Package 'cqrReg'

October 12, 2022

Title Quantile, Composite Quantile Regression and Regularized Versions

Type Package

Version 1.2.1	
Date 2022-06-07	
Author Jueyu Gao & Linglong Kong	
Maintainer Jueyu Gao <jueyu@ualberta.ca></jueyu@ualberta.ca>	
Description Estimate quantile regression(QR) and composite quantile regression (cqr) and with adaptive lasso penalty using interior point (IP), majorize and minimize(MM), coordinate descent (CD), and alternating direction method of multipliers algorithms(ADMM).	•
License GPL (>= 2)	
Depends Rcpp (>= 0.10.0),quantreg,R (>= 2.6)	
LinkingTo Rcpp,RcppArmadillo	
NeedsCompilation yes	
Repository CRAN	
Date/Publication 2022-06-07 05:00:02 UTC	
R topics documented:	
cqr.admm	2
T	3
1	5
1	6
-1r	7
1	8
4	9
cqr.lasso.mm	_
cqr.mm	_
CQRADMMCPP	
CQRCDCPP	_
CQRMMCPP)

 2 cqr.admm

cqr.	admm	Com, Meth		_		_		,	-	-		ie	A	lte	rn	ati	ing	3	D_i	ire	cti	on
Index																						27
	QRPMMCPP		• •	 	 •	 •	 •		•		•	•			•	•	•		•	٠	•	. 26
	QRPCDCPP																					
	QRPADMMCPP .			 																		. 26
	QRMMCPP																					
	qrfit.lasso																					
	qrfit																					
	QRCDCPP																					
	QRADMMCPP																					
	QR.mm																					
	QR.lasso.ip QR.lasso.mm																					
	QR.lasso.cd																					
	QR.lasso.admm																					
	QR.ip																					
	QR.cd																					
	QR.admm			 																		. 14
	CQRPMMCPP			 																		. 14

Description

Composite quantile regression (cqr) find the estimated coefficient which minimize the absolute error for various quantile level. The problem is well suited to distributed convex optimization and is based on Alternating Direction Method of Multipliers (ADMM) algorithm .

Usage

```
cqr.admm(X,y,tau,rho,beta, maxit, toler)
```

Arguments

Χ	the design matrix
У	response variable
tau	vector of quantile level
rho	augmented Lagrangian parameter
beta	initial value of estimate coefficient (default naive guess by least square estimation)
maxit	maxim iteration (default 200)
toler	the tolerance critical for stop the algorithm (default 1e-3)

cqr.cd 3

Value

a list structure is with components

beta the vector of estimated coefficient

b intercept

Note

cqr.admm(x,y,tau) work properly only if the least square estimation is good.

References

S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein. (2010) Distributed Optimization and Statistical Learning via the Alternating Direction. Method of Multipliers *Foundations and Trends in Machine Learning*, **3**, No. 1, 1–122

Hui Zou and Ming Yuan(2008). Composite Quantile Regression and the Oracle Model Selection Theory, *The Annals of Statistics*, **36**, Number 3, Page 1108–1126.

Examples

```
set.seed(1)
n=100
p=2
a=rnorm(n*p, mean = 1, sd =1)
x=matrix(a,n,p)
beta=rnorm(p,1,1)
beta=matrix(beta,p,1)
y=x%*%beta-matrix(rnorm(n,0.1,1),n,1)
tau=1:5/6
# x is 1000*10 matrix, y is 1000*1 vector, beta is 10*1 vector
cqr.admm(x,y,tau)
```

cqr.cd

Composite Quantile Regression (cqr) use Coordinate Descent (cd) Algorithms

Description

Composite quantile regression (cqr) find the estimated coefficient which minimize the absolute error for various quantile level. The algorithm base on greedy coordinate descent and Edgeworth's for ordinary l_1 regression.

Usage

```
cqr.cd(X,y,tau,beta,maxit,toler)
```

4 cqr.cd

Arguments

X the design matrix
y response variable
tau vector of quantile level
beta initial value of estimate coefficient (default naive guess by least square estimation)
maxit maxim iteration (default 200)
toler the tolerance critical for stop the algorithm (default 1e-3)

Value

a list structure is with components

beta the vector of estimated coefficient

b intercept

Note

cqr.cd(x,y,tau) work properly only if the least square estimation is good.

References

Wu, T.T. and Lange, K. (2008). Coordinate Descent Algorithms for Lasso Penalized Regression. *Annals of Applied Statistics*, **2**, No 1, 224–244.

Hui Zou and Ming Yuan(2008). Composite Quantile Regression and the Oracle Model Selection Theory, *The Annals of Statistics*, **36**, Number 3, Page 1108–1126.

```
set.seed(1)
n=100
p=2
a=rnorm(n*p, mean = 1, sd =1)
x=matrix(a,n,p)
beta=rnorm(p,1,1)
beta=matrix(beta,p,1)
y=x%*%beta-matrix(rnorm(n,0.1,1),n,1)
tau=1:5/6
# x is 1000*10 matrix, y is 1000*1 vector, beta is 10*1 vector
cqr.cd(x,y,tau)
```

cqr.fit 5

Composite Quantile Regression (cqr) model fitting
Composite Quantile Regression (cqr) model fitting

Description

Composite quantile regression (cqr) find the estimated coefficient which minimize the absolute error for various quantile level. High level function for estimating parameter by composite quantile regression.

Usage

```
cqr.fit(X,y,tau,beta,method,maxit,toler,rho)
```

Arguments

X	the design matrix
у	response variable
tau	vector of quantile level
method	"mm" for majorize and minimize method, "cd" for coordinate descent method, "admm" for Alternating method of mulipliers method, "ip" for interior point mehod
rho	augmented Lagrangian parameter
beta	initial value of estimate coefficient (default naive guess by least square estimation)
maxit	maxim iteration (default 200)
toler	the tolerance critical for stop the algorithm (default 1e-3)

Value

a list structure is with components

beta the vector of estimated coefficient

b intercept

Note

cqr.fit(x,y,tau) work properly only if the least square estimation is good. Interior point method is done by quantreg.

6 cqr.fit.lasso

cqr.fit.lasso	Composite (lasso)	Quantile	Regression	(cqr)	with	Adaptive	Lasso	Penalty	

Description

Composite quantile regression (cqr) find the estimated coefficient which minimize the absolute error for various quantile level. High level function for estimating and selecting parameter by composite quantile regression with adaptive lasso penalty.

Usage

```
cqr.fit.lasso(X,y,tau,lambda,beta,method,maxit,toler,rho)
```

Arguments

Χ	the design matrix
У	response variable
tau	vector of quantile level
method	"mm" for majorize and minimize method, "cd" for coordinate descent method, "admm" for Alternating method of mulipliers method
lambda	The constant coefficient of penalty function. (default lambda=1)
rho	augmented Lagrangian parameter
beta	initial value of estimate coefficient (default naive guess by least square estimation)
maxit	maxim iteration (default 200)
toler	the tolerance critical for stop the algorithm (default 1e-3)

Value

a list structure is with components

beta the vector of estimated coefficient

b intercept

Note

cqr.fit.lasso(x,y,tau) work properly only if the least square estimation is good.

cqr.ip 7

Composite Quantile Regression (cqr) use Interior Point (ip) Method

Description

The function use the interior point method from quantreg to solve the quantile regression problem.

Usage

```
cqr.ip(X,y,tau)
```

Arguments

X	the design matrix
у	response variable
tau	vector of quantile level

Value

a list structure is with components

beta the vector of estimated coefficient

b intercept

Note

Need to install quantreg package from CRAN.

References

Koenker, R. and S. Portnoy (1997). The Gaussian Hare and the Laplacian Tortoise: Computability of squared-error vs. absolute-error estimators, with discussion, *Statistical Science*, **12**, 279-300.

Hui Zou and Ming Yuan(2008). Composite Quantile Regression and the Oracle Model Selection Theory, *The Annals of Statistics*, **36**, Number 3, Page 1108–1126.

```
set.seed(1)
n=100
p=2
a=rnorm(n*p, mean = 1, sd =1)
x=matrix(a,n,p)
beta=rnorm(p,1,1)
beta=matrix(beta,p,1)
y=x%*%beta-matrix(rnorm(n,0.1,1),n,1)
tau=1:5/6
# x is 1000*10 matrix, y is 1000*1 vector, beta is 10*1 vector
#you should install quantreg first to run following command
#cqr.ip(x,y,tau)
```

8 cqr.lasso.admm

cqr.lasso.admm	Composite Quantile Regression (cqr) with Adaptive Lasso Penalty (lasso) use Alternating Direction Method of Multipliers (ADMM) algorithm

Description

The adaptive lasso parameter base on the estimated coefficient without penalty function. Composite quantile regression find the estimated coefficient which minimize the absolute error for various quantile level. The problem is well suited to distributed convex optimization and is based on Alternating Direction Method of Multipliers (ADMM) algorithm.

Usage

```
cqr.lasso.admm(X,y,tau,lambda,rho,beta,maxit)
```

Arguments

X the design matrix
y response variable
tau vector of quantile level

lambda The constant coefficient of penalty function. (default lambda=1)

rho augmented Lagrangian parameter

beta initial value of estimate coefficient (default naive guess by least square estima-

tion)

maxit maxim iteration (default 200)

Value

a list structure is with components

beta the vector of estimated coefficient

b intercept

Note

cqr.lasso.admm(x,y,tau) work properly only if the least square estimation is good.

References

S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein.(2010) Distributed Optimization and Statistical Learning via the Alternating Direction. Method of Multipliers *Foundations and Trends in Machine Learning*, **3**, No. 1, 1–122

Hui Zou and Ming Yuan(2008). Composite Quantile Regression and the Oracle Model Selection Theory, *The Annals of Statistics*, **36**, Number 3, Page 1108–1126.

cqr.lasso.cd 9

Examples

```
set.seed(1)
n=100
p=2
a=2*rnorm(n*2*p, mean = 1, sd =1)
x=matrix(a,n,2*p)
beta=2*rnorm(p,1,1)
beta=rbind(matrix(beta,p,1),matrix(0,p,1))
y=x%*%beta-matrix(rnorm(n,0.1,1),n,1)
tau=1:5/6
# x is 1000*20 matrix, y is 1000*1 vector, beta is 20*1 vector with last ten zero value elements.
cqr.lasso.admm(x,y,tau)
```

cgr.lasso.cd

Composite Quantile Regression (cqr) with Adaptive Lasso Penalty (lasso) use Coordinate Descent (cd) Algorithms

Description

The adaptive lasso parameter base on the estimated coefficient without penalty function. Composite quantile regression find the estimated coefficient which minimize the absolute error for various quantile level. The algorithm base on greedy coordinate descent and Edgeworth's for ordinary l_1 regression.

Usage

```
cqr.lasso.cd(X,y,tau,lambda,beta,maxit,toler)
```

Arguments

X the design matrix
y response variable
tau vector of quantile level

lambda The constant coefficient of penalty function. (default lambda=1)

beta initial value of estimate coefficient (default naive guess by least square estima-

tion)

maxit maxim iteration (default 200)

toler the tolerance critical for stop the algorithm (default 1e-3)

Value

a list structure is with components

beta the vector of estimated coefficient

b intercept

10 cqr.lasso.mm

Note

cqr.lasso.cd(x,y,tau) work properly only if the least square estimation is good.

References

Wu, T.T. and Lange, K. (2008). Coordinate Descent Algorithms for Lasso Penalized Regression. *Annals of Applied Statistics*, **2**, No 1, 224–244.

Hui Zou and Ming Yuan(2008). Composite Quantile Regression and the Oracle Model Selection Theory, *The Annals of Statistics*, **36**, Number 3, Page 1108–1126.

Examples

```
set.seed(1)
n=100
p=2
a=2*rnorm(n*2*p, mean = 1, sd =1)
x=matrix(a,n,2*p)
beta=2*rnorm(p,1,1)
beta=rbind(matrix(beta,p,1),matrix(0,p,1))
y=x%**%beta-matrix(rnorm(n,0.1,1),n,1)
tau=1:5/6
# x is 1000*20 matrix, y is 1000*1 vector, beta is 20*1 vector with last ten zero value elements.
cqr.lasso.cd(x,y,tau)
```

cqr.lasso.mm

Composite Quantile Regression (cqr) with Adaptive Lasso Penalty (lasso) use Majorize and Minimize (mm) Algorithm

Description

The adaptive lasso penalty parameter base on the estimated coefficient without penalty function. Composite quantile regression find the estimated coefficient which minimize the absolute error for various quantile level. The algorithm majorizing the objective function by a quadratic function followed by minimizing that quadratic.

Usage

```
cqr.lasso.mm(X,y,tau,lambda,beta,maxit,toler)
```

Arguments

Χ	the design matrix
У	response variable
tau	vector of quantile level

lambda The constant coefficient of penalty function. (default lambda=1)

beta initial value of estimate coefficient (default naive guess by least square estima-

tion)

cqr.mm

maxit maxim iteration (default 200)

toler the tolerance critical for stop the algorithm (default 1e-3)

Value

a list structure is with components

beta the vector of estimated coefficient
b intercept for various quantile level

Note

cqr.lasso.mm(x,y,tau) work properly only if the least square estimation is good.

References

David R.Hunter and Runze Li.(2005) Variable Selection Using MM Algorithms, *The Annals of Statistics* **33**, Number 4, Page 1617–1642.

Hui Zou and Ming Yuan(2008). Composite Quantile Regression and the Oracle Model Selection Theory, *The Annals of Statistics*, **36**, Number 3, Page 1108–1126.

Examples

```
set.seed(1)
n=100
p=2
a=2*rnorm(n*2*p, mean = 1, sd =1)
x=matrix(a,n,2*p)
beta=2*rnorm(p,1,1)
beta=rbind(matrix(beta,p,1),matrix(0,p,1))
y=x%*%beta-matrix(rnorm(n,0.1,1),n,1)
tau=1:5/6
# x is 1000*20 matrix, y is 1000*1 vector, beta is 20*1 vector with last ten zero value elements.
cqr.lasso.mm(x,y,tau)
```

cqr.mm

Composite Quantile Regression (cqr) use Majorize and Minimize (mm) Algorithm

Description

Composite quantile regression find the estimated coefficient which minimize the absolute error for various quantile level. The algorithm majorizing the objective function by a quadratic function followed by minimizing that quadratic.

Usage

```
cqr.mm(X,y,tau,beta,maxit,toler)
```

12 cqr.mm

Arguments

Χ	the design matrix
у	response variable
tau	vector of quantile level
beta	initial value of estimate coefficient (default naive guess by least square estimation)
maxit	maxim iteration (default 200)
toler	the tolerance critical for stop the algorithm (default 1e-3)

Value

a list structure is with components

beta the vector of estimated coefficient
b intercept for various quantile level

Note

cqr.mm(x,y,tau) work properly only if the least square estimation is good.

References

David R.Hunter and Kenneth Lange. Quantile Regression via an MM Algorithm, *Journal of Computational and Graphical Statistics*, **9**, Number 1, Page 60–77.

Hui Zou and Ming Yuan(2008). Composite Quantile Regression and the Oracle Model Selection Theory, *The Annals of Statistics*, **36**, Number 3, Page 1108–1126.

```
set.seed(1)
n=100
p=2
a=rnorm(n*p, mean = 1, sd =1)
x=matrix(a,n,p)
beta=rnorm(p,1,1)
beta=matrix(beta,p,1)
y=x%*%beta-matrix(rnorm(n,0.1,1),n,1)
tau=1:5/6
# x is 1000*10 matrix, y is 1000*1 vector, beta is 10*1 vector
cqr.mm(x,y,tau)
```

CQRADMMCPP 13

CQRADMMCPP	Composite Quantile regression (cqr) use Alternating Direction
	Method of Multipliers (ADMM) algorithm core computational part

Description

Composite quantile regression (cqr) find the estimated coefficient which minimize the absolute error for various quantile level. The problem is well suited to distributed convex optimization and is based on Alternating Direction Method of Multipliers (ADMM) algorithm .

CQRCDCPP	Composite Quantile Regression (cqr) use Coordinate Descent (cd) Algorithms core computational part
	0· ··· ··· · · · · · · · · · · · · · ·

Description

Composite quantile regression (cqr) find the estimated coefficient which minimize the absolute error for various quantile level. The algorithm base on greedy coordinate descent and Edgeworth's for ordinary l_1 regression.

CQRMMCPP	Composite Quantile Regression (cqr) use Majorize and Minimize
	(mm) Algorithm core computational part

Description

Composite quantile regression find the estimated coefficient which minimize the absolute error for various quantile level. The algorithm majorizing the objective function by a quadratic function followed by minimizing that quadratic.

CQRPADMMCPP	Composite Quantile Regression (cqr) with Adaptive Lasso Penalty (lasso) use Alternating Direction Method of Multipliers (ADMM) algorithm core computational part

Description

The adaptive lasso parameter base on the estimated coefficient without penalty function. Composite quantile regression find the estimated coefficient which minimize the absolute error for various quantile level. The problem is well suited to distributed convex optimization and is based on Alternating Direction Method of Multipliers (ADMM) algorithm .

14 QR.admm

CQRPCDCPP	Composite Quantile Regression (cqr) with Adaptive Lasso Penalty (lasso) use Coordinate Descent (cd) Algorithms core computational part

Description

The adaptive lasso parameter base on the estimated coefficient without penalty function. Composite quantile regression find the estimated coefficient which minimize the absolute error for various quantile level. The algorithm base on greedy coordinate descent and Edgeworth's for ordinary l_1 regression.

CQRPMMCPP	Composite Quantile Regression (cqr) with Adaptive Lasso Penalty (lasso) use Majorize and Minimize (mm) Algorithm core computational part
-----------	--

Description

The adaptive lasso penalty parameter base on the estimated coefficient without penalty function. Composite quantile regression find the estimated coefficient which minimize the absolute error for various quantile level. The algorithm majorizing the objective function by a quadratic function followed by minimizing that quadratic.

QR.admm	Quantile Regression (QR) use Alternating Direction Method of Multipliers (ADMM) algorithm
QR . adılıllı	~ ,~ ,

Description

The problem is well suited to distributed convex optimization and is based on Alternating Direction Method of Multipliers (ADMM) algorithm .

Usage

```
QR.admm(X,y,tau,rho,beta, maxit, toler)
```

QR.admm 15

Arguments

Χ	the design matrix
У	response variable
tau	quantile level
rho	augmented Lagrangian parameter
beta	initial value of estimate coefficient (default naive guess by least square estimation)
maxit	maxim iteration (default 200)

the tolerance critical for stop the algorithm (default 1e-3)

Value

toler

a list structure is with components

beta the vector of estimated coefficient

b intercept

Note

QR.admm(x,y,tau) work properly only if the least square estimation is good.

References

S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein. (2010) Distributed Optimization and Statistical Learning via the Alternating Direction. Method of Multipliers *Foundations and Trends in Machine Learning*, **3**, No.1, 1–122

Koenker, Roger. Quantile Regression, New York, 2005. Print.

```
set.seed(1)
n=100
p=2
a=rnorm(n*p, mean = 1, sd =1)
x=matrix(a,n,p)
beta=rnorm(p,1,1)
beta=matrix(beta,p,1)
y=x%*%beta-matrix(rnorm(n,0.1,1),n,1)
# x is 1000*10 matrix, y is 1000*1 vector, beta is 10*1 vector
QR.admm(x,y,0.1)
```

16 QR.cd

\sim	_		-1	
()	к	C	a	

Quantile Regression (QR) use Coordinate Descent (cd) Algorithms

Description

The algorithm base on greedy coordinate descent and Edgeworth's for ordinary l_1 regression.

Usage

```
QR.cd(X,y,tau,beta,maxit,toler)
```

Arguments

Χ	the design matrix
У	response variable
tau	quantile level

beta initial value of estimate coefficient (default naive guess by least square estima-

tion)

maxit maxim iteration (default 200)

toler the tolerance critical for stop the algorithm (default 1e-3)

Value

```
a list structure is with components
```

beta the vector of estimated coefficient

b intercept

Note

QR.cd(x,y,tau) work properly only if the least square estimation is good.

References

Wu, T.T. and Lange, K. (2008). Coordinate Descent Algorithms for Lasso Penalized Regression. *Annals of Applied Statistics*, **2**, No 1, 224–244.

Koenker, Roger. Quantile Regression, New York, 2005. Print.

```
set.seed(1)
n=100
p=2
a=rnorm(n*p, mean = 1, sd =1)
x=matrix(a,n,p)
beta=rnorm(p,1,1)
beta=matrix(beta,p,1)
```

QR.ip

```
y=x%*%beta-matrix(rnorm(n,0.1,1),n,1) # x is 1000*10 matrix, y is 1000*1 vector, beta is 10*1 vector QR.cd(x,y,0.1)
```

QR.ip

Quantile Regression (QR) use Interior Point (ip) Method

Description

The function use the interior point method from quantreg to solve the quantile regression problem.

Usage

```
QR.ip(X,y,tau)
```

Arguments

X the design matrix y response variable tau quantile level

Value

a list structure is with components

beta the vector of estimated coefficient

b intercept

Note

Need to install quantreg package from CRAN.

References

Koenker, Roger. Quantile Regression, New York, 2005. Print.

Koenker, R. and S. Portnoy (1997). The Gaussian Hare and the Laplacian Tortoise: Computability of squared-error vs. absolute-error estimators, with discussion, *Statistical Science*, **12**, 279-300.

```
set.seed(1)
n=100
p=2
a=rnorm(n*p, mean = 1, sd =1)
x=matrix(a,n,p)
beta=rnorm(p,1,1)
beta=matrix(beta,p,1)
y=x%*%beta-matrix(rnorm(n,0.1,1),n,1)
```

18 QR.lasso.admm

```
# x is 1000*10 matrix, y is 1000*1 vector, beta is 10*1 vector #you should install Rmosek first to run following command \#QR.ip(x,y,0.1)
```

QR.lasso.admm

Quantile Regression (QR) with Adaptive Lasso Penalty (lasso) use Alternating Direction Method of Multipliers (ADMM) algorithm

Description

The adaptive lasso parameter base on the estimated coefficient without penalty function. The problem is well suited to distributed convex optimization and is based on Alternating Direction Method of Multipliers (ADMM) algorithm .

Usage

```
QR.lasso.admm(X,y,tau,lambda,rho,beta,maxit)
```

Arguments

X the design matrix y response variable tau quantile level

lambda The constant coefficient of penalty function. (default lambda=1)

rho augmented Lagrangian parameter

beta initial value of estimate coefficient (default naive guess by least square estima-

tion)

maxit maxim iteration (default 200)

Value

a list structure is with components

beta the vector of estimated coefficient

b intercept

Note

QR.lasso.admm(x,y,tau) work properly only if the least square estimation is good.

References

S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein. (2010) Distributed Optimization and Statistical Learning via the Alternating Direction. Method of Multipliers *Foundations and Trends in Machine Learning*, **3**, No.1, 1–122

Wu, Yichao and Liu, Yufeng (2009). Variable selection in quantile regression. *Statistica Sinica*, **19**, 801–817.

QR.lasso.cd

Examples

```
set.seed(1)
n=100
p=2
a=2*rnorm(n*2*p, mean = 1, sd =1)
x=matrix(a,n,2*p)
beta=2*rnorm(p,1,1)
beta=rbind(matrix(beta,p,1),matrix(0,p,1))
y=x%*%beta-matrix(rnorm(n,0.1,1),n,1)
# x is 1000*20 matrix, y is 1000*1 vector, beta is 20*1 vector with last ten zero value elements.
QR.lasso.admm(x,y,0.1)
```

OR.lasso.cd

Quantile Regression (QR) with Adaptive Lasso Penalty (lasso) use Coordinate Descent (cd) Algorithms

Description

The adaptive lasso parameter base on the estimated coefficient without penalty function. The algorithm base on greedy coordinate descent and Edgeworth's for ordinary l_1 regression. As explored by Tong Tong Wu and Kenneth Lange.

Usage

```
QR.lasso.cd(X,y,tau,lambda,beta,maxit,toler)
```

Arguments

X the design matrix y response variable tau quantile level

lambda The constant coefficient of penalty function. (default lambda=1)

beta initial value of estimate coefficient (default naive guess by least square estima-

tion)

maxit maxim iteration (default 200)

toler the tolerance critical for stop the algorithm (default 1e-3)

Value

a list structure is with components

beta the vector of estimated coefficient

b intercept

Note

QR.lasso.cd(x,y,tau) work properly only if the least square estimation is good.

QR.lasso.ip

References

Wu, T.T. and Lange, K. (2008). Coordinate Descent Algorithms for Lasso Penalized Regression. *Annals of Applied Statistics*, **2**, No 1, 224–244.

Wu, Yichao and Liu, Yufeng (2009). Variable selection in quantile regression. *Statistica Sinica*, **19**, 801–817.

Examples

```
set.seed(1)
n=100
p=2
a=2*rnorm(n*2*p, mean = 1, sd =1)
x=matrix(a,n,2*p)
beta=2*rnorm(p,1,1)
beta=rbind(matrix(beta,p,1),matrix(0,p,1))
y=x%*%beta-matrix(rnorm(n,0.1,1),n,1)
# x is 1000*20 matrix, y is 1000*1 vector, beta is 20*1 vector with last ten zero value elements.
QR.lasso.cd(x,y,0.1)
```

QR.lasso.ip

Quantile Regression (QR) with Adaptive Lasso Penalty (lasso) use Interior Point (ip) Method

Description

The function use the interior point method from quantreg to solve the quantile regression problem.

Usage

```
QR.lasso.ip(X,y,tau,lambda)
```

Arguments

X the design matrix y response variable tau quantile level

lambda The constant coefficient of penalty function. (default lambda=1)

Value

a list structure is with components

beta the vector of estimated coefficient

b intercept

lambda The constant coefficient of penalty function. (default lambda=1)

QR.lasso.mm 21

Note

Need to install quantreg package from CRAN.

References

Koenker, R. and S. Portnoy (1997). The Gaussian Hare and the Laplacian Tortoise: Computability of squared-error vs. absolute-error estimators, with discussion, *Statistical Science*, **12**, 279-300.

Wu, Yichao and Liu, Yufeng (2009). Variable selection in quantile regression. *Statistica Sinica*, **19**, 801–817.

Examples

```
set.seed(1)
n=100
p=2
a=2*rnorm(n*2*p, mean = 1, sd =1)
x=matrix(a,n,2*p)
beta=2*rnorm(p,1,1)
beta=rbind(matrix(beta,p,1),matrix(0,p,1))
y=x%**%beta-matrix(rnorm(n,0.1,1),n,1)
# x is 1000*20 matrix, y is 1000*1 vector, beta is 20*1 vector with last ten zero value elements.
#you should install Rmosek first to run following command
#QR.lasso.ip(x,y,0.1)
```

QR.lasso.mm

Quantile Regression (QR) with Adaptive Lasso Penalty (lasso) use Majorize and Minimize (mm) algorithm

Description

The adaptive lasso parameter base on the estimated coefficient without penalty function. The algorithm majorizing the objective function by a quadratic function followed by minimizing that quadratic.

Usage

```
QR.lasso.mm(X,y,tau,lambda,beta,maxit,toler)
```

Arguments

toler

X	the design matrix.
у	response variable.
tau	quantile level.
lambda	The constant coefficient of penalty function. (default lambda=1)
beta	initial value of estimate coefficient.(default naive guess by least square estimation)
maxit	maxim iteration. (default 200)

the tolerance critical for stop the algorithm. (default 1e-3)

QR.mm

Value

```
a list structure is with components
```

beta the vector of estimated coefficient

b intercept

Note

QR.lasso.mm(x,y,tau) work properly only if the least square estimation is good.

References

David R.Hunter and Runze Li.(2005) Variable Selection Using MM Algorithms, *The Annals of Statistics* **33**, Number 4, Page 1617–1642.

Examples

```
set.seed(1)
n=100
p=2
a=2*rnorm(n*2*p, mean = 1, sd =1)
x=matrix(a,n,2*p)
beta=2*rnorm(p,1,1)
beta=rbind(matrix(beta,p,1),matrix(0,p,1))
y=x%*%beta-matrix(rnorm(n,0.1,1),n,1)
# x is 1000*20 matrix, y is 1000*1 vector, beta is 20*1 vector with last ten zero value elements.
QR.lasso.mm(x,y,0.1)
```

QR.mm

Quantile Regression (QR) use Majorize and Minimize (mm) algorithm

Description

The algorithm majorizing the objective function by a quadratic function followed by minimizing that quadratic.

Usage

```
QR.mm(X,y,tau,beta,maxit,toler)
```

Arguments

Χ	the design matrix
у	response variable
tau	quantile level

beta initial value of estimate coefficient (default naive guess by least square estima-

tion)

maxit maxim iteration (default 200)

toler the tolerance critical for stop the algorithm (default 1e-3)

QRADMMCPP 23

Value

```
a list structure is with componentsbeta the vector of estimated coefficientb intercept
```

Note

QR.mm(x,y,tau) work properly only if the least square estimation is good.

References

David R.Hunter and Kenneth Lange. Quantile Regression via an MM Algorithm, *Journal of Computational and Graphical Statistics*, **9**, Number 1, Page 60–77

Examples

```
set.seed(1)
n=100
p=2
a=rnorm(n*p, mean = 1, sd =1)
x=matrix(a,n,p)
beta=rnorm(p,1,1)
beta=matrix(beta,p,1)
y=x%*%beta-matrix(rnorm(n,0.1,1),n,1)
# x is 1000*10 matrix, y is 1000*1 vector, beta is 10*1 vector
QR.mm(x,y,0.1)
```

QRADMMCPP

Quantile Regression (QR) use Alternating Direction Method of Multipliers (ADMM) algorithm core computational part

Description

The problem is well suited to distributed convex optimization and is based on Alternating Direction Method of Multipliers (ADMM) algorithm .

QRCDCPP	Quantile Regression (QR) use Coordinate Descent (cd) Algorithms
	core computational part

Description

The algorithm base on greedy coordinate descent and Edgeworth's for ordinary l_1 regression.

24 qrfit

		_		
a	r	f.	i	t

Quantile Regression (qr) model fitting

Description

High level function for estimating parameters by quantile regression

Usage

```
qrfit(X,y,tau,beta,method,maxit,toler,rho)
```

Arguments

Χ	the design matrix
У	response variable
tau	quantile level
method	"mm" for majorize and minimize method, "cd" for coordinate descent method, "admm" for Alternating method of mulipliers method, "ip" for interior point mehod
rho	augmented Lagrangian parameter
beta	initial value of estimate coefficient (default naive guess by least square estimation)
maxit	maxim iteration (default 200)

the tolerance critical for stop the algorithm (default 1e-3)

Value

toler

a list structure is with components

beta the vector of estimated coefficient

b intercept

Note

qrfit(x,y,tau) work properly only if the least square estimation is good. Interior point method is done by quantreg.

qrfit.lasso 25

qrfit.lasso	Quantile Regression (qr) with Adaptive Lasso Penalty (lasso)	

Description

High level function for estimating and selecting parameter by quantile regression with adaptive lasso penalty.

Usage

```
qrfit.lasso(X,y,tau,lambda,beta,method,maxit,toler,rho)
```

Arguments

Χ	the design matrix
У	response variable
tau	quantile level
method	"mm" for majorize and minimize method,"cd" for coordinate descent method, "admm" for Alternating method of mulipliers method,"ip" for interior point mehod
lambda	The constant coefficient of penalty function. (default lambda=1)
rho	augmented Lagrangian parameter
beta	initial value of estimate coefficient (default naive guess by least square estimation)
maxit	maxim iteration (default 200)
toler	the tolerance critical for stop the algorithm (default 1e-3)

Value

a list structure is with components

beta the vector of estimated coefficient

b intercept

Note

qrfit.lasso(x,y,tau) work properly only if the least square estimation is good. Interior point method is done by quantreg.

26 QRPMMCPP

QRMMCPP	Quantile Regression (QR) use Majorize and Minimize (mm) algorithm core computational part

Description

The algorithm majorizing the objective function by a quadratic function followed by minimizing that quadratic.

QRPADMMCPP	Quantile Regression (QR) with Adaptive Lasso Penalty (lasso) use Alternating Direction Method of Multipliers (ADMM) algorithm core computational part
	computational part

Description

The adaptive lasso parameter base on the estimated coefficient without penalty function. The problem is well suited to distributed convex optimization and is based on Alternating Direction Method of Multipliers (ADMM) algorithm .

QRPCDCPP	Quantile Regression (QR) with Adaptive Lasso Penalty (lasso) use Co-
	ordinate Descent (cd) Algorithms core computational part

Description

The adaptive lasso parameter base on the estimated coefficient without penalty function. The algorithm base on greedy coordinate descent and Edgeworth's for ordinary l_1 regression. As explored by Tong Tong Wu and Kenneth Lange.

QRPMMCPP	Quantile Regression (QR) with Adaptive Lasso Penalty (lasso) use Majorize and Minimize (mm) algorithm core computational part

Description

The adaptive lasso parameter base on the estimated coefficient without penalty function. The algorithm majorizing the objective function by a quadratic function followed by minimizing that quadratic.

Index

cqr.admm, 2 cqr.lasso.admm, 8 cQRADMMCPP, 13 cQRADMMCPP, 13 qR.admm, 14 qR.lasso.admm, 18 qRADMMCPP, 23 qQRPADMMCPP, 26 *CD cqr.d, 3 cqr.lasso.cd, 9 cQRCDCPP, 13 cQRPCDCPP, 14 qR.cd, 16 qR.lasso.cd, 19 qRCDCPP, 23 qQRCDCPP, 23 qQRCDCPP, 26 *Composite quantile regression cqr.fit, 5 cqr.fit.lasso, 6 *IP cqr.ip, 7 qR.ip, 17 qR.lasso.ip, 20 *MM cqr.mm, 11 cQRMMCPP, 13 cqr.MMCPP, 23 qRRDCPP, 26 *MM cqr.ip, 7 qR.ip, 17 qR.lasso.im, 10 cqr.mm, 11 cQRMMCPP, 13 cqr.MMCPP, 23 qRCDCPP, 24 qR.lasso.ip, 20 qR.lasso.ip, 20 qR.lasso.ip, 20 qR.lasso.ip, 20 qRADMMCPP, 23 qRCDCPP, 23 qRefit, 24 qrfit.lasso, 25 cqr.mm, 22 qRMMCPP, 14 qR.lasso.mm, 21 qR.mm, 22 qRMMCPP, 26 qRMMC	* ADMM	cqr.fit,5
CQRADMMCPP, 13	cqr.admm,2	cqr.fit.lasso,6
CQRPADMMCPP, 13 QR. admm, 14 QR. lasso.admm, 18 QRADMMCPP, 23 QRPADMMCPP, 26 CQRCDCPP, 13 CQRPADMMCPP, 13 CQRPCDCPP, 13 CQRPCDCPP, 14 QR. cd, 16 QR. lasso.cd, 19 QRCDCPP, 23 QRPCDCPP, 26 * Composite quantile regression cqr.fit, 5 cqr.fit.lasso, 6 * IP CqR.ip, 17 QR.lasso.ip, 20 * MIM cqr.lasso.mm, 10 cqr.mm, 11 CQRPMCPP, 13 CQRPMCPP, 14 QR. cd, 16 QR. admm, 14 QR. cd, 16 QR. admm, 14 QR. cd, 16 QR. admm, 14 QR. cd, 16 QR. lasso. cd, 19 QR. lasso. cd, 19 QR. lasso. cd, 19 QR. lasso. de, 10 QR. lasso. de, 19 QR.	cqr.lasso.admm,8	cqr.ip,7
QR.admm, 14 QR.lasso.admm, 18 QRADMMCPP, 23 QRPADMMCPP, 26 CQRCDCPP, 13 cqr.cd, 3 cqr.lasso.cd, 9 CQRCDCPP, 14 QR.cd, 16 QR.lasso.cd, 19 QRCDCPP, 23 QRPCDCPP, 24 QRCDCPP, 23 QRPCDCPP, 26 ** Composite quantile regression cqr.fit, 5 cqr.ip, 7 QR.ip, 17 QR.lasso.ip, 20 ** MM cqr.lasso.ip, 20 ** MM cqr.lasso.mm, 10 cqr.mm, 11 CQRMMCPP, 13 CQRPMCPP, 24 QR.dlesso.ip, 20 QR.lasso.ip, 20 QR.lasso.ip, 20 QR.lasso.ip, 20 QRADMMCPP, 23 QRCDCPP, 24 QR.lasso.ip, 20 QR.lasso.ip, 20 QR.lasso.ip, 20 QR.lasso.ip, 20 QRADMMCPP, 23 QRCDCPP, 23 QRCDCPP, 23 QRADMMCPP, 23 QRCDCPP, 23 QRADMMCPP, 23 QRADMMCPP, 23 QRADMMCPP, 26 QRADMMCPP, 26 QRADMMCPP, 26 QRPMMCPP, 26 QRP	CQRADMMCPP, 13	cqr.lasso.admm,8
QR.lasso.admm, 18 QRADMMCPP, 23 QRPADMMCPP, 26 CD Cqr.cd, 3 cqr.lasso.cd, 9 CQRCDCPP, 13 CQRPCDCPP, 13 CQRPCDCPP, 14 QR.cd, 16 QR.lasso.cd, 19 QRCDCPP, 23 QRPCDCPP, 26 **Composite quantile regression cqr.fit, 5 cqr.ip, 7 QR.lasso.ip, 20 **MM cqr.lasso.im, 10 cqr.mm, 11 CQRMCPP, 13 CQRPMCPP, 14 QR.cd, 16 QR.lasso.ip, 20 QR.admm, 14 QR.cd, 16 QR.lasso.ip, 20 QR.lasso.ip, 20 QR.lasso.ip, 20 QR.lasso.ip, 20 QRADMMCPP, 23 QRCDCPP, 23 QRPCDCPP, 23 QRADMMCPP, 23 QRADMMCPP, 23 QRADMMCPP, 23 QRADMMCPP, 26 QRADMMCPP, 26 QRADMMCPP, 26 QRADMMCPP, 26 QRADMMCPP, 26 QRADMMCPP, 26 QRADMCPP, 26 QRADM	CQRPADMMCPP, 13	cqr.lasso.cd, 9
QRADMMCPP, 23 QRPADMMCPP, 26 * CD cqr.cd, 3 cqr.lasso.cd, 9 CQRCDCPP, 13 CQRPCDCPP, 13 CQRPCDCPP, 13 CQRPCDCPP, 13 CQRPCDCPP, 14 QR.cd, 16 QR.lasso.cd, 19 QRCDCPP, 23 QRPCDCPP, 26 * Composite quantile regression cqr.fit, 5 cqr.fit.lasso, 6 * IP cqr.ip, 7 QR.lasso.ip, 20 * MM cqr.lasso.mm, 10 cqr.mm, 11 CQRMMCPP, 14 QR.lasso.mm, 21 QR.lasso.mm, 21 QR.lasso.mm, 21 QR.lasso.mm, 21 QR.lasso.mm, 22 QRMMCPP, 26 * QRADMMCPP, 26 QRADMMCPP, 26 QRADMMCPP, 26 QRADMMCPP, 26 QRADMMCPP, 26 QRPMMCPP, 26 QRPM	QR.admm, 14	cqr.lasso.mm, 10
QRPADMMCPP, 26 CQRCDCPP, 13 * CD CQRMMCPP, 13 cqr.cd, 3 CQRPADMMCPP, 13 cqr.lasso.cd, 9 CQRPCDCPP, 14 CQRCDCPP, 13 CQRPMCPP, 14 QR.cd, 16 QR.lasso.cd, 19 QRCDCPP, 23 QR.cd, 16 QRCDCPP, 26 QR.ip, 17 * Composite quantile regression QR.ip, 17 cqr.fit.5 QR.lasso.admm, 18 cqr.ip, 7 QR.lasso.ip, 20 * IP QR.lasso.ip, 20 cqr.ip, 7 QR.lasso.ip, 20 RMM QRCDCPP, 23 cqr.lasso.mm, 10 qrfit, 24 cqr.mm, 11 qrfit, 24 QR.lasso.mm, 21 QRADMMCPP, 26 QRPADMCPP, 26 QRPADMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 QRDCPP, 26 QRDCPP, 26 QRPMCPP, 26 QRPMMCPP, 26	QR.lasso.admm, 18	cqr.mm, 11
* CD	QRADMMCPP, 23	CQRADMMCPP, 13
cqr.cd, 3 cqr.lasso.cd, 9 CQRCDCPP, 13 CQRCDCPP, 14 QR.cd, 16 QR.lasso.cd, 19 QRCDCPP, 23 QRPCDCPP, 26 * Composite quantile regression cqr.fit, 5 cqr.fit.lasso, 6 * IP cqr.ip, 7 QR.lasso.ip, 20 * MMM cqr.lasso.mm, 10 cqr.mm, 11 CQRMMCPP, 13 CQRPMMCPP, 14 QR.lasso.mm, 21 QR.mm, 22 QRMMCPP, 13 CQRMMCPP, 14 QR.lasso.mm, 21 QR.mm, 22 QRMMCPP, 26 QRPMMCPP, 26 QRPMCPP, 26 QRPMCP	QRPADMMCPP, 26	CQRCDCPP, 13
cqr.lasso.cd, 9 CQRPCDCPP, 14 CQRCDCPP, 13 CQRPMMCPP, 14 CQR.cd, 16 list, 3–9, 11, 12, 15–20, 22–25 QR.lasso.cd, 19 QR.admm, 14 QRCDCPP, 23 QR.cd, 16 QRPCDCPP, 26 QR.cd, 16 * Composite quantile regression QR.ip, 17 cqr.fit, 5 QR.lasso.admm, 18 cqr.ip, 7 QR.lasso.ip, 20 * IP QR.lasso.ip, 20 cqr.ip, 7 QR.lasso.mm, 21 QR.lasso.ip, 20 QRADMMCPP, 23 qRDMCPP, 23 QRCDCPP, 23 qrfit, 24 qrfit, 24 qr, 1asso.mm, 10 qrfit, 24 QRMMCPP, 13 QRPADMMCPP, 26 QRMMCPP, 26 QRPADMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 QRPMICPR, 24 QRILLASSO, 25	* CD	CQRMMCPP, 13
CQRCDCPP, 13	cqr.cd, 3	CQRPADMMCPP, 13
CQRPCDCPP, 14 QR.cd, 16 QR.lasso.cd, 19 QRCDCPP, 23 QRPCDCPP, 26 * Composite quantile regression cqr.fit, 5 cqr.fit.lasso, 6 * IP Cqr.ip, 7 QR.lasso.ip, 20 * MM cqr.lasso.mm, 10 cqr.mm, 11 CQRMMCPP, 13 CQRPMCPP, 14 QR.lasso.mm, 21 QR.lasso.mm, 21 QR.mm, 22 QRMMCPP, 16 QR.lasso.mm, 21 QRMPCPP, 26 QRMPMCPP, 26 QRMMCPP, 26 QRMMC	cqr.lasso.cd,9	CQRPCDCPP, 14
QR. cd, 16 QR. lasso.cd, 19 QRCDCPP, 23 QRPCDCPP, 26 * Composite quantile regression cqr.fit, 5 cqr.fit.lasso, 6 * IP cqr.ip, 7 QR. lasso.ip, 20 * MM cqr.lasso.mm, 10 cqr.mm, 11 CQRMMCPP, 13 CQRPMCPP, 14 QR. lasso.mm, 21 QR. mm, 22 QRMMCPP, 14 QR. lasso.mm, 21 QR. mm, 22 QRMMCPP, 26 QRPMCPP, 26 QRPMC	CQRCDCPP, 13	CQRPMMCPP, 14
QR.lasso.cd, 19 QRCDCPP, 23 QRPCDCPP, 26 * Composite quantile regression cqr.fit, 5 cqr.fit.lasso, 6 * IP cqr.ip, 7 QR.lasso.ip, 20 * MM cqr.lasso.mm, 10 cqr.mm, 11 cQRMMCPP, 13 cQRPMCPP, 14 QR.lasso.mm, 21 QR.lasso.mm, 21 QR.mm, 22 QRMMCPP, 26 QRPMMCPP, 26 * Quantile regression qrfit, 24 qrfit.lasso, 25 cqr.admm, 2	CQRPCDCPP, 14	
QRCDCPP, 23 QRPCDCPP, 26 * Composite quantile regression cqr.fit, 5 cqr.fit.lasso, 6 * IP cqr.ip, 7 QR.lasso.ip, 20 R.lasso.ip, 20 * MM cqr.lasso.mm, 10 cqr.mm, 11 CQRMMCPP, 13 CQRPMMCPP, 14 QR.lasso.mm, 21 QR.lasso.mm, 21 QR.lasso.mm, 21 QR.mm, 22 QRMMCPP, 26 QRPMMCPP, 26 CQRPMCPP, 26 QRPMMCPP, 26 CQRPMCPP, 2	QR.cd, 16	list, 3–9, 11, 12, 15–20, 22–25
QRCDCPP, 26 * Composite quantile regression	QR.lasso.cd, 19	
**Composite quantile regression cqr.fit, 5 cqr.fit.lasso, 6 **IP **Composite quantile regression cqr.fit, 5 cqr.fit.lasso, 6 **IP **Cqr.ip, 7 QR.lasso.ip, 20 **MM **Cqr.lasso.mm, 10 cqr.mm, 11 CQRMMCPP, 13 CQRPMMCPP, 14 QR.lasso.mm, 21 QR.mm, 22 QRADMMCPP, 26 QRPMMCPP, 26 QRMMCPP, 26 QRMMCPP, 26 **QRMMCPP, 26 **Quantile regression **qrfit, 24 **qrfit.lasso, 25 **Cqr.admm, 2	QRCDCPP, 23	
QR.lasso.admm, 18 cqr.fit, 5 cqr.fit.lasso, 6 * IP cqr.ip, 7 QR.ip, 17 QR.lasso.ip, 20 * MM cqr.lasso.mm, 10 cqr.mm, 11 CQRMMCPP, 13 CQRPMMCPP, 14 QR.lasso.mm, 21 QR.lasso.mm, 21 QR.MMCPP, 26 QRPDMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 QRPMCPP, 26 CQRPMCPP, 26 QRPMCPP, 26 CQRPMCPP, 26 CQRPM	QRPCDCPP, 26	- · · · · · · · · · · · · · · · · · · ·
<pre>cqr.fit.lasso,6 cqr.fit.lasso,6 * IP</pre>	* Composite quantile regression	
* IP cqr.ip, 7 QR.ip, 17 QR.mm, 22 QRADMMCPP, 23 * MM cqr.lasso.mm, 10 cqr.mm, 11 CQRMMCPP, 13 CQRPMMCPP, 14 QR.lasso.mm, 21 QR.mm, 22 QRADMMCPP, 26 QRPADMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 CQRMMCPP, 26 CQRMMCPP, 26 CQRPMMCPP,	cqr.fit,5	
* IP cqr.ip, 7 QR.ip, 17 QR.lasso.ip, 20 * MMM cqr.lasso.mm, 10 cqr.mm, 11 CQRMMCPP, 13 CQRPMMCPP, 14 QR.lasso.mm, 21 QR.mm, 22 QRMDCPP, 26 QRPMMCPP, 26 QRMMCPP, 26 QRMMCPP, 26 QRMMCPP, 26 QRMMCPP, 26 QRMMCPP, 26 QRMMCPP, 26 CQRMMCPP, 2	cqr.fit.lasso,6	-
QR.ip, 17 QR.ip, 17 QR.lasso.ip, 20 * MIM cqr.lasso.mm, 10 cqr.mm, 11 CQRMMCPP, 13 CQRPMMCPP, 14 QR.lasso.mm, 21 QR.MmCPP, 26 QRPMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 CQRPMMCPP, 26 CQRPMMCP		
QR.ip, 17 QR.lasso.ip, 20 * MIM cqr.lasso.mm, 10 cqr.mm, 11 CQRMMCPP, 13 CQRPMCPP, 14 QR.lasso.mm, 21 QR.mm, 22 QRMDCPP, 26 QRPMCPP, 26 CQRMMCPP, 26 CQRMMCPP, 26 CQRMCPP, 26 CQRMCPP, 26 CQRMCPP, 26 CQRMCPP, 26 CQRPMCPP, 26 CQRMCPP, 26 CQRPMCPP, 26 CQRMCPP, 26	cqr.ip,7	-
* MM QRCDCPP, 23 cqr.lasso.mm, 10 cqr.mm, 11 CQRMMCPP, 13 CQRPMMCPP, 14 QR.lasso.mm, 21 QR. nmm, 22 QRMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 CQRPMMCPP, 26 QRPMMCPP, 26 CQRPMMCPP,		
cqr.lasso.mm, 10 cqr.mm, 11 CQRMMCPP, 13 CQRPMMCPP, 14 QR.lasso.mm, 21 QR.mm, 22 QRMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 CQRPMMCPP, 26 CQRPMMCPP, 26 CQRPMMCPP, 26 CQRPMMCPP, 26 CQRPMTCPP, 2	QR.lasso.ip, 20	
cqr.mm, 11	* MM	=
QRMMCPP, 26 QRPADMMCPP, 26 QRPADMMCPP, 26 QRPCDCPP, 26 QRPCDCPP, 26 QRPMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 * Quantile regression qrfit, 24 qrfit.lasso, 25 cqr.admm, 2	cqr.lasso.mm, 10	·
CQRPMMCPP, 15 CQRPMMCPP, 14 QR.lasso.mm, 21 QR.mm, 22 QRMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 * Quantile regression qrfit, 24 qrfit.lasso, 25 cqr.admm, 2	cqr.mm, 11	·
QRPCDCPP, 26 QR. lasso.mm, 21 QR. mm, 22 QRMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 * Quantile regression qrfit, 24 qrfit.lasso, 25 cqr.admm, 2	CQRMMCPP, 13	
QRPMMCPP, 26 QRPMMCPP, 26 QRPMMCPP, 26 * Quantile regression qrfit, 24 qrfit.lasso, 25 cqr.admm, 2	CQRPMMCPP, 14	-
QR.HIIII, 22 QRMMCPP, 26 QRPMMCPP, 26 * Quantile regression qrfit, 24 qrfit.lasso, 25 cqr.admm, 2	QR.lasso.mm, 21	-
QRMMCPP, 26 QRPMMCPP, 26 * Quantile regression qrfit, 24 qrfit.lasso, 25 cqr.admm, 2		QRPMMCPP, 26
QRPMMCPP, 26 * Quantile regression qrfit, 24 qrfit.lasso, 25 cqr.admm, 2		
* Quantile regression qrfit, 24 qrfit.lasso, 25 cqr.admm, 2		
<pre>qrfit, 24 qrfit.lasso, 25 cqr.admm, 2</pre>	* Quantile regression	
qrfit.lasso, 25 cqr.admm, 2		
·	qrfit.lasso, 25	
•	cqr.admm, 2	
	•	