

Package ‘ConfidenceQuant’

May 15, 2018

Type Package

Title Computation of Confidence Bands and Regions for Comparative Experiments

Version 1.1.0

Author Likun Zhang <lfz5044@psu.edu>

Maintainer Likun Zhang <lfz5044@psu.edu>

Description The ConfidenceQuant R package developed for the computation of confidence bands and regions for comparative experiments fits Penalized Quantile Smoothing Splines using the rqss function in the quantreg package, developed by R. Koenker.

License Internal use.

Imports quantreg,
doParallel,
foreach,
splines,
tripack,
akima,
rootSolve,
plotly,
ggplot2,
grDevices,
scales

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

R topics documented:

BLB_Discretize	2
BLB_Discretize_2dim	4
BLB_rqss	6
BLB_rqss_2dim	7
Bootstrap_rqss	9
Bootstrap_rqss_2dim	10
CompExp	12
CompExperiment_1dim	14
CompExperiment_1dim_boot	16

CompExperiment_1dim_D	18
CompExperiment_2dim	21
CompExperiment_2dim_boot	23
CompExperiment_2dim_D	25
control	28
control_2dim	28
foldsGenerator	29
lambdaSearch	29
multifoldCV	31
one	32
rqss_new	33
SurfaceFit	35
treatment	37
treatment_2dim	37
two	38

Index	39
--------------	-----------

BLB_Discretize	<i>Bag of Little Bootstraps (Discretized).</i>
----------------	--

Description

BLB_Discretize Bag of Little Bootstraps (BLB) for data sets with 1-dim covariate. Used to generate confidence bands for a quantile smoothing splines fitted with the `rqss` function from package [quantreg](#). What's special about this function is that it discretizes the data set to decrease the sample size, and it utilizes ALL the observations.

Usage

```
BLB_Discretize(cores = NULL, data, m = 80, alpha = 0.05, tau = 0.25,
  lambda = 2, D = 100, s = cores, b = floor(nrow(data)/s), r = 100,
  Range = c(0.001, 0.999), Search = FALSE)
```

Arguments

cores	The number of cores to use for parallel execution. If not specified, the number of cores is set to the value of <code>options("cores")</code> , if specified, or to one-half the number of cores detected by the parallel package.
data	A 2-dim optional data frame (or object coercible by as.data.frame to a data frame) containing the variables in the model. The column names should be specified in a way that “x” is for the predictor, and “y” for the response.
m	A numeric value that controls how fine that data set should be discretized.
alpha	The confidence level required. The default is 0.05.
tau	A specific quantile to be estimated. Must be a number between 0 and 1.
lambda	The smoothing parameter governing the tradeoff between fidelity and the penalty component for the triogram term. If <code>Search=TRUE</code> , there is no need for users to specify a value.
D	A number that specifies for how many x values you want to compute confidence intervals; the confidence band is made of pointwise intervals at various x values. If specified, it will look at <i>D</i> equidistant points.

s	The number of subsamples used in BLB algorithm. The default value is the number of cores the user assigned for the function.
b	The subsample size in the BLB algorithm. Kleiner et al. suggest that the size should be around $n^{0.6}$, where n is the data size. It is better to have $b * s = nrow(data)$ so the asymptotic property is achieved.
r	The number of bootstrap iterations (samples with with replacement). $r = 100$ is suggested.
Range	A vector with 2 values that specifys the range of the data set where the user wants to perform BLB over. It is defined using lower and upper quantile. The default value is (0.001, 0.999).
Search	If TRUE (which is recommended), then the function will first search for an optimum smoothing parameter λ .

Details

This function is based on the "Bag of Little Bootstraps" (BLB) method by Kleiner et al.(2012). It performs parallelization to speed up the calculation.

Value

A list with three parts:

1. `x0` and `CI_average`, where `x0` contains the x values at which the confidence intervals are evaluated, and `CI_average` is 2-dim matrix which contains the corresponding lower and upper bounds.
2. `lambda`, which is the optimum smoothing parameter selected by `BLB_Discretize`. If it is done automatically, the function also returns `Lambda` and `Fid`, which respectively stand for a vector of `lambda` values and their corresponding cross-validation MCV values.

References

Micheal, I. J et al.(2012). *AScalableBootstrapforMassiveData*.

See Also

[BLB_Discretize_2dim](#) for BLB with 2-dim covariate dataset.

Examples

```
data(one)
result<-BLB_Discretize(cores=7,data=one, alpha=0.05, tau=0.5, Search=TRUE)

plot<-FitPlot_1dim(data=one, result=result, xlab='x',ylab='y')
plot
```

BLB_Discretize_2dim *Bag of Little Bootstraps (Discretized).*

Description

BLB_Discretize_2dim Bag of Little Bootstraps (BLB) for data sets with 2-dim covariate. Used to generate confidence bands for a quantile smoothing splines fitted with the `rqss` function from package [quantreg](#). What's special about this function is that it discretizes the data set to decrease the sample size, and it utilizes ALL the observations.

Discretize is a function to discretize the data set.

Usage

```
BLB_Discretize_2dim(cores = NULL, data, m = 30, alpha = 0.05,
  tau = 0.25, lambda = 2, D = 50, s = cores, b = floor(nrow(data)/s),
  r = 100, Range = c(0.001, 0.999), Search = FALSE)
```

```
Discretize(dats, m = 30)
```

Arguments

cores	The number of cores to use for parallel execution. If not specified, the number of cores is set to the value of <code>options("cores")</code> , if specified, or to one-half the number of cores detected by the parallel package.
data	A 3-dim optional data frame (or object coercible by as.data.frame to a data frame) containing the variables in the model. The column names should be specified in a way that "x1" and "x2" are for the two predictors, and "y" for the response.
m	A numeric value that controls how fine that data set should be discretized.
alpha	The confidence level required. The default is 0.05.
tau	A specific quantile to be estimated. Must be a number between 0 and 1.
lambda	The smoothing parameter governing the tradeoff between fidelity and the penalty component for the triogram term. If <code>Search=TRUE</code> , there is no need for users to specify a value.
D	A number that specifies for how many x values you want to compute confidence intervals; the confidence surface is made of pointwise intervals at various x values. If specified, it will compute intervals on an equidistant DD grid.
s	The number of subsamples used in BLB algorithm. The default value is the number of cores the user assigned for the function.
b	The subsample size in the BLB algorithm. Kleiner et al. suggest that the size should be around $n^{0.6}$, where n is the data size. It is better to have $b * s = nrow(data)$ so the asymptotic property is achieved.
r	The number of bootstrap iterations (samples with replacement). $r = 100$ is suggested.
Range	A vector with 2 values that specifies the range of the data set where the user wants to perform BLB over. It is defined using lower and upper quantile. The default value is (0.001, 0.999).
Search	If TRUE (which is recommended), then the function will first search for an optimum smoothing parameter λ .

Details

This function is based on the "Bag of Little Bootstraps" (BLB) method by Kleiner et al.(2012). It performs parallelization to speed up the calculation.

Value

A list with three parts:

1. `x0` and `CI_average`, where `x0` contains the `x` values at which the confidence intervals are evaluated, and `CI_average` is 2-dim matrix which contains the corresponding lower and upper bounds.
2. `lambda`, which is the optimum smoothing parameter chosen by `BLB_Discretize_2dim`. If it is done automatically, the function also returns `Lambda` and `Fid`, which respectively stand for a vector of `lambda` values and their corresponding cross-validation MCV values.

References

- Kleiner, I. J et al. JRSS B, 2012. *AScalableBootstrapforMassiveData*.
 Roger Koenker.(2005). *Quantileregression*. Cambridge university press.

See Also

[BLB_rqss_2dim](#) for BLB without the discretization.

Examples

```
data(two)
result<-BLB_Discretize_2dim(cores=7, data=two, alpha=0.05, tau=0.5, Search=TRUE)

#1. 3D plot
plot<-FitPlot_2dim(data=two, result=result, xlab='x', ylab='y', zlab='z')
plot

#2. Contour plot
#Plot the confidence bands
library(akima)
valid<-!is.na(result$CI_average[,1])
X<-result$x0$x1[valid];Y<-result$x0$x2[valid]
lower<-result$CI_average[valid,1]; upper<-result$CI_average[valid,2]
x1range<-range(X);x2range<-range(Y)
akima1<-interp(X,Y,lower,xo=seq(x1range[1],x1range[2],length=200),yo=seq(x2range[1],x2range[2],length=200))
akima2<-interp(X,Y,upper,xo=seq(x1range[1],x1range[2],length=200),yo=seq(x2range[1],x2range[2],length=200))

plot(X,Y,main="Prediction Contour Plot",type="n",xlab="x1",ylab="x2")
image(akima1,add=TRUE,col = terrain.colors(12)) ###YOU CAN CHANGE THE NUMBER OF COLOR LEVELS!!!!
contour(akima1,add=TRUE,nlevels=12)
```

BLB_rqss

*Bag of Little Bootstraps.***Description**

BLB_rqss Bag of Little Bootstraps (BLB) for data sets with 1-dim covariate. Used to generate confidence bands for a quantile smoothing splines fitted with the rqss function from package [quantreg](#).

FitPlot_1dim is a function to plot the results.

Usage

```
BLB_rqss(cores = NULL, data, alpha = 0.05, tau = 0.25, lambda = 2,
  D = 100, b = ceiling(nrow(data)^0.6), s = 15, r = 100,
  Search = FALSE)
```

```
FitPlot_1dim(data, result, xlab = "x", ylab = "y")
```

Arguments

cores	The number of cores to use for parallel execution. If not specified, the number of cores is set to the value of options("cores"), if specified, or to one-half the number of cores detected by the parallel package.
data	A 2-dim optional data frame (or object coercible by as.data.frame to a data frame) containing the variables in the model. The column names should be specified in a way that "x" is for the predictor, and "y" for the response.
alpha	The confidence level required. The default is 0.05.
tau	A specific quantile to be estimated. Must be a number between 0 and 1.
lambda	The smoothing parameter governing the tradeoff between fidelity and the penalty component for the triogram term. If Search=TRUE, there is no need for users to specify a value.
D	A number that specifies for how many x values you want to compute confidence intervals; the confidence band is made of pointwise intervals at various x values. If specified, it will look at D equidistant points.
b	The subsample size in the BLB algorithm. Kleiner et al. suggest that the size should be around $n^{0.6}$, where n is the data size.
s	The number of subsamples used in BLB algorithm. Jordan et al. suggest that s should be 10~20.
r	The number of bootstrap iterations (samples with with replacement). $r = 100$ is suggested.
Search	If TRUE (which is recommended), then the function will first search for an optimum smoothing parameter λ .
xlab, ylab	Titles for x and y axis. See title .

Details

This function is based on the "Bag of Little Bootstraps" (BLB) method by Kleiner et al.(2012). It performs parallelization to speed up the calculation.

Value

A list with three parts:

1. `x0` and `CI_average`, where `x0` contains the `x` values at which the confidence intervals are evaluated, and `CI_average` is 2-dim matrix which contains the corresponding lower and upper bounds.
2. `lambda`, which is the optimum smoothing parameter selected by `BLB_rqss`. If it is done automatically, the function also returns `Lambda` and `Fid`, which respectively stand for a vector of `lambda` values and their corresponding cross-validation MCV values.

References

Micheal, I. J et al.(2012). *AScalableBootstrapforMassiveData*.

See Also

[BLB_rqss_2dim](#) for BLB with 2-dim covariate dataset.

Examples

```
data(one)
result<-BLB_rqss(cores=7, data=one, alpha=0.05, tau=0.5, Search=TRUE)

plot<-FitPlot_1dim(data=one, result=result, xlab='x',ylab='y')
plot
```

BLB_rqss_2dim

Bag of Little Bootstraps.

Description

`BLB_rqss_2dim` Bag of Little Bootstraps (BLB) for data sets with 2-dim covariate. Used to generate confidence bands for a quantile smoothing splines fitted with the `rqss` function from package [quantreg](#).

`FitPlot_2dim` is a function to plot the results.

Usage

```
BLB_rqss_2dim(cores = NULL, data, alpha = 0.05, tau = 0.25, lambda = 2,
  D = 50, b = ceiling(nrow(data)^0.6), s = 15, r = 100,
  Search = FALSE)

FitPlot_2dim(data, result, Range = c(0.002, 0.998), xlab = "log(avtp_med)",
  ylab = "log(niqr_med)", zlab = "log(playdelay_med)")
```

Arguments

`cores` The number of cores to use for parallel execution. If not specified, the number of cores is set to the value of `options("cores")`, if specified, or to one-half the number of cores detected by the [parallel](#) package.

data	A 3-dim optional data frame (or object coercible by as.data.frame to a data frame) containing the variables in the model. The column names should be specified in a way that "x1" and "x2" are for the two predictors, and "y" for the response.
alpha	The confidence level required. The default is 0.05.
tau	A specific quantile to be estimated. Must be a number between 0 and 1.
lambda	The smoothing parameter governing the tradeoff between fidelity and the penalty component for the triogram term. If Search=TRUE, there is no need for users to specify a value.
D	A number that specifies for how many x values you want to compute confidence intervals; the confidence surface is made of pointwise intervals at various x values. If specified, it will compute intervals on an equidistant DD grid.
b	The subsample size in the BLB algorithm. Kleiner et al. suggest that the size should be around $n^{0.6}$, where n is the data size.
s	The number of subsamples used in BLB algorithm. Kleiner et al. suggest that s should be 10~20.
r	The number of bootstrap iterations (samples with with replacement). $r = 100$ is suggested.
Search	If TRUE (which is recommended), then the function will first search for an optimum smoothing parameter λ .
xlab, ylab, zlab	Titles for the axes. See title .

Details

This function is based on the "Bag of Little Bootstraps" (BLB) method by Kleiner et al.(2012). It performs parallelization to speed up the calculation.

Value

A list with three parts:

1. x_0 and $CI_average$, where x_0 contains the x values at which the confidence intervals are evaluated, and $CI_average$ is 2-dim matrix which contains the corresponding lower and upper bounds.
2. $lambda$, which is the optimum smoothing parameter selected by BLB_rqss_2dim. If it is done automatically, the function also returns $Lambda$ and Fid , which respectively stand for a vector of $lambda$ values and their corresponding cross-validation MCV values.

References

- Kleiner, I. J et al. JRSS B, 2012. *AScalableBootstrapforMassiveData*.
 Roger Koenker.(2005). *Quantileregression*. Cambridge university press.

See Also

[BLB_rqss](#) for BLB with 1-dim covariate dataset.

Examples

```
data(two)
result<-BLB_rqss_2dim(cores=7, data=two, alpha=0.05, tau=0.5, Search=TRUE)

#1. 3D plot
plot<-FitPlot_2dim(data=two, result=result, xlab='x', ylab='y', zlab='z')
plot

#2. Contour plot
#Plot the confidence bands
library(akima)
valid<-!is.na(result$CI_average[,1])
X<-result$x0$x1[valid];Y<-result$x0$x2[valid]
lower<-result$CI_average[valid,1]; upper<-result$CI_average[valid,2]
x1range<-range(X);x2range<-range(Y)
akima1<-interp(X,Y,lower,xo=seq(x1range[1],x1range[2],length=200),yo=seq(x2range[1],x2range[2],length=200))
akima2<-interp(X,Y,upper,xo=seq(x1range[1],x1range[2],length=200),yo=seq(x2range[1],x2range[2],length=200))

plot(X,Y,main="Prediction Contour Plot",type="n",xlab="x1",ylab="x2")
image(akima1,add=TRUE,col = terrain.colors(12)) ###YOU CAN CHANGE THE NUMBER OF COLOR LEVELS!!!!
contour(akima1,add=TRUE,nlevels=12)
```

Bootstrap_rqss

Regular Bootstrap Method.

Description

Bootstrap_rqss Regular bootstrap for data sets with 1-dim covariate. Used to generate the confidence bands for a quantile smoothing spline fitted with the rqss function from package [quantreg](#).

Usage

```
Bootstrap_rqss(cores = NULL, data = parent.frame(), alpha = 0.05,
  tau = 0.25, lambda = 2, B = 1500, D = 100, warning.catch = FALSE)
```

Arguments

cores	The number of cores to use for parallel execution. If not specified, the number of cores is set to the value of <code>options("cores")</code> , if specified, or to one-half the number of cores detected by the parallel package.
data	A 2-dim optional data frame (or object coercible by as.data.frame to a data frame) containing the variables in the model. The column names should be specified in a way that “x” is for the predictor, and “y” for the response.
alpha	The confidence level required. The default is 0.05.
tau	A specific quantile to be estimated. Must be a number between 0 and 1.
lambda	The smoothing parameter used to specify the tradeoff between fidelity and roughness of the fitted quantile function.
B	The number of Monte Carlo iterations using bootstrap with replacement. $B = 1500$ is by default.
D	A number that specifies how many x values you want to examine the confidence bands at. If specified, it will look at D equal-distanced points.

Details

This function is based on a regular bootstrap method, which calculates confidence bands for one quantile. It performs parallelization to speed up the calculation.

Value

A list with two parts: `x0` and `CIs`, where `x0` contains the `x` values that we are examining the confidence intervals at, and `CIs` is 2-dim matrix which contains the corresponding lower bound and upper bound.

References

Micheal, I. J et al.(2012). *A Scalable Bootstrap for Massive Data*.

See Also

[BLB_rqss](#) for BLB with 1-dim covariate data set.

[Bootstrap_rqss_2dim](#) for regular Bootstrap method with 2-dim covariate data set.

Examples

```
data(one)
result<-Bootstrap_rqss(data=one, alpha=0.05, tau=0.5, lambda=2, B=1000)

#exclude the points without predictions
valid<-!is.na(result$CIs[,1])

#Plot the confidence bands
plot(result$x0[valid],result$CIs[valid,1],type="l",col="blue",xlab="x",
      ylab="y",main="Confidence Band",ylim=c(2,6))
lines(result$x0[valid],result$CIs[valid,2],col="red")
```

Bootstrap_rqss_2dim	<i>Regular Bootstrap Method.</i>
---------------------	----------------------------------

Description

`Bootstrap_rqss_2dim` Regular bootstrap for data sets with 2-dim covariate. Used to generate the confidence bands for a quantile smoothing splines fitted with the `rqss` function from package [quantreg](#).

Usage

```
Bootstrap_rqss_2dim(cores = NULL, data = parent.frame(), alpha = 0.05,
  tau = 0.25, lambda = 2, B = 1500, D = 50, x0 = NULL)
```

Arguments

cores	The number of cores to use for parallel execution. If not specified, the number of cores is set to the value of <code>options("cores")</code> , if specified, or to one-half the number of cores detected by the parallel package.
data	A 3-dim optional data frame (or object coercible by as.data.frame to a data frame) containing the variables in the model. The column names should be specified in a way that "x1" and "x2" are for the two predictors respectively, and "y" for the response.
alpha	The confidence level required. The default is 0.05.
tau	A specific quantile to be estimated. Must be a number between 0 and 1.
lambda	The smoothing parameter used to specify the tradeoff between fidelity and roughness of the fitted quantile surface.
B	The number of Monte Carlo iterations using bootstrap with replacement. $B = 1500$ is by default.
D	A number that determines the density of a grid of x values over which pointwise confidence intervals will be computed. If specified, a DD grid of equidistant points on the plane is created.

Details

This function is based on a regular bootstrap method, which calculates confidence bands for one quantile. It performs parallelization to speed up the calculation.

Value

A list with two parts: `x0` and `CIs`, where `x0` contains the x values that we are examining the confidence intervals at, and `CIs` is 2-dim matrix which contains the corresponding lower bound and upper bound.

References

Micheal, I. J et al.(2012). *AScalableBootstrapforMassiveData*.
 Roger Koenker.(2005). *Quantileregression*. Cambridge university press.

See Also

[BLB_rqss_2dim](#) for BLB with 2-dim covariate data set.
[Bootstrap_rqss](#) for regular Bootstrap method with 1-dim covariate data set

Examples

```
data(two)
result<-Bootstrap_rqss_2dim(data=two, alpha=0.05, tau=0.5,lambda=2,B=1000)

#Exclude the points without predictions
valid<-!is.na(result$CIs[,1])

#Plot the confidence bands
library(akima)
X<-result$x0$x1[valid];Y<-result$x0$x2[valid]
lower<-result$CIs[valid,1]; upper<-result$CIs[valid,2]
```

```

x1range<-range(X);x2range<-range(Y)
akima1<-interp(X,Y,lower,xo=seq(x1range[1],x1range[2],length=200),yo=seq(x2range[1],x2range[2],length=200))
akima2<-interp(X,Y,upper,xo=seq(x1range[1],x1range[2],length=200),yo=seq(x2range[1],x2range[2],length=200))

#1. Contour plot
plot(X,Y,main="Prediction Contour Plot",type="n",xlab="x1",ylab="x2")
image(akima1,add=TRUE,col = terrain.colors(12)) ###YOU CAN CHANGE THE NUMBER OF COLOR LEVELS!!!!
contour(akima1,add=TRUE,nlevels=12)

#2. 3-D plot
library(plotly)
plot_ly(showscale = FALSE) %>%
  add_surface(x=akima1$x,y=akima1$y,z=akima1$z) %>%
  add_surface(x=akima2$x,y=akima2$y,z=akima2$z, opacity = 0.98)

```

CompExp

Mode 2: Comparative Experiment evaluated using confidence bands

Description

CompExp incorporates [lambdaSearch](#) and [CompExperiment_1dim](#) (or [CompExperiment_2dim](#)) together. The user inputs a data set, and the function does everything - choosing an optimum lambda first, and then compare the confidence bands to observe the treatment effect.

Usage

```

CompExp(cores = NULL, treatment = parent.frame(),
  control = parent.frame(), lambda_trt = NULL, lambda_ctr = NULL,
  method = "bootstrap", alpha = 0.05, tau = 0.5, B1 = 1000, B2 = 1000,
  D = NULL, col_trt = "green", col_ctr = "red", col_mid = "yellow",
  xlab = "Log(throughput[scaled])", ylab = "Log(playdelay[scaled])",
  x1 = "x1", x2 = "x2", nfolds = 10, init.lower = 0, init.upper = 15,
  resolution = 0.1, autosearch = FALSE, tol = 1e-05)

```

Arguments

cores	The number of cores to use for parallel execution. If not specified, the number of cores is set to the value of <code>options("cores")</code> (if specified) or to one-half the number of cores detected by the parallel package.
treatment	A 2-dim or 3-dim optional data frame for treatment group (or object coercible by as.data.frame to a data frame) containing the variables in the model. The last column of the data frame must be the response.
control	A 2-dim or 3-dim optional data frame for control group (or object coercible by as.data.frame to a data frame) containing the variables in the model. The last column of the data frame must be the response.
lambda_trt, lambda_ctr	Both can be a vector of real values, which are the candidate smoothing parameters for treatment and control respectively. If null, lambdaSearch will find optimum lambdas for the user.
method	Choose between "BLB" or "bootstrap". The default method is "bootstrap".

alpha	The confidence level required. The default is 0.05.
tau	A specific quantile to be estimated. Must be a number between 0 and 1.
B1	The number of Monte Carlo iterations for treatment when using bootstrap with replacement. $B = 1000$ is by default.
B2	The number of Monte Carlo iterations for control when using bootstrap with replacement. $B = 1000$ is by default.
D	Number of x-values at which the confidence bands will be estimated. If specified, pointwise confidence intervals will be estimated at D equal-distanced points.
xlab, ylab	Titles for x and y axis. See title .
x1, x2	Titles for x and y axis. See title .
nfolds	The number of folds for the cross validation.
init.lower	An initial lower bound used for lambda search; the default value is 0.
init.upper	An initial upper bound used for lambda search; the default value is 15.
resolution	Resolution used for plotting the multifold CV figure.
autosearch	if FALSE (which is recommended), then the function will show the MCV plot to the user who will be asked to enter an upper bound for conducting the search for λ . If TRUE, then the function will automatically choose the upper bound.
tol	The desired accuracy when calling optimize .

Details

This function can choose an optimum lambda automatically; also, the user can specify a sequence of lambda values, and the function can choose the best one out of these values.

Value

Please refer to [CompExperiment_1dim](#) (or [CompExperiment_2dim](#)).

References

Micheal, I. J et al.(2012). *A Scalable Bootstrap for Massive Data*.

See Also

[CompExperiment_1dim](#) for comparative experiment with 1-dim covariate data set using BLB.

[CompExperiment_1dim_boot](#) for comparative experiment with 1-dim covariate data set using regular bootstrap.

Examples

```
data(treatment)
data(control)
result<-CompExp(cores=6, treatment=treatment,control=control, tau=0.5)

#Or you can specify a vectors of lambdas
result<-CompExp(cores=6, treatment=treatment,control=control, tau=0.5,lambda_ctr=c(1,2,3))

data(treatment_2dim)
data(control_2dim)
result<-CompExp(cores=6, treatment=treatment_2dim,control=control_2dim, tau=0.5)
```

CompExperiment_1dim *Comparative Experiment.*

Description

CompExperiment_1dim compares the confidence bands at a given quantile for two different datasets, one related to a treatment and the other to a control. It applies the BLB algorithm to each dataset to get confidence bands using quantile smoothing splines for 1-dim covariate.

searchLambda is a wrapper function to calculate the optimum lambda.

CompPlot_1dim is a function to plot the results.

Usage

```
CompExperiment_1dim(cores = NULL, treatment, control, alpha = 0.05,
  tau = 0.25, lambda = 2, D = 100, b1 = ceiling(nrow(treatment)^0.6),
  b2 = ceiling(nrow(control)^0.6), s = 15, r = 100, Search = FALSE)
```

```
searchLambda(lambda, data, s, indices, tau, Rboot)
```

```
CompPlot_1dim(treatment, control, all, xlab = "x", ylab = "y",
  alpha = 0.05)
```

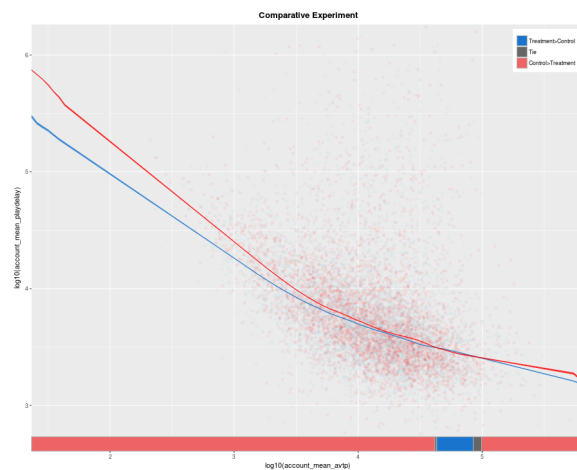
Arguments

cores	The number of cores to use for parallel execution. If not specified, the number of cores is set to the value of options("cores"), if specified, or to one-half the number of cores detected by the parallel package.
treatment	A 2-dim optional data frame for the treatment group (or object coercible by as.data.frame to a data frame) containing the variables in the model. The last column of this data frame must be the response for the experiment.
control	A 2-dim optional data frame for the control group (or object coercible by as.data.frame to a data frame) containing the variables in the model. The last column of this data frame must be the response for the experiment.
alpha	The confidence level required. The default is 0.05. In CompPlot_1dim this stands for how many percentage of data you want to throw away for the final plot.
tau	A specific quantile to be estimated. Must be a number between 0 and 1.
lambda	The smoothing parameter governing the tradeoff between fidelity and the penalty component for the triogram term. If Search=TRUE, there is no need for users to specify a value.
D	A number that specifies for how many x values you want to compute confidence intervals; the confidence band is made of pointwise intervals at various x values. If specified, it will look at D equidistant points.
b1	The subsample size in the BLB algorithm for treatment. Kleiner et al. suggest that the size should be around $n1^{0.6}$, where n1 is the data size for treatment.
b2	The subsample size in the BLB algorithm for control. It is also suggested that the size should be around $n2^{0.6}$, where n2 is the data size for control.

s	The number of subsamples used in the BLB algorithm. Kleiner et al. suggest that s should be 10~20.
r	The number of bootstrap iterations (samples with with replacement). $r = 100$ is suggested.
Search	If TRUE (which is recommended), then the function will first search for an optimum smoothing parameter λ .
xlab, ylab	Titles for x and y axis. See title .

Details

This function runs BLB twice, once for each dataset. It is based on [BLB_rqss](#), which implements BLB for quantile smoothing splines with a one dimensional covariate dataset. It performs parallelization to speed up the calculation.



CompPlot_1dim takes the results and use ggplot/plotly to visualize them, in which different colors represent different scenarios. See figure above.

Value

A list with three parts - result1, result2, and Diff, which respectively return confidence bands for treatment, control and the difference between the two dataset. Each part includes the following:

1. x0 and CI_average, where x0 contains the x values at which the confidence intervals are evaluated, and CI_average is 2-dim matrix which contains the corresponding lower and upper bounds.
2. lambda, which is the optimum smoothing parameter selected by BLB_rqss. If it is done automatically, the function also returns Lambda and Fid, which respectively stand for a vector of lambda values and their corresponding cross-validation MCV values.

References

Kleiner, I. J et al. JRSS B, 2012. *AScalableBootstrapforMassiveData*.

Akima, H. (1978). *AMethodofBivariateInterpolationandSmoothSurfaceFittingforIrregularlyDistributedData*. ACM Transactions on Mathematical Software 4, 148-164.

See Also

[contour](#), [image](#)

[BLB_rqss](#) for BLB with one dataset that has 1-dim covariate.

[CompExperiment_2dim](#) for comparative experiments with 2-dim covariate data sets.

Examples

```
data(treatment)
data(control)

#alpha=0.05;tau=0.5
all<-CompExperiment_1dim(cores=7, cores=2, treatment, control)

plot<-CompPlot_1dim(treatment = treatment, control=control, all = all, xlab = 'x', ylab = 'y')
```

CompExperiment_1dim_boot

Comparative Experiment.

Description

CompExperiment_1dim_boot compares the confidence bands at a given quantile for two different datasets, one related to a treatment and the other to a control. It applies the regular bootstrap algorithm to each dataset to get confidence bands using quantile smoothing splines for 1-dim covariate.

Usage

```
CompExperiment_1dim_boot(cores = NULL, treatment = parent.frame(),
  control = parent.frame(), alpha = 0.05, tau = 0.5, lambda_trt = 2,
  lambda_ctr = 2, D = 100, B1 = 1000, B2 = 1000, figure = TRUE,
  col_trt = "green", col_ctr = "red", col_mid = "yellow",
  xlab = "Log(throughput[scaled])", ylab = "Log(playdelay[scaled])",
  warning.catch = FALSE)
```

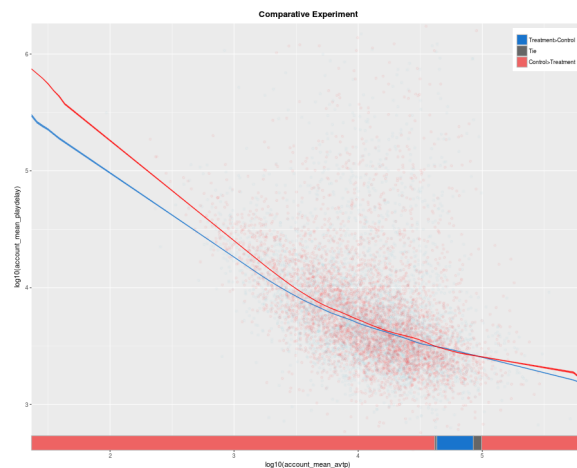
Arguments

cores	The number of cores to use for parallel execution. If not specified, the number of cores is set to the value of <code>options("cores")</code> , if specified, or to one-half the number of cores detected by the parallel package.
treatment	A 2-dim optional data frame for the treatment group (or object coercible by as.data.frame to a data frame) containing the variables in the model. The last column of this data frame must be the response for the experiment.
control	A 2-dim optional data frame for the control group (or object coercible by as.data.frame to a data frame) containing the variables in the model. The last column of this data frame must be the response for the experiment.
alpha	The confidence level required. The default is 0.05.
tau	A specific quantile to be estimated. Must be a number between 0 and 1.

<code>lambda_trt</code>	The smoothing parameter used for treatment, which governs the tradeoff between fidelity and the penalty component for the triogram term.
<code>lambda_ctr</code>	The smoothing parameter used for control, which governs the tradeoff between fidelity and the penalty component for the triogram term.
<code>D</code>	A number that specifies how many x values you want to examine the confidence bands at. If specified, we will look at D equal-distanced points.
<code>B1</code>	The number of Monte Carlo iterations for treatment when using bootstrap with replacement. $B = 1000$ is by default.
<code>B2</code>	The number of Monte Carlo iterations for control when using bootstrap with replacement. $B = 1000$ is by default.
<code>figure</code>	Logical value. If <code>figure = TRUE</code> , then a contour plot with different colors will be returned.
<code>col_trt</code>	Color code or name. The default is “green”. If <code>figure = TRUE</code> , then x -intervals over which this color appears mean that the confidence band for treatment is uniformly higher than the confidence band for control.
<code>col_ctr</code>	Color code or name. The default is “red”. x -intervals with this color indicate the confidence band for control is higher than the confidence band for treatment.
<code>col_mid</code>	Color code or name. The default is “yellow”. x -intervals with this color indicate that the confidence bands for control and treatment overlap.
<code>xlab, ylab</code>	Titles for x and y axis. See title .

Details

This function runs Bootstrap twice, once for each dataset. It is based on [Bootstrap_rqss](#), which implements regular bootstrap for quantile smoothing splines with one dimensional covariate dataset. It performs parallelization to speed up the calculation.



This function can also return a colored contour plot, in which different colors represent different scenarios. See figure above.

Value

A function that indicates you whether the treatment group performs better than the control group at a specific point. See examples. If the function returns 1, then it indicates better performance for treatment group, or vice versa for -1 .

References

Micheal, I. J et al.(2012). *A Scalable Bootstrap for Massive Data*.

Akima, H. (1978). *A Method of Bivariate Interpolation and Smooth Surface Fitting for Irregularly Distributed Data*.
ACM Transactions on Mathematical Software 4, 148-164.

See Also

[contour, image](#)

[Bootstrap_rqss](#) for simple bootstrap with one dataset that has 1-dim covariate.

[CompExperiment_2dim_boot](#) for comparative experiments with 2-dim covariate data sets.

Examples

```
data(treatment)
data(control)

#alpha=0.05;tau=0.5;lambda_trt=2;lambda_ctr=2
result<-CompExperiment_1dim_boot(cores=2,treatment,control)

#You can also change the colors for each section by:
result<-CompExperiment_1dim_boot(cores=2,treatment,control,col_trt="red",col_ctr="green",col_mid="yellow")

#You can also change the x and y labels!

curve(result,from=-3,to=3)
```

CompExperiment_1dim_D *Comparative Experiment (Discretized)*.

Description

CompExperiment_1dim_D compares the confidence bands at a given quantile for two different datasets, one related to a treatment and the other to a control. It applies the BLB algorithm to each dataset to get confidence bands using quantile smoothing splines for 1-dim covariate. What's special about this function is that it discretizes the data set to decrease the sample size, and it utilizes ALL the observations.

searchLambda_1dim_D is a wrapper function to calculate the optimum lambda.

Usage

```
CompExperiment_1dim_D(cores = NULL, treatment, control, alpha = 0.05,
  tau = 0.25, lambda = 2, D = 100, s = 15,
  b1 = floor(nrow(treatment)/s), b2 = floor(nrow(control)/s), r = 100,
  M = 80, Range = c(0.001, 0.999), Search = FALSE)

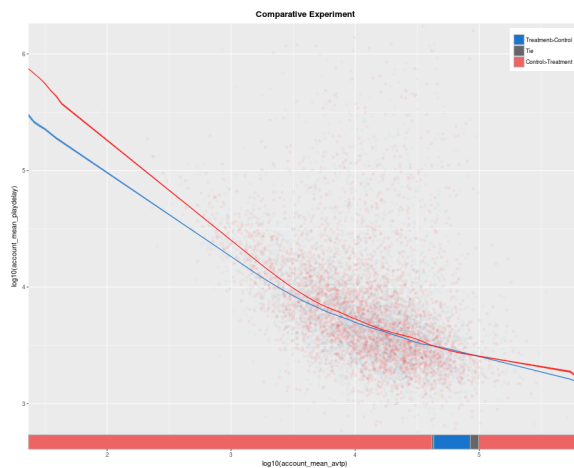
searchLambda_1dim_D(lambda, Tmp, s, indices, tau, Rboot)
```

Arguments

cores	The number of cores to use for parallel execution. If not specified, the number of cores is set to the value of <code>options("cores")</code> , if specified, or to one-half the number of cores detected by the parallel package.
treatment	A 2-dim optional data frame for the treatment group (or object coercible by as.data.frame to a data frame) containing the variables in the model. The last column of this data frame must be the response for the experiment.
control	A 2-dim optional data frame for the control group (or object coercible by as.data.frame to a data frame) containing the variables in the model. The last column of this data frame must be the response for the experiment.
alpha	The confidence level required. The default is 0.05.
tau	A specific quantile to be estimated. Must be a number between 0 and 1.
lambda	The smoothing parameter used for treatment & control if <code>Search = FALSE</code> , which governs the tradeoff between fidelity and the penalty component for the triogram term.
D	A number that specifies for how many x values you want to compute confidence intervals; the confidence band is made of pointwise intervals at various x values. If specified, it will look at D equidistant points.
s	The number of subsamples used in the BLB algorithm. Kleiner et al. suggest that s should be 10~20.
b1	The subsample size in the BLB algorithm for treatment. Kleiner et al. suggest that the size should be around $n1^{0.6}$, where $n1$ is the data size for treatment.
b2	The subsample size in the BLB algorithm for control. It is also suggested that the size should be around $n2^{0.6}$, where $n2$ is the data size for control.
r	The number of bootstrap iterations (samples with with replacement). $r = 100$ is suggested.
M	A numeric value that controls how fine that data set should be discretized.
Range	A vector with 2 values that specifies the range of the data set where the user wants to perform BLB over. It is defined using lower and upper quantile. The default value is (0.001, 0.999).
Search	If TRUE (which is recommended), then the function will first search for an optimum smoothing parameter λ .

Details

This function runs BLB twice, once for each dataset. It is based on [BLB_Discretize](#), which implements BLB for quantile smoothing splines with a one dimensional covariate dataset. It performs parallelization to speed up the calculation.



`CompPlot_1dim` takes the results and use ggplot/plotly to visualize them, in which different colors represent different scenarios. See figure above.

Value

A list with three parts - `result1`, `result2`, and `Diff`, which respectively return confidence bands for treatment, control and the difference between the two dataset. Each part includes the following:

1. `x0` and `CI_average`, where `x0` contains the x values at which the confidence intervals are evaluated, and `CI_average` is 2-dim matrix which contains the corresponding lower and upper bounds.
2. `lambda`, which is the optimum smoothing parameter selected by `BLB_Discretize`. If it is done automatically, the function also returns `Lambda` and `Fid`, which respectively stand for a vector of lambda values and their corresponding cross-validation MCV values.

References

Kleiner, I. J et al. JRSS B, 2012. *AScalableBootstrapforMassiveData*.

Akima, H. (1978). *AMethodofBivariateInterpolationandSmoothSurfaceFittingforIrregularlyDistributedL*
ACM Transactions on Mathematical Software 4, 148-164.

See Also

[contour, image](#)

[BLB_rqss](#) for BLB with one dataset that has 1-dim covariate.

[CompExperiment_2dim](#) for comparative experiments with 2-dim covariate data sets.

Examples

```
data(treatment)
data(control)

#alpha=0.05;tau=0.5
all<-CompExperiment_1dim_D(cores=7,treatment,control)

plot<-CompPlot_1dim(treatment = treatment, control=control, all = all, xlab = 'x', ylab = 'y')
```

CompExperiment_2dim *Comparative Experiment.*

Description

CompExperiment_2dim compares the confidence regions for a given quantile for two different datasets, one related to a treatment and the other to a control. It applies the BLB algorithm to each dataset to get confidence regions using quantile smoothing splines for 2-dim covariate.

searchLambda_2dim is a wrapper function to calculate the optimum lambda.

CompPlot_2dim is a function to plot the results.

Usage

```
CompExperiment_2dim(cores = NULL, treatment, control, alpha = 0.05,
  tau = 0.25, lambda = 2, D = 50, b1 = ceiling(nrow(treatment)^0.6),
  b2 = ceiling(nrow(control)^0.6), s = 15, r = 100, Search = FALSE)
```

```
searchLambda_2dim(lambda, data, s, indices, tau, n, b)
```

```
CompPlot_2dim(control, treatment, all, Range = c(0.002, 0.998),
  xlab = "log(avtp_med)", ylab = "log(niqr_med)",
  zlab = "log(playdelay_med)")
```

Arguments

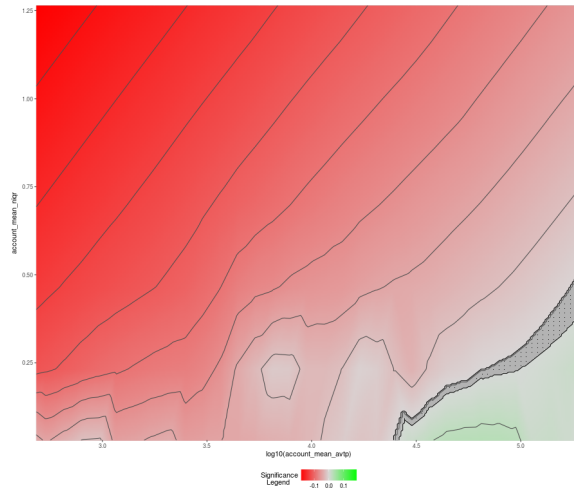
cores	The number of cores to use for parallel execution. If not specified, the number of cores is set to the value of options("cores"), if specified, or to one-half the number of cores detected by the parallel package.
treatment	A 3-dim optional data frame for the treatment group (or object coercible by as.data.frame to a data frame) containing the variables in the model. The last column of this data frame must be the response for the experiment.
control	A 3-dim optional data frame for the control group (or object coercible by as.data.frame to a data frame) containing the variables in the model. The last column of this data frame must be the response for the experiment.
alpha	The confidence level required. The default is 0.05.
tau	A specific quantile to be estimated. Must be a number between 0 and 1.
lambda	The smoothing parameter used for treatment & control if <i>Search = FALSE</i> , which governs the tradeoff between fidelity and the penalty component for the triogram term.
D	A number that determines the density of a grid of x values at which the quantile function will be predicted. If specified, it will evaluate a confidence surface on a <i>DD</i> grid.
b1	The subsample size in the BLB algorithm for treatment. Kleiner et al. suggest that the size should be around $n1^{0.6}$, where $n1$ is the data size for treatment.
b2	The subsample size in the BLB algorithm for control. It is also suggested that the size should be around $n2^{0.6}$, where $n2$ is the data size for control.
s	The number of subsamples used in the BLB algorithm. Kleiner et al. suggest that <i>s</i> should be 10~20.

r The number of bootstrap iterations (samples with with replacement). $r = 100$ is suggested.

xlab, ylab, zlab Titles for the axes. See [title](#).

Details

This function runs BLB twice, once for each dataset. It is based on [BLB_rqss_2dim](#), which implements BLB for quantile smoothing splines with a two-dimensional covariate dataset. It performs parallelization to speed up the calculation.



CompPlot_2dim takes the results and use ggplot/plotly to visualize them, in which different colors represent different scenarios. See figure above.

Value

A list with three parts - result1, result2, and Diff, which respectively return confidence bands for treatment, control and the difference between the two dataset. Each part includes the following:

1. x0 and CI_average, where x0 contains the x values at which the confidence intervals are evaluated, and CI_average is 2-dim matrix which contains the corresponding lower and upper bounds.
2. lambda, which is the optimum smoothing parameter selected by BLB_rqss_2dim. If it is done automatically, the function also returns Lambda and Fid, which respectively stand for a vector of lambda values and their corresponding cross-validation MCV values.

References

Kleiner, I. J et al. JRSS B, 2012. *AScalableBootstrapforMassiveData*.

Akima, H. (1978). *AMethodofBivariateInterpolationandSmoothSurfaceFittingforIrregularlyDistributedData*. ACM Transactions on Mathematical Software 4, 148-164.

See Also

[contour](#), [image](#)

[BLB_rqss_2dim](#) for BLB with one dataset that has 1-dim covariate.

[CompExperiment_1dim](#) for comparative experiments with 1-dim covariate data sets.

Examples

```
data(treatment_2dim)
data(control_2dim)

#alpha=0.05;tau=0.5
all<-CompExperiment_2dim(cores=7, treatment_2dim, control_2dim, Search=TRUE)

plot<-CompPlot_2dim(control = control,treatment = treatment,all = all,xlab="x1",
ylab="x2",zlab="z")
```

CompExperiment_2dim_boot

Comparative Experiment.

Description

CompExperiment_2dim_boot compares the confidence regions for a given quantile for two different datasets, one related to a treatment and the other to a control. It applies the regular bootstrap algorithm to each dataset to get confidence regions using quantile smoothing splines for 2-dim covariate.

Usage

```
CompExperiment_2dim_boot(cores = NULL, treatment = parent.frame(),
  control = parent.frame(), alpha = 0.05, tau = 0.5, lambda_trt = 2,
  lambda_ctr = 2, B1 = 1000, B2 = 1000, D = 50, figure = TRUE,
  col_trt = "green", col_ctr = "red", col_mid = "yellow", x1 = "x1",
  x2 = "x2")
```

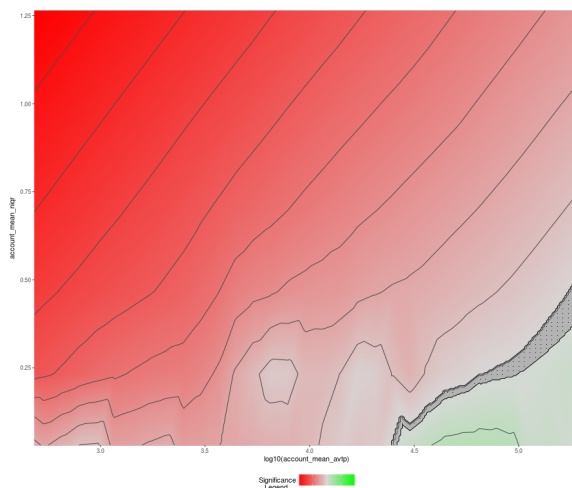
Arguments

cores	The number of cores to use for parallel execution. If not specified, the number of cores is set to the value of options("cores"), if specified, or to one-half the number of cores detected by the parallel package.
treatment	A 3-dim optional data frame for the treatment group (or object coercible by as.data.frame to a data frame) containing the variables in the model. The last column of this data frame must be the response for the experiment.
control	A 3-dim optional data frame for the control group (or object coercible by as.data.frame to a data frame) containing the variables in the model. The last column of this data frame must be the response for the experiment.
alpha	The confidence level required. The default is 0.05.
tau	A specific quantile to be estimated. Must be a number between 0 and 1.
lambda_trt	The smoothing parameter used for treatment, which governs the tradeoff between fidelity and the penalty component for the triogram term.
lambda_ctr	The smoothing parameter used for control, which governs the tradeoff between fidelity and the penalty component for the triogram term.
B1	The number of Monte Carlo iterations for treatment when using bootstrap with replacement. $B = 1000$ is by default.

B2	The number of Monte Carlo iterations for control when using bootstrap with replacement. $B = 1000$ is by default.
D	A number that determines the density of your grid of x values that you want to predict on. If specified, we will examine the confidence bands on a DD grid.
figure	Logical value. If <i>figure</i> = <i>TURE</i> , then a contour plot with different colors will be returned.
col_trt	Color code or name. The default is “green”. If <i>figure</i> = <i>TURE</i> , then in the section with this color, the confidence bands for treatment will be uniformly larger than the confidence bands for control.
col_ctr	Color code or name. The default is “red”. In the section with this color, the confidence bands for control will be uniformly larger than the confidence bands for treatment.
col_mid	Color code or name. The default is “yellow”. In the section with this color, the confidence bands for both datasets are inter-wined with each other.
x1, x2	Titles for x and y axis. See title .

Details

This function runs Bootstrap twice, once for each dataset. It is based on [Bootstrap_rqss_2dim](#), which implements regular bootstrap for quantile smoothing splines with two dimensional covariate dataset. It performs parallelization to speed up the calculation.



This function can also return a colored contour plot, in which different colors represent different scenarios. See figure above.

Value

List with 3 components:

x, *y* vectors of x- and y- coordinates of output grid, which are evenly spaced over the range of the input data. They are used to evaluate the difference between the confidence bands for treatment and control.

z matrix with values NA, 1, 0, -1. The value *z*[*i*,*j*] is computed at the *x*, *y* point *x*[*i*], *y*[*j*]. *z* has dimensions length(*x*) times length(*y*). 1 means treatment has larger confidence bands. -1 means control has larger confidence bands. 0 means the two confidence bands are entangled with each other. NA means this point is out of the convex hull of the data input.

References

Micheal, I. J et al.(2012). *A Scalable Bootstrap for Massive Data*.

Akima, H. (1978). *A Method of Bivariate Interpolation and Smooth Surface Fitting for Irregularly Distributed Data*.
ACM Transactions on Mathematical Software 4, 148-164.

See Also

[contour, image](#)

[Bootstrap_rqss_2dim](#) for simple bootstrap method with one dataset that has 1-dim covariate.

[CompExperiment_1dim_boot](#) for comparative experiments with 1-dim covariate data sets.

Examples

```
data(treatment_2dim)
data(control_2dim)

#alpha=0.05;tau=0.5;lambda_trt=2;lambda_ctr=2
result<-CompExperiment_2dim_boot(cores=2,treatment_2dim,control_2dim,figure=TRUE)

#You can also change the colors for each section by:
result<-CompExperiment_2dim_boot(cores=2,treatment_2dim,control_2dim,col_trt="red",col_ctr="green",col_mid="black")

#You can also change the x and y labels!
```

CompExperiment_2dim_D *Comparative Experiment (Discretized)*.

Description

CompExperiment_2dim_D compares the confidence regions for a given quantile for two different datasets, one related to a treatment and the other to a control. It applies the BLB algorithm to each dataset to get confidence regions using quantile smoothing splines for 2-dim covariate. What's special about this function is that it discretizes the data set to decrease the sample size, and it utilizes ALL the observations.

searchLambda_2dim_D is a wrapper function to calculate the optimum lambda.

Usage

```
CompExperiment_2dim_D(cores = NULL, treatment, control, alpha = 0.05,
  tau = 0.25, lambda = 2, D = 50, s = 15,
  b1 = floor(nrow(treatment)/s), b2 = floor(nrow(control)/s), r = 100,
  M = 30, Range = c(0.001, 0.999), Search = FALSE)

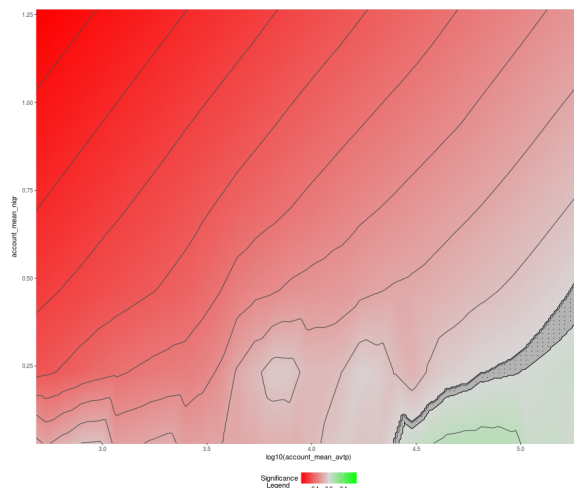
searchLambda_2dim_D(lambda, Tmp, s, indices, tau, n, b)
```

Arguments

cores	The number of cores to use for parallel execution. If not specified, the number of cores is set to the value of <code>options("cores")</code> , if specified, or to one-half the number of cores detected by the <code>parallel</code> package.
treatment	A 3-dim optional data frame for the treatment group (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. The last column of this data frame must be the response for the experiment.
control	A 3-dim optional data frame for the control group (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. The last column of this data frame must be the response for the experiment.
alpha	The confidence level required. The default is 0.05.
tau	A specific quantile to be estimated. Must be a number between 0 and 1.
lambda	The smoothing parameter used for treatment & control if <code>Search = FALSE</code> , which governs the tradeoff between fidelity and the penalty component for the triogram term.
D	A number that determines the density of a grid of x values at which the quantile function will be predicted. If specified, it will evaluate a confidence surface on a DD grid.
s	The number of subsamples used in the BLB algorithm. Kleiner et al. suggest that s should be 10~20.
b1	The subsample size in the BLB algorithm for treatment. Kleiner et al. suggest that the size should be around $n1^{0.6}$, where $n1$ is the data size for treatment.
b2	The subsample size in the BLB algorithm for control. It is also suggested that the size should be around $n2^{0.6}$, where $n2$ is the data size for control.
r	The number of bootstrap iterations (samples with with replacement). $r = 100$ is suggested.
M	A numeric value that controls how fine that data set should be discretized.
Range	A vector with 2 values that specifys the range of the data set where the user wants to perform BLB over. It is defined using lower and upper quantile. The default value is (0.001, 0.999).
Search	If TRUE (which is recommended), then the function will first search for an optimum smoothing parameter λ .

Details

This function runs BLB twice, once for each dataset. It is based on `BLB_Discretize_2dim`, which implements BLB for quantile smoothing splines with a two-dimensional covariate dataset. It performs parallelization to speed up the calculation.



[CompPlot_2dim](#) takes the results and use ggplot/plotly to visualize them, in which different colors represent different scenarios. See figure above.

Value

A list with three parts - result1, result2, and Diff, which respectively return confidence bands for treatment, control and the difference between the two dataset. Each part includes the following:

1. x0 and CI_average, where x0 contains the x values at which the confidence intervals are evaluated, and CI_average is 2-dim matrix which contains the corresponding lower and upper bounds.
2. lambda, which is the optimum smoothing parameter selected by BLB_Discretize_2dim. If it is done automatically, the function also returns Lambda and Fid, which respectively stand for a vector of lambda values and their corresponding cross-validation MCV values.

References

Kleiner, I. J et al. JRSS B, 2012. *A Scalable Bootstrap for Massive Data*.

Akima, H. (1978). *A Method of Bivariate Interpolation and Smooth Surface Fitting for Irregularly Distributed Data*. ACM Transactions on Mathematical Software 4, 148-164.

See Also

[contour](#), [image](#)

[BLB_Discretize_2dim](#) for BLB with one dataset that has 1-dim covariate.

[CompExperiment_1dim_D](#) for comparative experiments with 1-dim covariate data sets.

Examples

```
data(treatment_2dim)
data(control_2dim)

#alpha=0.05;tau=0.5
all<-CompExperiment_2dim_D(cores=72, treatment_2dim, control_2dim, Search=TRUE)

plot<-CompPlot_2dim(control = control, treatment = treatment, all = all, xlab="x1",
ylab="x2", zlab="z")
```

control	<i>One-covariate control dataset for comparative experiment</i>
---------	---

Description

A dataset containing the covariate 'x' and the response 'y' for 3,200 observations.

Usage

```
control
```

Format

A data frame with 3200 rows and 2 variables:

x covariate

y response...

control_2dim	<i>Two-covariate treatment dataset for comparative experiment</i>
--------------	---

Description

A dataset containing the covariates 'x1', 'x2', and the response 'y' for 3,200 observations.

Usage

```
control_2dim
```

Format

A data frame with 3200 rows and 3 variables:

x1 covariate

x2 covariate

y response...

foldsGenerator	<i>Fold Generator for Cross-Validation.</i>
----------------	---

Description

foldsGenerator is used to generate folds for a specific data set when performing crossvalidation.

Usage

```
foldsGenerator(sim.data, nfolds = 10, n = NULL)
```

Arguments

sim.data	A data frame (or object coercible by <code>as.data.frame</code> to a data frame) on which the cross validation is performed.
nfolds	The number of folds for the cross validation.
n	The number of rows if the data frame not provided.

Details

This function first identifies the number of observations in the data set. With the pre-specified number of folds `nfolds`, the function generates a sequence of randoms numbers which vary from 1 to `nfolds`. Each number indicates which fold an observation belongs to.

Value

A vector of random numbers which vary from 1 to `nfolds`.

Examples

```
data(one)

folds<-foldsGenerator(one,nfolds=10)
```

lambdaSearch	<i>Smoothness Selection for Quantile Smoothing Splines.</i>
--------------	---

Description

lambdaSearch finds the best smoothing parameter λ by minimizing the multifold crossvalidation (MCV) criterion for a given dataset

Usage

```
lambdaSearch(cores = NULL, data = parent.frame(), tau = 0.5,
  nfolds = 10, init.lower = 0, init.upper = 15, resolution = 0.1,
  autosearch = FALSE, tol = 1e-05, INFO = FALSE)
```

Arguments

cores	The number of cores to use for parallel execution. If not specified, the number of cores is set to the value of <code>options("cores")</code> , if specified, or to one-half the number of cores detected by the parallel package.
data	A 2- or 3-dim data frame (or object coercible by as.data.frame to a data frame) for which we are selecting the smoothing parameter.
tau	A specific quantile to be estimated. Must be a number between 0 and 1.
nfolds	The number of folds for the cross validation.
init.lower	An initial lower bound used for lambda search; the default value is 0.
init.upper	An initial upper bound used for lambda search; the default value is 15.
resolution	Resolution used for plotting the MCV figure.
autosearch	if FALSE (which is recommended), then the function will show the MCV plot to the user who will be asked to enter an upper bound for conducting the search for λ . If TRUE, then the function will automatically choose the upper bound.
tol	The desired accuracy tolerance for optimize .

Details

If `autosearch=FALSE`, this function first roughly draws a plot of MCV vs. λ values, after which an initial upper bound for finding the optimum λ can be specified by the user. (This was implemented due to the numerical issues caused by [rqss](#). Those issues can make the curve oscillate dramatically, which would make [optimize](#) fail to find a valid minimum.)

If `autosearch=TRUE`, the function chooses automatically a range for [optimize](#).

Value

After some time, the optimum λ is returned.

References

Reiss et al. (2012). *SmoothnessSelectionforpenalizedQuantileRegressionSplines*.

See Also

[foldsGenerator](#), [multifoldCV](#), [multifoldCV_2dim](#)

Examples

```
data(one)    #1 covariate

##1. The function can find lambda automatically
lambda<-lambdaSearch(cores=3,data=one,tau=0.5,nfolds=10,init.upper=15,resolution=0.1,autosearch=TRUE)
lambda

##2. Or you can look at the plot and then decide an upper bound for lambda search
lambda<-lambdaSearch(cores=3,data=one,tau=0.5,nfolds=10,init.upper=15,resolution=0.1,autosearch=FALSE)

data(two)    #2 covariates
lambda<-lambdaSearch(cores=3,data=two,tau=0.5,nfolds=10,init.upper=15,resolution=0.1,autosearch=FALSE)
```

multifoldCV

*MCV Criterion for the Selection of λ .***Description**

multifoldCV calculates the multi-fold crossvalidation (MCV) criterion for choosing the smoothness parameter in quantile smoothing splines. The MCV criterion was proposed by Reiss et al. in 2012. Reiss et al. and our own simulations demonstrate this criterion is suitable as a reliable automatic procedure to choose the optimal smoothing parameter.

multifoldCV_2dim calculates the same criterion for 2-dim covariate data sets.

Usage

```
multifoldCV(cores = NULL, lambda, tau, sim.data, cv.idx)
```

```
multifoldCV_2dim(cores = NULL, lambda, tau, sim.data, cv.idx)
```

Arguments

cores	The number of cores to use for parallel execution. If not specified, the number of cores is set to the value of <code>options("cores")</code> , if specified, or to one-half the number of cores detected by the parallel package.
lambda	The smoothing parameter governing the tradeoff between fidelity and the penalty component for the triogram term.
tau	A specific quantile to be estimated. Must be a number between 0 and 1.
sim.data	A data frame (or object coercible by as.data.frame to a data frame) for which we are selecting the smoothing parameter.
cv.idx	The folds (the divided data) randomly generated by foldsGenerator .

Details

In general, the optimal degree of smoothness will depend on the quantile of interest. Given a value $\tau \in [0, 1]$, nonparametric quantile regression seeks to estimate the conditional quantile function $g(x)$ minimizing:

$$\sum_{i=1}^n \rho_{\tau}[y_i - g(x_i)] + \lambda J(g)$$

where $J(g)$ is the total variation of $g(x)$. Multifold CV (Reiss, 2012) can greatly reduce the computational burden. Here we divide the n observations into V_1, \dots, V_k of (approximately) equal size, and define the criterion:

$$MCV = \frac{1}{n} \sum_{j=1}^k \sum_{i \in V_j} \rho_{\tau}[y_i - \hat{g}_{\lambda}^{[-V_j]}(x_i)]$$

where $\hat{g}_{\lambda}^{[-V_j]}$ is the function estimate based on all observations except those in fold V_j . Small values of k produce downward-biased estimates of prediction error, whereas larger values produce more variable results and impose a higher computational burden; $k = 5$ or 10 is often recommended as a compromise.

Value

The MCV criterion is returned.

References

Reiss et al. (2012). *Smoothness Selection for penalized Quantile Regression Splines*.

See Also

[foldsGenerator](#)

Examples

```
data(one)
folds<-foldsGenerator(sim.data=one,nfolds=10)
multifoldCV(cores=3,lambda=2,tau=0.5,sim.data=one,cv.idx=folds)

data(two)
folds<-foldsGenerator(sim.data=two,nfolds=10)
multifoldCV_2dim(cores=3,lambda=2,tau=0.5,sim.data=two,cv.idx=folds)
```

one

One-covariate dataset for Testing

Description

A dataset containing the covariate 'x' and the response 'y' for almost 3,000 observations.

Usage

one

Format

A data frame with 2999 rows and 2 variables:

x covariate

y response ...

Description

`rqss_new` is based on `rqss` function in `quantreg` package. The main difference is that this new function implements the `weights` option, which can be quite useful. It is used to fit function for additive quantile regression models with possible univariate and/or bivariate nonparametric terms estimated by total variation regularization.

Usage

```
rqss_new(formula, tau = 0.5, data = parent.frame(), weights, na.action,
  method = "sfnc", lambda = NULL, contrasts = NULL, ztol = 1e-05,
  control = sfnc.control(), ...)
```

Arguments

<code>formula</code>	a formula object, with the response on the left of a '~' operator, and terms, separated by '+' operators, on the right. The terms may include qss terms that represent additive nonparametric components. These terms can be univariate or bivariate. See <code>qss</code> for details on how to specify these terms.
<code>tau</code>	the quantile to be estimated, this must be a number between 0 and 1.
<code>data</code>	a data.frame in which to interpret the variables named in the formula, or in the subset and the weights argument.
<code>weights</code>	vector of observation weights. The length of weights must be the same as the number of observations. The weights must be positive integers which basically indicates the number of repeats.
<code>na.action</code>	a function to filter missing data. This is applied to the model.frame after any subset argument has been used. The default (with <code>na.fail</code>) is to create an error if any missing values are found. A possible alternative is <code>na.omit</code> , which deletes observations that contain one or more missing values.
<code>method</code>	<p>the algorithmic method used to compute the fit. There are currently two options. Both are implementations of the Frisch–Newton interior point method described in detail in Portnoy and Koenker(1997). Both are implemented using sparse Cholesky decomposition as described in Koenker and Ng (2003).</p> <p>Option "sfnc" is used if the user specifies inequality constraints. Option "sfnc" is used if there are no inequality constraints. Linear inequality constraints on the fitted coefficients are specified by a matrix R and a vector r, specified inside the qss terms, representing the constraints in the form $Rb \geq r$.</p> <p>The option <code>method = "lasso"</code> allows one to penalize the coefficients of the covariates that have been entered linearly as in <code>rq.fit.lasso</code>; when this is specified then there should be an additional <code>lambda</code> argument specified that determines the amount of shrinkage.</p>
<code>lambda</code>	can be either a scalar, in which case all the slope coefficients are assigned this value, or alternatively, the user can specify a vector of length equal to the number of linear covariates plus one (for the intercept) and these values will be used as coordinate dependent shrinkage factors.

contrasts	a list giving contrasts for some or all of the factors default = NULL appearing in the model formula. The elements of the list should have the same name as the variable and should be either a contrast matrix (specifically, any full-rank matrix with as many rows as there are levels in the factor), or else a function to compute such a matrix given the number of levels.
ztol	A zero tolerance parameter used to determine the number of zero residuals in the fitted object which in turn determines the effective dimensionality of the fit.
control	control argument for the fitting routines (see sfn.control)
...	Other arguments passed to fitting routines

Details

Total variation regularization for univariate and bivariate nonparametric quantile smoothing is described in Koenker, Ng and Portnoy (1994) and Koenker and Mizera(2003) respectively. The additive model extension of this approach depends crucially on the sparse linear algebra implementation for R described in Koenker and Ng (2003). There are extractor methods [logLik](#) and [AIC](#) that is relevant to lambda selection. A more detailed description of some recent developments of these methods is available from within the package with `vignette("rqss")`. Since this function uses sparse versions of the interior point algorithm it may also prove to be useful for fitting linear models without [qss](#) terms when the design has a sparse structure, as for example when there is a complicated factor structure.

If the **MatrixModels** and **Matrix** packages are both loadable then the linear in parameters portion of the design matrix is made in sparse matrix form, this is helpful in large applications with many factor variables for which dense formation of the design matrix would take too much space.

Value

The function returns a fitted object representing the estimated model specified in the formula. See [rqss.object](#) for further details on this object, and references to methods to look at it.

Author(s)

Roger Koenker, Modified by Likun Zhang

References

- Koenker, R. and S. Portnoy (1997) The Gaussian Hare and the Laplacean Tortoise: Computability of Squared-error vs Absolute Error Estimators, (with discussion). *Statistical Science* **12**, 279–300.
- Koenker, R., P. Ng and S. Portnoy, (1994) Quantile Smoothing Splines; *Biometrika* **81**, 673–680.
- Koenker, R. and I. Mizera, (2003) Penalized Triograms: Total Variation Regularization for Bivariate Smoothing; *JRSS(B)* **66**, 145–163.
- Koenker, R. and P. Ng (2003) SparseM: A Sparse Linear Algebra Package for R, *J. Stat. Software*.
- Esmond G. Ng and Barry W. Peyton, "Block sparse Cholesky algorithms on advanced uniprocessor computers". *SIAM J. Sci. Stat. Comput.* 14 (1993), pp. 1034-1056.
- John R. Gilbert, Esmond G. Ng, and Barry W. Peyton, "An efficient algorithm to compute row and column counts for sparse Cholesky factorization". *SIAM J. Matrix Anal. Appl.* 15 (1994), pp. 1075-1091.

See Also

[rqss](#), [qss](#)

Examples

```
One<-one[sample(1:2999,2000),]
tau<-0.5
fit0<-rqss(y ~ qss(x, lambda = 1), tau = tau, data = One)

x0<-seq(2.3,5.1,length=200)
response<-rep(NA,200)
extrapolated <- x0 > max(One$x) | x0 < min(One$x)
response[!extrapolated] = predict(fit0, newdata = data.frame(x=x0[!extrapolated]))

plot(One,col="grey",pch=20,main="Implement Weights in rqss()")
lines(x0,response,lty=2,lwd=3)

times<-sample(1:30,2000,replace=TRUE)
index<-rep(1:2000,times=times)
One_new<-One[index,]
fit_rep<-rqss(y ~ qss(x, lambda = 1), tau = tau, data = One_new)

response1<-rep(NA,200)
extrapolated <- x0 > max(One_new$x) | x0 < min(One_new$x)
response1[!extrapolated] = predict(fit_rep, newdata = data.frame(x=x0[!extrapolated]))

lines(x0,response1,col="red",lwd=2)
legend("topright",lty=c(2,1),lwd=c(3,2),col=c("black","red"),legend=c("Original Data","Repeated Data"))
```

SurfaceFit

Mode 1: Surface fit with confidence bands.

Description

SurfaceFit incorporates [lambdaSearch](#) and [BLB_rqss](#) (or [Bootstrap_rqss](#)) together. After entering a data set, this function finds an optimum lambda (smoothing constant) first, and then obtains a confidence band.

Usage

```
SurfaceFit(cores = NULL, data = parent.frame(), lambda = NULL,
  method = "bootstrap", alpha = 0.05, tau = 0.5, B = 1000, D = NULL,
  nfolds = 10, init.lower = 0, init.upper = 15, resolution = 0.1,
  autosearch = FALSE, tol = 1e-05)
```

Arguments

cores	The number of cores to use for parallel execution. If not specified, the number of cores is set to the value of <code>options("cores")</code> (if this is specified) or to one-half the number of cores detected by the parallel package.
data	A 2-dim or 3-dim optional data frame (or object coercible by as.data.frame to a data frame) containing the variables in the model. The last column of the data frame must be the response.

<code>lambda</code>	A vector of real values, which are the candidate smoothing parameters governing the tradeoff between fidelity and the penalty component for the triogram term. If null, lambdaSearch will find an optimum lambda for the user.
<code>method</code>	Choose between "BLB" or "bootstrap". The default method is "bootstrap".
<code>alpha</code>	The confidence level required. The default is 0.05.
<code>tau</code>	A specific quantile to be estimated. Must be a number between 0 and 1.
<code>B</code>	If <code>method=='bootstrap'</code> , the number of Monte Carlo iterations using bootstrap with replacement. $B = 1000$ is by default.
<code>D</code>	Number of x-values at which the confidence bands will be estimated. If specified, pointwise confidence intervals will be estimated at D equal-distanced points.
<code>nfolds</code>	The number of folds for the cross validation.
<code>init.lower</code>	An initial lower bound used for lambda search; the default value is 0.
<code>init.upper</code>	An initial upper bound used for lambda search; the default value is 15.
<code>resolution</code>	Resolution used for plotting the multifold CV figure.
<code>autosearch</code>	if FALSE (which is recommended), then the function will show the MCV plot to the user who will be asked to enter an upper bound for conducting the search for λ . If TRUE, then the function will automatically choose the upper bound.
<code>tol</code>	The desired accuracy when calling optimize .

Details

This function can choose an optimum lambda automatically; alternatively, the user can specify a sequence of lambda values, and the function can choose the best one out of these values.

Value

A list with two parts: `x0` and `CIs`, where `x0` contains the x values at which the pointwise confidence intervals were computed, and `CIs`, a 2-dim matrix that contains the corresponding lower and upper bounds for each interval.

References

Micheal, I. J et al.(2012). *AScalableBootstrapforMassiveData*.

See Also

[BLB_rqss](#) for BLB with 1-dim covariate data set.

[Bootstrap_rqss](#) for regular Bootstrap method with 1-dim covariate data set.

Examples

```
data(one)
result<-SurfaceFit(data=one, alpha=0.05, tau=0.5, method='bootstrap', B=1000)

#Or you can specify a vectors of lambdas
result<-SurfaceFit(data=one, alpha=0.05, tau=0.5, method='bootstrap', B=1000, lambda=c(1,2,3))

data(two)
result<-SurfaceFit(data=two, alpha=0.05, tau=0.5, method='bootstrap', B=1000)
result$plot
```

treatment	<i>One-covariate treatment dataset for comparative experiment</i>
-----------	---

Description

A dataset containing the covariate 'x' and the response 'y' for 3,200 observations.

Usage

```
treatment
```

Format

A data frame with 3200 rows and 2 variables:

x covariate

y response...

treatment_2dim	<i>Two-covariate treatment dataset for comparative experiment</i>
----------------	---

Description

A dataset containing the covariates 'x1', 'x2', and the response 'y' for 3,200 observations.

Usage

```
treatment_2dim
```

Format

A data frame with 3200 rows and 3 variables:

x1 covariate

x2 covariate

y response...

`two`*Two-covariate dataset for Testing*

Description

A dataset containing the covariates 'x1', "x2" and the response 'y' for almost 3,000 observations.

Usage

```
two
```

Format

A data frame with 2999 rows and 2 variables:

x1 first covariate

x2 second covariate

y response ...

Index

*Topic **datasets**
control, 28
control_2dim, 28
one, 32
treatment, 37
treatment_2dim, 37
two, 38

*Topic **regression**,
rqss_new, 33

*Topic **robust**
rqss_new, 33

*Topic **smooth**,
rqss_new, 33

AIC, 34
as.data.frame, 2, 4, 6, 8, 9, 11, 12, 14, 16, 19, 21, 23, 26, 29–31, 35

BLB_Discretize, 2, 19
BLB_Discretize_2dim, 3, 4, 26, 27
BLB_rqss, 6, 8, 10, 15, 16, 20, 35, 36
BLB_rqss_2dim, 5, 7, 7, 11, 22
Bootstrap_rqss, 9, 11, 17, 18, 35, 36
Bootstrap_rqss_2dim, 10, 10, 24, 25

CompExp, 12
CompExperiment_1dim, 12, 13, 14, 22
CompExperiment_1dim_boot, 13, 16, 25
CompExperiment_1dim_D, 18, 27
CompExperiment_2dim, 12, 13, 16, 20, 21
CompExperiment_2dim_boot, 18, 23
CompExperiment_2dim_D, 25
CompPlot_1dim, 20
CompPlot_1dim (CompExperiment_1dim), 14
CompPlot_2dim, 27
CompPlot_2dim (CompExperiment_2dim), 21
contour, 16, 18, 20, 22, 25, 27
control, 28
control_2dim, 28

Discretize (BLB_Discretize_2dim), 4

FitPlot_1dim (BLB_rqss), 6
FitPlot_2dim (BLB_rqss_2dim), 7
foldsGenerator, 29, 30–32

image, 16, 18, 20, 22, 25, 27

lambdaSearch, 12, 29, 35, 36
logLik, 34

multifoldCV, 30, 31
multifoldCV_2dim, 30
multifoldCV_2dim (multifoldCV), 31

one, 32
optimize, 13, 30, 36

parallel, 2, 4, 6, 7, 9, 11, 12, 14, 16, 19, 21, 23, 26, 30, 31, 35

qss, 33, 34
quantreg, 2, 4, 6, 7, 9, 10, 33

rq.fit.lasso, 33
rqss, 30, 33, 34
rqss.object, 34
rqss_new, 33

searchLambda (CompExperiment_1dim), 14
searchLambda_1dim_D
(CompExperiment_1dim_D), 18
searchLambda_2dim
(CompExperiment_2dim), 21
searchLambda_2dim_D
(CompExperiment_2dim_D), 25
sfn.control, 34
SurfaceFit, 35

title, 6, 8, 13, 15, 17, 22, 24
treatment, 37
treatment_2dim, 37
two, 38