Package ‘alqrfe’

May 7, 2026

Title Adaptive Lasso Quantile Regression with Fixed Effects

Version 1.3

Description
Quantile regression with fixed effects solves longitudinal data, considering the individual intercepts as fixed effects. The parametric set of this type of problem used to be huge. Thus penalized methods such as Lasso are currently applied. Adaptive Lasso presents oracle proprieties, which include Gaussianity and correct model selection. Bayesian information criteria (BIC) estimates the optimal tuning parameter lambda. Plot tools are also available.

License GPL (>= 2)

Depends R (>= 4.4.0)

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

LinkingTo Rcpp, RcppArmadillo

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation yes

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

Maintainer Ian Meneghel Danilevicz <iandanilevicz@gmail.com>

Repository CRAN

Date/Publication 2025-12-08 15:20:02 UTC

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

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)
```

mqr *multiple penalized quantile regression*

Description

Estimate QR for several taus

Usage

```
mqr(x, y, subj, tau = 1:9/10, method = "qr", ngrid = 20, inf = 1e-08, digt = 4)
```

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.
method	Factor, "qr" quantile regression, "qrfe" quantile regression with fixed effects, "lqrfe" Lasso quantile regression with fixed effects, "alqr" adaptive Lasso quantile regression with fixed effects.
ngrid	Numeric scalar greater than one, number of BIC to test.
inf	Numeric scalar, internal value, small value.
digt	Numeric scalar, internal value greater than one, define "zero" coefficient.

Value

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

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 = x %*% beta  + matrix(rep(alpha, each=m) + eps)
y = as.vector(y)

Beta = mqr(x,y,subj,tau=1:9/10, method="qr", ngrid = 10)
Beta
```

mqr_alpha *multiple penalized quantile regression - alpha*

Description

Estimate QR intercepts for several taus

Usage

```
mqr_alpha(
  x,
  y,
  subj,
  tau = 1:9/10,
  method = "qr",
  ngrid = 20,
  inf = 1e-08,
  digt = 4
)
```

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.
method	Factor, "qr" quantile regression, "qrfe" quantile regression with fixed effects, "lqrfe" Lasso quantile regression with fixed effects, "alqr" adaptive Lasso quantile regression with fixed effects.
ngrid	Numeric scalar greater than one, number of BIC to test.
inf	Numeric scalar, internal value, small value.
digt	Numeric scalar, internal value greater than one, define "zero" coefficient.

Value

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

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 = x %*% beta  + matrix(rep(alpha, each=m) + eps)
y = as.vector(y)

Alpha = mqr(x,y,subj,tau=1:9/10, method="qr", ngrid = 10)
Alpha
```

`plot_alpha` *plot multiple penalized quantile regression - alpha*

Description

plot QR intercepts for several taus

Usage

```
plot_alpha(
  Beta,
  tau = 1:9/10,
  D,
  ylab = expression(alpha[1]),
  col = 2,
  lwd = 1,
  lty = 2,
  pch = 1,
  cex.axis = 1,
  cex.lab = 1,
  main = ""
)
```

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`	intercept's number.
`ylab`	y legend
`col`	color.
`lwd`	line width.
`lty`	line type.
`pch`	point character.
`cex.axis`	cex axis length.
`cex.lab`	cex axis length.
`main`	title.

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 = x %*% beta  + matrix(rep(alpha, each=m) + eps)
y = as.vector(y)

Beta = mqr_alpha(x,y,subj,tau=1:9/10, method="qr", ngrid = 10)
plot_alpha(Beta,tau=1:9/10,D=1)
```

---

plot_taus *plot multiple penalized quantile regression*

Description

plot QR for several taus

Usage

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

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.

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 = x %*% beta  + matrix(rep(alpha, each=m) + eps)
y = as.vector(y)

Beta = mqr(x,y,subj,tau=1:9/10, method="qr", ngrid = 10)
plot_taus(Beta,tau=1:9/10,D=1)
```

`qr` *quantile regression*

Description

Estimate quantile regression parameters for any quantile.

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 ‘method‘ is ‘"qrfe"‘, ‘"lqrfe"‘ or ‘"alqrfe"‘, 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

```
qr(x, y, subj, tau = 0.5, method = "qrfe", ngrid = 20, inf = 1e-08, digt = 4)
```

Arguments

`x`	Numeric matrix, covariates
`y`	Numeric vector, outcome.
`subj`	Numeric vector, identifies the unit to which the observation belongs.
`tau`	Numeric, identifies the percentile.
`method`	Factor, "qr" quantile regression, "qrfe" quantile regression with fixed effects, "lqrfe" Lasso quantile regression with fixed effects, "alqrfe" adaptive Lasso quantile regression with fixed effects.
`ngrid`	Numeric scalar greater than one, number of BIC to test.
`inf`	Numeric scalar, internal value, small value.
`digt`	Numeric scalar, internal value greater than one, define "zero" coefficient.

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.

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.

method Factor, method applied.

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

```
# Example 1
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 = x %*% beta  + matrix(rep(alpha, each=m) + eps)
y = as.vector(y)
m1 = qr(x,y,subj,tau=0.75, method="qrfe")
m1
m2 = qr(x,y,subj,tau=0.3, method="lqrfe", ngrid = 10)
m2

# Example 2, from MASS package
Rabbit = MASS::Rabbit
Rabbit$Treatment = ifelse(Rabbit$Treatment=="Control",0,1)
Rabbit$Animal = ifelse(Rabbit$Animal == "R1",1,ifelse(Rabbit$Animal == "R2",2,
ifelse(Rabbit$Animal == "R3",3,ifelse(Rabbit$Animal == "R4",4,5))))
X = matrix(cbind(Rabbit$Dose,Rabbit$Treatment), ncol=2)
m3 = qr(x=X, y=Rabbit$BPchange, subj=Rabbit$Animal,tau=0.5, method="alqrfe", ngrid = 10)
m3
```

Index