Package ‘pqrfe’

December 1, 2022

Type Package

Title Penalized Quantile Regression with Fixed Effects

Version 1.1

Date 2022-12-01

Description

Quantile regression with fixed effects is a general model for longitudinal data. Here we proposed to solve it by several methods. The estimation methods include three loss functions as check, asymmetric least square and asymmetric Huber functions; and three structures as simple regression, fixed effects and fixed effects with penalized intercepts by LASSO.

License GPL (>= 2)

Imports Rcpp (>= 1.0.5), MASS (>= 7.3-49)

LinkingTo Rcpp, RcppArmadillo

Suggests tinytest (>= 1.3.1)

RoxygenNote 7.2.1

NeedsCompilation yes

Author Ian Meneghel Danilevicz [aut, cre]
(<https://orcid.org/0000-0003-4541-0524>),
Valderio A Reisen [aut],
Pascal Bondon [aut]

Maintainer Ian Meneghel Danilevicz <iandanilevicz@gmail.com>

Repository CRAN

Date/Publication 2022-12-01 11:20:02 UTC

R topics documented:

`pqrfe-package` *Penalized Quantile Regression with Fixed Effects*

Description

Quantile regression with fixed effects is a general model for longitudinal data. Here we proposed to solve it by several methods. The estimation methods include three loss functions as check, asymmetric least square and asymmetric Huber functions; and three structures as simple regression, fixed effects and fixed effects with penalized intercepts by LASSO.

Package Content

Index of help topics:

```
check_lambda            check lambda
choice_p                choice model
clean_data              Clean missings
d_psi_als               D Psi ALS
d_psi_mq                D Psi M-quantile
f_den                   Kernel density
f_tab                   Tabular function
loss_er                 Loss expectile regression
loss_erfe               Loss expectile regression with fixed effects
loss_erlasso            Loss lasso expectile regression with fixed
                        effects
loss_mqr                Loss M-quantile regression
loss_mqrfe              Loss M-quantile regression with fixed effects
loss_mqrlasso           Loss lasso M-quantile regression with fixed
                        effects
loss_qr                 Loss quantile regression
loss_qrfe               Loss quantile regression with fixed effects
loss_qrlasso            Loss lasso quantile regression with fixed
                        effects
mpqr                    Multiple penalized quantile regression
optim_er                optim expectile regression
optim_erfe              optim expectile regression with fixed effects
optim_erlasso           optim expectile regression with fixed effects
                        and LASSO
optim_mqr               optim M-quantile regression
optim_mqrfe             optim quantile regression with fixed effects
optim_mqrlasso          optim M-quantile regression with fixed effects
                        and LASSO
optim_qr                optim quantile regression
optim_qrfe              optim quantile regression with fixed effects
optim_qrlasso           optim quantile regression with fixed effects
                        and LASSO
plot_taus               Plot multiple penalized quantile regression
pqr                     Penalized quantile regression with fixed
                        effects
pqrfe-package           Penalized Quantile Regression with Fixed
                        Effects
print.PQR               Print an PQR
psi_als                 Psi ALS
psi_mq                  Psi M-quantile
q_cov                   Covariance
rho_koenker             Rho Koenker
rho_mq                  Rho M-quantile
sgf                     Identify significance
```

Maintainer

NA

Author(s)

NA

`check_lambda` *check lambda*

Description

check lambda

Usage

```
check_lambda(lambda, infb, supb)
```

Arguments

`lambda`	Numeric, value of lambda.
`infb`	Numeric, lower bound of lambda.
`supb`	Numeric, upper bound of lambda.

Value

lambda Numeric, valid value of lambda.

`choice_p` *choice model*

Description

choice model

Usage

```
choice_p(effect)
```

Arguments

`effect`	Factor, simple, fixed or lasso.

Value

penalty Numeric, 1, 2 and 3.

clean_data *Clean missings*

Description

Clean missings

Usage

```
clean_data(y, x, id)
```

Arguments

y	Numeric vector, outcome.
x	Numeric matrix, covariates
id	Numeric vector, identifies the unit to which the observation belongs.

Value

list with the same objects y, x, id, but without missings.

Examples

```
n = 10
m = 4
d = 3
N = n*m
L = N*d
x = matrix(rnorm(L), ncol=d, nrow=N)
subj = rep(1:n, each=m)
alpha = rnorm(n)
beta = rnorm(d)
eps = rnorm(N)
y = x %*% beta  + matrix(rep(alpha, each=m) + eps)
y = as.vector(y)
x[1,3] = NA
clean_data(y=y, x=x, id=subj)
```

d_psi_als *D Psi ALS*

Description

Derivative of Psi asymetric least square

Usage

```
d_psi_als(x, tau)
```

Arguments

x	generic vector
tau	percentile

Value

y vector, linear transformation by derivative ALS psi

d_psi_mq *D Psi M-quantile*

Description

Derivative of psi M-quantile

Usage

```
d_psi_mq(x, tau, c)
```

Arguments

x	generic vector
tau	percentile
c	tuning

Value

y vector, linear transformation by second derivative m-rho

f_den *Kernel density*

Description

Kernel density

Usage

```
f_den(x)
```

Arguments

x	Numeric vector.

Value

y vector, kernel density estimation.

Examples

```
x = rnorm(10)
f_den(x)
```

f_tab *Tabular function*

Description

Tabular function

Usage

```
f_tab(N, n, d, theta, sig2, kind)
```

Arguments

N	sample size.
n	length of alpha.
d	length of beta.
theta	Numeric vector.
sig2	Numeric vector.
kind	Numeric, 1 means alpha, 2 means beta

Value

a list with a dataframe Core and a matrix Matx, both display the same information

---

`loss_er` *Loss expectile regression*

Description

This function returns the core of expectile regression to be minimized

Usage

```
loss_er(beta, x, y, tau, N, d)
```

Arguments

`beta`	initial values
`x`	design matrix
`y`	vector output
`tau`	percentile
`N`	sample size
`d`	columns of x

Value

eta Numeric, sum of expectile regression

`loss_erfe` *Loss expectile regression with fixed effects*

Description

This function returns the core of expectile regression with fixed effects to be minimized

Usage

```
loss_erfe(theta, x, y, z, tau, n, d, mm)
```

Arguments

`theta`	initial values
`x`	design matrix
`y`	vector output
`z`	incident matrix
`tau`	percentile
`n`	N sample size
`d`	columns of x
`mm`	n columns of z

Value

eta Numeric, sum of expectile regression with fixed effects

`loss_erlasso` *Loss lasso expectile regression with fixed effects*

Description

This function returns the core of lasso expectile regression with fixed effects to be minimized

Usage

```
loss_erlasso(theta, x, y, z, tau, n, d, mm, lambda)
```

Arguments

`theta`	initial values
`x`	design matrix
`y`	vector output
`z`	incident matrix
`tau`	percentile
`n`	N sample size
`d`	columns of x
`mm`	n columns of z
`lambda`	constriction parameter

Value

eta Numeric, sum of lasso expectile regression with fixed effects

`loss_mqr` *Loss M-quantile regression*

Description

This function returns the core of M-quantile regression to be minimized

Usage

```
loss_mqr(beta, x, y, tau, N, d, c)
```

Arguments

`beta`	initial values
`x`	design matrix
`y`	vector output
`tau`	percentile
`N`	sample size
`d`	columns of x
`c`	tuning

Value

eta Numeric, sum of M-quantile regression

`loss_mqrfe` *Loss M-quantile regression with fixed effects*

Description

This function returns the core of M-quantile regression with fixed effects to be minimized

Usage

```
loss_mqrfe(theta, x, y, z, tau, n, d, mm, c)
```

Arguments

`theta`	initial values
`x`	design matrix
`y`	vector output
`z`	incident matrix
`tau`	percentile
`n`	N sample size
`d`	columns of x
`mm`	n columns of z
`c`	tuning

Value

eta Numeric, sum of M-quantile regression with fixed effects

`loss_mqrlasso` *Loss lasso M-quantile regression with fixed effects*

Description

This function returns the core of lasso M-quantile regression with fixed effects to be minimized

Usage

```
loss_mqrlasso(theta, x, y, z, tau, n, d, mm, c, lambda)
```

Arguments

`theta`	initial values
`x`	design matrix
`y`	vector output
`z`	incident matrix
`tau`	percentile
`n`	N sample size
`d`	columns of x
`mm`	n columns of z
`c`	tuning
`lambda`	constriction parameter

Value

eta Numeric, sum of lasso M-quantile regression with fixed effects

loss_qr *Loss quantile regression*

Description

This function returns the core of quantile regression to be minimized

Usage

```
loss_qr(beta, x, y, tau, N, d)
```

Arguments

beta	initial values
x	design matrix
y	vector output
tau	percentile
N	sample size
d	columns of x

Value

eta Numeric, sum of quantile regression

loss_qrfe *Loss quantile regression with fixed effects*

Description

This function returns the core of quantile regression with fixed effects to be minimized

Usage

```
loss_qrfe(theta, x, y, z, tau, n, d, mm)
```

Arguments

theta	initial values
x	design matrix
y	vector output
z	incident matrix
tau	percentile
n	N sample size
d	columns of x
mm	n columns of z

Value

eta Numeric, sum of quantile regression with fixed effects

`loss_qrlasso` *Loss lasso quantile regression with fixed effects*

Description

This function returns the core of lasso quantile regression with fixed effects to be minimized

Usage

```
loss_qrlasso(theta, x, y, z, tau, n, d, mm, lambda)
```

Arguments

`theta`	initial values
`x`	design matrix
`y`	vector output
`z`	incident matrix
`tau`	percentile
`n`	N sample size
`d`	columns of x
`mm`	n columns of z
`lambda`	constriction parameter

Value

eta Numeric, sum of lasso quantile regression with fixed effects

`mpqr` *Multiple penalized quantile regression*

Description

Estimate penalized quantile regression for several taus

Usage

```
mpqr(x, y, subj, tau = 1:9/10, effect = "simple", c = 0)
```

Arguments

`x` Numeric matrix, covariates

`y` Numeric vector, outcome.

`subj` Numeric vector, identifies the unit to which the observation belongs.

`tau` Numeric vector, identifies the percentiles.

`effect` Factor, "simple" simple regression, "fixed" regression with fixed effects, "lasso" penalized regression with fixed effects.

`c` Numeric, 0 is quantile, Inf is expectile, any number between zero and infinite is M-quantile.

Value

Beta Numeric array, with three dimmensions: 1) tau, 2) coef., lower bound, upper bound, 3) exploratory variables.

Beta array with dimension (ntau, 3, d), where Beta[i,1,k] is the i-th tau estimation of beta_k, Beta[i,2,k] is the i-th tau lower bound 95% confidence of beta_k, and Beta[i,3,k] is the i-th tau lower bound 95% confidence of beta_k.

Examples

```
n = 10
m = 5
d = 4
N = n*m
L = N*d
x = matrix(rnorm(L), ncol=d, nrow=N)
subj = rep(1:n, each=m)
alpha = rnorm(n)
beta = rnorm(d)
eps = rnorm(N)
y = as.vector(x %*% beta + rep(alpha, each=m) + eps)

Beta = mpqr(x,y,subj,tau=1:9/10, effect="fixed", c = 1.2)
Beta
```

`optim_er` *optim expectile regression*

Description

This function solves a expectile regression

Usage

```
optim_er(beta, x, y, tau, N, d)
```

Arguments

`beta`	Numeric vector, initials values beta.
`x`	Numeric matrix, covariates.
`y`	Numeric vector, output.
`tau`	Numeric scalar, the percentile.
`N`	Numeric integer, sample size.
`d`	Numeric integer, X number of columns.

Value

parametric vector and residuals.

`optim_erfe` *optim expectile regression with fixed effects*

Description

This function solves a expectile regression with fixed effects

Usage

```
optim_erfe(beta, alpha, x, y, z, tau, N, d, n)
```

Arguments

`beta`	Numeric vector, initials values beta.
`alpha`	Numeric vector, initials values alpha.
`x`	Numeric matrix, covariates.
`y`	Numeric vector, output.
`z`	Numeric matrix, incidence matrix.
`tau`	Numeric scalar, the percentile.
`N`	Numeric integer, sample size.
`d`	Numeric integer, X number of columns.
`n`	Numeric integer, length of alpha.

Value

parametric vector and residuals.

`optim_erlasso` *optim expectile regression with fixed effects and LASSO*

Description

This function solves a expectile regression with fixed effects and LASSO

Usage

```
optim_erlasso(beta, alpha, x, y, z, tau, N, d, n)
```

Arguments

`beta`	Numeric vector, initials values beta.
`alpha`	Numeric vector, initials values alpha.
`x`	Numeric matrix, covariates.
`y`	Numeric vector, output.
`z`	Numeric matrix, incidence matrix.
`tau`	Numeric scalar, the percentile.
`N`	Numeric integer, sample size.
`d`	Numeric integer, X number of columns.
`n`	Numeric integer, length of alpha.

Value

parametric vector and residuals.

`optim_mqr` *optim M-quantile regression*

Description

This function solves a M-quantile regression

Usage

```
optim_mqr(beta, x, y, tau, N, d, c)
```

Arguments

beta	Numeric vector, initials values beta.
x	Numeric matrix, covariates.
y	Numeric vector, output.
tau	Numeric scalar, the percentile.
N	Numeric integer, sample size.
d	Numeric integer, X number of columns.
c	Numeric, positive real value.

Value

parametric vector and residuals.

optim_mqrfe *optim quantile regression with fixed effects*

Description

This function solves a quantile regression with fixed effects

Usage

```
optim_mqrfe(beta, alpha, x, y, z, tau, N, d, n, c)
```

Arguments

beta	Numeric vector, initials values beta.
alpha	Numeric vector, initials values alpha.
x	Numeric matrix, covariates.
y	Numeric vector, output.
z	Numeric matrix, incidence matrix.
tau	Numeric scalar, the percentile.
N	Numeric integer, sample size.
d	Numeric integer, X number of columns.
n	Numeric integer, length of alpha.
c	Numeric, positive real value.

Value

parametric vector and residuals.

optim_mqrlasso *optim M-quantile regression with fixed effects and LASSO*

Description

This function solves a M-quantile regression with fixed effects and LASSO

Usage

```
optim_mqrlasso(beta, alpha, x, y, z, tau, N, d, n, c)
```

Arguments

beta	Numeric vector, initials values beta.
alpha	Numeric vector, initials values alpha.
x	Numeric matrix, covariates.
y	Numeric vector, output.
z	Numeric matrix, incidence matrix.
tau	Numeric scalar, the percentile.
N	Numeric integer, sample size.
d	Numeric integer, X number of columns.
n	Numeric integer, length of alpha.
c	Numeric, positive real value.

Value

parametric vector and residuals.

optim_qr *optim quantile regression*

Description

This function solves a quantile regression

Usage

```
optim_qr(beta, x, y, tau, N, d)
```

Arguments

`beta`	Numeric vector, initials values.
`x`	Numeric matrix, covariates.
`y`	Numeric vector, output.
`tau`	Numeric scalar, the percentile.
`N`	Numeric integer, sample size.
`d`	Numeric integer, X number of columns.

Value

parametric vector and residuals.

`optim_qrfe` *optim quantile regression with fixed effects*

Description

This function solves a quantile regression with fixed effects

Usage

```
optim_qrfe(beta, alpha, x, y, z, tau, N, d, n)
```

Arguments

`beta`	Numeric vector, initials values beta.
`alpha`	Numeric vector, initials values alpha.
`x`	Numeric matrix, covariates.
`y`	Numeric vector, output.
`z`	Numeric matrix, incidence matrix.
`tau`	Numeric scalar, the percentile.
`N`	Numeric integer, sample size.
`d`	Numeric integer, X number of columns.
`n`	Numeric integer, length of alpha.

Value

parametric vector and residuals.

optim_qrlasso *optim quantile regression with fixed effects and LASSO*

Description

This function solves a quantile regression with fixed effects and LASSO

Usage

```
optim_qrlasso(beta, alpha, x, y, z, tau, N, d, n)
```

Arguments

beta	Numeric vector, initials values beta.
alpha	Numeric vector, initials values alpha.
x	Numeric matrix, covariates.
y	Numeric vector, output.
z	Numeric matrix, incidence matrix.
tau	Numeric scalar, the percentile.
N	Numeric integer, sample size.
d	Numeric integer, X number of columns.
n	Numeric integer, length of alpha.

Value

parametric vector and residuals.

plot_taus *Plot multiple penalized quantile regression*

Description

plot penalized quantile regression for several taus

Usage

```
plot_taus(
  Beta,
  tau = 1:9/10,
  D,
  col = 2,
  lwd = 1,
  lty = 2,
  pch = 16,
  cex.axis = 1,
  cex.lab = 1,
  main = "",
  shadow = "gray90"
)
```

Arguments

Beta	Numeric array, with three dimmensions: 1) tau, 2) coef., lower bound, upper bound, 3) exploratory variables.
tau	Numeric vector, identifies the percentiles.
D	covariate's number.
col	color.
lwd	line width.
lty	line type.
pch	point character.
cex.axis	cex axis length.
cex.lab	cex axis length.
main	title.
shadow	color of the Confidence Interval 95%

Value

None

Examples

```
n = 10
m = 5
d = 4
N = n*m
L = N*d
x = matrix(rnorm(L), ncol=d, nrow=N)
subj = rep(1:n, each=m)
alpha = rnorm(n)
beta = rnorm(d)
eps = rnorm(N)
y = as.vector(x %*% beta + rep(alpha, each=m) + eps)

Beta = mpqr(x,y,subj,tau=1:9/10, effect="lasso", c = Inf)
plot_taus(Beta,tau=1:9/10,D=1)
```

pqr *Penalized quantile regression with fixed effects*

Description

Estimate parameters and tuning parameter.

Usage

```
pqr(x, y, subj, tau = 0.5, effect = "simple", c = 1)
```

Arguments

`x` Numeric matrix, covariates

`y` Numeric vector, outcome.

`subj` Numeric vector, identifies the unit to which the observation belongs.

`tau` Numeric scalar between zero and one, identifies the percentile.

`effect` Factor, "simple" simple regression, "fixed" regression with fixed effects, "lasso" penalized regression with fixed effects.

`c` Numeric, 0 is quantile, Inf is expectile, any number between zero and infinite is M-quantile.

Value

alpha Numeric vector, intercepts' coefficients.

beta Numeric vector, exploratory variables' coefficients.

lambda Numeric, estimated lambda.

res Numeric vector, percentile residuals.

tau Numeric scalar, the percentile.

penalty Numeric scalar, indicate the chosen effect.

c Numeric scalar, indicate the chosen c.

sig2_alpha Numeric vector, intercepts' standard errors.

sig2_beta Numeric vector, exploratory variables' standard errors.

Tab_alpha Data.frame, intercepts' summary.

Tab_beta Data.frame, exploratory variables' summary.

Mat_alpha Numeric matrix, intercepts' summary.

Mat_beta Numeric matrix, exploratory variables' summary.

References

Koenker, R. (2004) "Quantile regression for longitudinal data", J. Multivar. Anal., 91(1): 74-89, <doi:10.1016/j.jmva.2004.05.006>

Examples

```
n = 10
m = 5
d = 4
N = n*m
x = matrix(rnorm(d*N), ncol=d, nrow=N)
subj = rep(1:n, each=m)
alpha = rnorm(n)
beta = rnorm(d)
eps = rnorm(N)
y = as.vector(x %*% beta + rep(alpha, each=m) + eps)
m1 = pqr(x=x, y=y, subj=subj, tau=0.75, effect="lasso", c = 0)
m1$Tab_beta
```

`print.PQR` *Print an PQR*

Description

Define the visible part of the object class PQR

Usage

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

Arguments

`x`	An object of class "PQR"
`...`	further arguments passed to or from other methods.

Value

None

`psi_als` *Psi ALS*

Description

Psi asymetric least square

Usage

```
psi_als(x, tau)
```

Arguments

x	generic vector
tau	percentile

Value

y vector, linear transformation by ALS psi

psi_mq *Psi M-quantile*

Description

Psi M-quantile

Usage

```
psi_mq(x, tau, c)
```

Arguments

x	generic vector
tau	percentile
c	tuning

Value

y vector, linear transformation by m-rho derivative

q_cov *Covariance*

Description

Estimate Covariance matrix

Usage

```
q_cov(n, N, d, Z, X, tau, res, penalty, c)
```

Arguments

`n`	length of alpha.
`N`	sample size.
`d`	length of beta.
`Z`	Numeric matrix, incident matrix.
`X`	Numeric matrix, covariates.
`tau`	Numeric, identifies the percentile.
`res`	Numeric vector, residuals.
`penalty`	Numeric, 1 quantile regression, 2 quantile regression with fixed effects, 3 Lasso quantile regression with fixed effects
`c`	Numeric, tuning

Value

a list with two matrices: sig2_alpha (which is the matrix of covariance of estimated alpha) and sig2_beta (which is the matrix of covariance of estimated beta)

`rho_koenker` *Rho Koenker*

Description

Rho Koenker

Usage

```
rho_koenker(x, tau)
```

Arguments

`x`	generic vector
`tau`	percentile

Value

y vector, linear transformation by rho

`rho_mq` *Rho M-quantile*

Description

Rho M-quantile

Usage

```
rho_mq(x, tau, c)
```

Arguments

`x`	generic vector
`tau`	percentile
`c`	tuning

Value

y vector, linear transformation by m-rho

`sgf` *Identify significance*

Description

Identify significance

Usage

```
sgf(x)
```

Arguments

`x`	Numeric vector.

Value

y vector Factor, symbol flag of significant p-values.

a vector of Factors, i.e., the symbols to help p-value interpretation

Examples

```
n = 10
pvalue = rgamma(10,1,10)
sgf(pvalue)
```

Index