grid search
grid.k	the grid for the number of upper order statistics involved in Hill estimator; used for searching for the data-adaptive k. If the lenfth of grid.k is 1, then k is fixed at grid.k and no selection is performed.

tau.lam — the quantile level used for estimating the transformation parameter

a — location shift parameter in the power transformation (introduced to avoid negative y values)

tol — the tolerance level for checking quantile crossing issue

Value

A list is returned with the following components

lam: the estimated power-transformation parameter

k: the selected k, the number of upper order statistics involved in Hill estimator

gamma.x: the estimated x-dependent extreme value index (EVI)

cgmma: the pooled EVI estimation

Q3Stage: the three-stage estimator of the tau.e-th conditional quantile of Y given xstar based on the x-dependent EVI estimation

Q3StageP: the three-stage estimator of the tau.e-th conditional quantile of Y given xstar based on the pooled EVI estimation

References

Wang, H. and Li, D. (2013). Estimation of conditional high quantiles through power transformation. Journal of the American Statistical Association, 108, 1062-1074.

See Also

TwoStage

Examples

```
#A simulation example (sqrt transformation, heteroscedastic error)
library(EXRQ)
n=500
tau.e = c(0.99, 0.993, 0.995)
set.seed(12368819)
x1 = runif(n, -1, 1)
x2 = runif(n, -1, 1)
sqrty = 2 + x1 + x2 + (1+0.8*x1)*rpareto(n, 0.5)
x = as.matrix(cbind(x1, x2))
y = sqrty^2
xstar = rbind(c(-0.5,0),c(0,-0.5),c(0,0),c(0.5,0),c(0,0.5))
## 3Stage estimator
out.3stage <- ThreeStage(y, x, xstar, tau.e, grid.lam=seq(-0.5, 1.5, 0.1), grid.k=50, tau.lam=0.9)
```

TwoStage *Two-Stage Extreme Conditional Quantile Estimator*

Description

This function provides the Two-Stage estimator in Wang, Li and He (2012) for conditional extreme quantiles based on covariate-dependent extreme value index estimation. The intermediate conditional quantile is estimated by quantile regression of the response on the original scale without any transformation. The method is based on Hill estimator for the extreme value index and works for heavy-tailed distributions.

Usage

```
TwoStage(y, x, xstar, tau.e, k, tol = 1e-04)
```

Arguments

`y`	a vector of length n representing the response
`x`	a n x p matrix of n observations and p predictors
`xstar`	a m x p matrix of m observations and p predictors representing the covariate of interest
`tau.e`	the extreme quantile level of interest
`k`	the number of upper order statistics used in Hill estimator
`tol`	the tolerance level used for checking quantile crossing

Value

A list of the following commponents is returned

Q2Stage: the estimated (extrapolated) conditional extreme quantile of the response given x=xstar at the quantile level tau.e

gamma.x: the estimated covariate-dependent extreme value index (Hill estimator associated with x=xstar)

References

Wang, H., Li, D., and He, X. (2012). Estimation of high conditional quantiles for heavytailed distributions, Journal of the American Statistical Association, 107, 1453-1464.

See Also

ThreeStage

Examples

```
#A simulation example (sqrt transformation, heteroscedastic error)
library(EXRQ)
n=500
tau.e = c(0.99, 0.993, 0.995)
set.seed(12368819)
x1 = runif(n, -1, 1)
x2 = runif(n, -1, 1)
sqrty = 2 + x1 + x2 + (1+0.8*x1)*rpareto(n, 0.5)
x = as.matrix(cbind(x1, x2))
y = sqrty^2
xstar = rbind(c(-0.5,0),c(0,-0.5),c(0,0),c(0.5,0),c(0,0.5))
## 2Stage method in Wang, Li and He (2012), no transformation
out.2stage <- TwoStage(y, x, xstar, tau.e, k=50)
```

Index