

## USAGE

`rq.no.cross(y, x, taus)`

## ARGUMENTS

<code>y</code>	Response vector of length $n$
<code>x</code>	Matrix of predictors with $p$ columns and $n$ rows. A column of ones will automatically be added for the intercept. This should not be included in the design matrix
<code>taus</code>	Vector of desired quantiles to perform the regression. Each value should be in $(0, 1)$ . At least two values are required, otherwise the function <code>rq</code> should be used.

## VALUE

<code>bhat</code>	Matrix of estimated coefficients (including the intercept). This is a $(p + 1) \times m$ matrix, where $p$ is the number of predictors and $m$ is the number of quantiles. Each column represents a quantile level (in ascending order of the input vector of quantiles). Each row represents a predictor variable. The first row gives the estimated intercept at each quantile, while the remaining rows give the slopes.
<code>se.bhat</code>	Matrix corresponding to the standard errors for each of the coefficient estimates. Each entry of the matrix denotes the standard error for the estimated coefficient in the corresponding location of the <code>bhat</code> matrix.
<code>cov.bhat</code>	A 3-dimensional array of $m$ matrices, each is a $(p+1) \times (p+1)$ covariance matrix. Each matrix is the variance-covariance matrix for the estimated coefficients at each of the $m$ different quantile levels.
<code>taus</code>	The ordered values of quantiles used.

## NOTES

This function performs linear quantile regression while enforcing the non-crossing constraints as in Bondell, Reich, and Wang (2010). The quantile functions will be non-crossing on the full rectangle formed by the product space of the univariate range of each of the predictors. Each predictor is internally scaled to be on the interval [0,1], and non-crossing is enforced on the unit hypercube, then everything is transformed back to the original scale of the data.

## REFERENCE

Bondell, H. D., Reich, B. J., and Wang, H. (2010). Non-crossing quantile regression curve estimation. *Biometrika*, to appear.

## EXAMPLE

```
n = 200
p = 3
beta = rep(1,p)
x = matrix(rnorm(n*p), nrow=n, ncol=p)
y = x%%beta + rnorm(n)
example_fit = rq.no.cross(y, x, taus=seq(0.1,0.9,.1))
```