Package 'pqrfe'

May 9, 2026

Type Package

Title Penalized Quantile Regression with Fixed Effects

Version 1.3

Date 2025-12-08

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

Encoding UTF-8

Suggests testthat (>= 3.0.0), tinytest (>= 1.3.1)

RoxygenNote 7.3.3

NeedsCompilation yes

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

Maintainer Ian Meneghel Danilevicz <iandanilevicz@gmail.com>

Repository CRAN

Config/testthat/edition 3

Date/Publication 2025-12-08 13:50:02 UTC

Contents

`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)
```

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
```

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 regression parameters and tuning parameters for quantile, expectile, or M-quantile regression.

Remarks:

1. If the first column of ‘x‘ is entirely equal to 1, then the first element of ‘beta‘ represents the common intercept. Otherwise, there is no default common intercept (unlike the default behavior in ‘lm‘).

2. If there is a common intercept and ‘effect‘ is ‘"fixed"‘ or ‘"lasso"‘, a ’sum-to-zero constraint’ is applied on the ‘alpha‘ parameters:

$$\sum_{i=1}^{n} \alpha_i = 0$$

This follows the approach in Danilevicz (2025).

Usage

```
pqr(x, y, subj, tau = 0.5, effect = "fixed", c = 0)
```

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

Danilevicz, I.M., Bondon, P., Reisen, V.A. (2025) "Adaptive LASSO Quantile Regression with Fixed Effects", Appl. Math. Model., xx (xx), <doi:10.1016/j.apm.2025.116600>

Danilevicz, I.M., Reisen, V.A., Bondon, P. (2024) "Expectile and M-quantile regression for panel data", Stat. Comput., 34 (97), <doi:10.1007/s11222-024-10396-7>

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
```

Index