Package 'Rearrangement'

January 20, 2025

Type Package

Title Monotonize Point and Interval Functional Estimates by Rearrangement

Version 2.1

Author Wesley Graybill, Mingli Chen, Victor Chernozhukov, Ivan Fernandez-Val, Alfred Galichon

Maintainer Ivan Fernandez-Val <ivanf@bu.edu>

Description The rearrangement operator (Hardy, Littlewood, and Polya 1952) for univariate, bivariate, and trivariate point estimates of monotonic functions. The package additionally provides a function that creates simultaneous confidence intervals for univariate functions and applies the rearrangement operator to these confidence intervals.

License GPL (>= 2)

LazyLoad yes

Depends quantreg, splines

NeedsCompilation no

Repository CRAN

Date/Publication 2016-03-02 01:48:30

Contents

`Rearrangement-package` *Point and Interval Rearrangement*

Description

This package implements the rearrangement operator (Hardy, Littlewood, and Polya 1952) for univariate, bivariate, and trivariate point estimates of monotonic functions. It additionally provides a function that creates simultaneous confidence intervals for univariate functions and applies the rearrangement operator to these confidence intervals.

Details

Package:	Rearrangement
Type:	Package
Version:	1.0
Date:	2011-09-11
License:	GPL(>=2)
LazyLoad:	yes

This package is used for rearranging both point and interval estimates of a target function. Given an original point estimate of a target function, one may use `rearrangement` to monotonize this estimate. One may also create simultaneous confidence interval estimates using `simconboot` and monotonize these estimates using `rconint`.

Author(s)

Wesley Graybill, Mingli Chen, Victor Chernozhukov, Ivan Fernandez-Val, Alfred Galichon

Maintainer: Ivan Fernandez-Val <ivanf@bu.edu>

References

Chernozhukov, V., I. Fernandez-Val, and a. Galichon. 2009. Improving point and interval estimators of monotone functions by rearrangement. Biometrika 96 (3): 559-575.

Chernozhukov, V., I. Fernandez-Val, and a. Galichon. 2010. Quantile and Probability Curves Without Crossing. Econometrica 78(3): 1093-1125.

Hardy, G.H., J.E. Littlewood, and G. Polya, Inequalities,2nd ed, Cambridge U. Press,1952

Examples

```
##rearrangement example:
library(splines)
data(GrowthChart)
attach(GrowthChart)

ages <- unique(sort(age))
aknots <- c(3, 5, 8, 10, 11.5, 13, 14.5, 16, 18)
splines_age <- bs(age,kn=aknots)
sformula <- height~splines_age
sfunc <- approxfun(age,lm(sformula)$fitted.values)
splreg <- sfunc(ages)
rsplreg <- rearrangement(list(ages),splreg)
plot(age,height,pch=21,bg='gray',cex=.5,xlab="Age(years)",
ylab="Height(cms)", main="CEF (Regression Splines)",col='gray')
lines(ages,splreg,col='red',lwd=3)
lines(ages,rsplreg,col='blue',lwd=2)
legend("topleft",c('Original','Rearranged'),lty=1,col=c('red','blue'),bty='n')
detach(GrowthChart)

##rconint example:
## Not run:
data(GrowthChart)
attach(GrowthChart)

nage <- 2 * pi * (age - min(age)) / (max(age) - min(age))

formula <- height~I(sin(nage))+I(cos(nage))+I(sin(2*nage)) +
            I(cos(2*nage))+I(sin(3*nage))+
            I(cos(3*nage))+ I(sin(4*nage)) + I(cos(4*nage))

j <- simconboot(nage,height,lm,formula)
k <- rconint(j)
plot(k, border=NA, col='darkgray')
polygon.conint(j, border=NA, col='lightgray')
polygon.conint(k, border=NA, col='darkgray', density=50)
points(nage,height)
detach(GrowthChart)

## End(Not run)
```

GrowthChart *Age and Height of White Males*

Description

This data set contains age and height of US-born white males age two through twenty. Note that age is measured in months and expressed in years, and height is measured in centimeters.

Usage

```
data(GrowthChart)
```

Format

A data frame with 533 observations on the following 3 variables.

`sex` a numeric vector. Male = 1

`height` a numeric vector. Height in cm

`age` a numeric vector. Age in years

Source

The data consist of repeated cross sectional measurements of height and age from the 2003-2004 National Health and Nutrition Survey collected by the US National Center for Health Statistics.

Examples

```
data(GrowthChart)
attach(GrowthChart)
plot(age,height,pch=21,bg='gray',cex=.5,
xlab="Age (years)",ylab="Height (cms)",col='gray')
detach(GrowthChart)
```

---

lclm *Local Constant Estimator for Conditional Mean Functions*

Description

Implements the local nonparametric method kernel estimator–with box kernel (default), for conditional mean functions.

Usage

```
lclm(x, y, h, xx)
```

Arguments

`x`	The conditioning covariate
`y`	The response variable
`h`	The bandwidth parameter
`xx`	The points at which the function is to be estimated

Details

The function uses a box kernel.

Value

xx	The design points at which the evaluation occurs
fitted.values	The estimated function values at these design points

Author(s)

Wesley Graybill, Mingli Chen, Victor Chernozhukov, Ivan Fernandez-Val, Alfred Galichon

Examples

```
data(GrowthChart)
attach(GrowthChart)

ages <- unique(sort(age))
lclm.fit1 <- lclm(age,height,h=1,xx=ages)

detach(GrowthChart)
```

---

lcrq2 *Local Constant Estimator for Conditional Quantile Functions*

Description

Implements the local nonparametric method kernel estimator–with box kernel (default), for conditional quantile functions. This is a modification of Koenker's lprq (from package **quantreg**).

Usage

```
lcrq2(x, y, h, xx, tau)
```

Arguments

x	The conditioning covariate
y	The response variable
h	The bandwidth parameter
xx	The points at which the function is to be estimated
tau	The quantile(s) to be estimated. This should be a list of quantiles if the function estimates the quantile process

Details

The function uses a box kernel.

Value

xx	The design points at which the evaluation occurs
fitted.values	The estimated function values at these design points

Author(s)

Wesley Graybill, Mingli Chen, Victor Chernozhukov, Ivan Fernandez-Val, Alfred Galichon

See Also

lprq

Examples

```
require(quantreg)
data(GrowthChart)
attach(GrowthChart)

ages <- unique(sort(age))
lcq.fit1 <- lcrq2(age,height,h=1,xx=ages,tau=0.01)

detach(GrowthChart)
```

---

lines.conint *Lines Method for Simultaneous Confidence Intervals*

Description

A method for the lines generic. It graphs both the upper and lower end-point functions of a confidence interval as lines on a plot.

Usage

```
## S3 method for class 'conint'
lines(x, ...)
```

Arguments

x	object of class conint
...	further arguments to lines.default

Details

This is intended for plotting confidence intervals produced by the output of simconboot or rconint.

Author(s)

Wesley Graybill, Mingli Chen, Victor Chernozhukov, Ivan Fernandez-Val, Alfred Galichon

See Also

lines,plot.conint,points.conint

Examples

```
data(GrowthChart)
attach(GrowthChart)

nage <- 2*pi*(age-min(age))/(max(age)-min(age))
formula<-height~I(sin(nage))+I(cos(nage))+I(sin(2*nage))+
          I(cos(2*nage))+I(sin(3*nage))+ I(cos(3*nage))+I(sin(4*nage))+I(cos(4*nage))

j<-simconboot(nage,height,lm,formula)
plot(nage,height,pch=21,bg='gray',cex=.5,xlab="Age (years)",ylab="Height (cms)",col='gray',xaxt='n')
axis(1, at = seq(-2*pi*min(age)/(max(age)-min(age)),
     2*pi+1, by=5*2*pi/(max(age)-min(age))), label = seq(0, max(age)+1, by=5))
lines(j)

detach(GrowthChart)
```

---

lplm *Local Linear Regression Methods for Conditional Mean Functions*

Description

Implements the local nonparametric method, local linear regression estimator with box kernel (default), for conditional mean functions.

Usage

```
lplm(x, y, h, xx)
```

Arguments

x	The conditioning covariate
y	The response variable
h	The bandwidth parameter
xx	The points at which the function is to be estimated

Details

The function uses a box kernel.

Value

xx	The design points at which the evaluation occurs
fitted.values	The estimated function values at these design points

Author(s)

Wesley Graybill, Mingli Chen, Victor Chernozhukov, Ivan Fernandez-Val, Alfred Galichon

Examples

```
data(GrowthChart)
attach(GrowthChart)

ages <- unique(sort(age))
lplm.fit1 <- lplm(age,height,h=1,xx=ages)

detach(GrowthChart)
```

`lprq2` *Local Linear Regression Methods for Conditional Quantile Functions*

Description

Implements the local nonparametric method, local linear regression estimator with box kernel (default), for conditional quantile functions. This is a modification of Koenker's `lprq` (from package **quantreg**).

Usage

```
lprq2(x, y, h, xx, tau)
```

Arguments

`x`	The conditioning covariate
`y`	The response variable
`h`	The bandwidth parameter
`xx`	The points at which the function is to be estimated
`tau`	The quantile(s) to be estimated. This should be a list of quantiles if the function estimates the quantile process

Details

The function uses a box kernel.

Value

`xx`	The design points at which the evaluation occurs
`fitted.values`	The estimated function values at these design points

Author(s)

Wesley Graybill, Mingli Chen, Victor Chernozhukov, Ivan Fernandez-Val, Alfred Galichon

Examples

```
require(quantreg)
data(GrowthChart)
attach(GrowthChart)

ages <- unique(sort(age))
llq.fit1 <- lprq2(age,height,h=1,xx=ages,tau=0.2)
detach(GrowthChart)
```

`plot.conint` *Plot Method for Simultaneous Confidence Intervals*

Description

A method for the `plot` generic. It graphs both the upper and lower end-point functions of a confidence interval as an unfilled polygon.

Usage

```
## S3 method for class 'conint'
plot(x, border, col, ...)
```

Arguments

`x`	object of class `conint`
`border`, `col`	same usage as in `polygon`
`...`	further arguments to `plot.default`

Details

This is intended for plotting confidence intervals produced by the output of `simconboot` or `rconint`.

Author(s)

Wesley Graybill, Mingli Chen, Victor Chernozhukov, Ivan Fernandez-Val, Alfred Galichon

See Also

`plot`, `lines.conint`, `points.conint`

Examples

```
data(GrowthChart)
attach(GrowthChart)

nage <- 2 * pi * (age - min(age)) / (max(age) - min(age))
formula <- height ~ I(sin(nage))+I(cos(nage))+I(sin(2*nage))+I(cos(2*nage))+
            I(sin(3*nage))+I(cos(3*nage))+ I(sin(4*nage))+I(cos(4*nage))
j<-simconboot(nage,height,lm,formula)
plot(j)
points(nage,height)

detach(GrowthChart)
```

---

points.conint *Points Method for Simultaneous Confidence Intervals*

Description

A method for the points generic. It graphs both the upper and lower end-point functions of a confidence interval as points on a plot.

Usage

```
## S3 method for class 'conint'
points(x, ...)
```

Arguments

x	object of class conint
...	further arguments to points.default

Details

This is intended for plotting confidence intervals produced by the output of simconboot or rconint.

Author(s)

Wesley Graybill, Mingli Chen, Victor Chernozhukov, Ivan Fernandez-Val, Alfred Galichon

See Also

points, plot.conint, lines.conint

Examples

```
data(GrowthChart)
attach(GrowthChart)

nage <- 2 * pi * (age - min(age)) / (max(age) - min(age))
formula<-height~I(sin(nage))+I(cos(nage))+I(sin(2*nage))+I(cos(2*nage))+
          I(sin(3*nage))+I(cos(3*nage))+I(sin(4*nage))+I(cos(4*nage))
j <- simconboot(nage,height,lm,formula)
plot(nage,height,pch=21,bg='gray',cex=.5,xlab="Age (years)",ylab="Height (cms)",col="gray")
points(j)

detach(GrowthChart)
```

`polygon.conint` *polygon Method for Simultaneous Confidence Intervals*

Description

`polygon.conint` graphs both the upper and lower end-point functions of a confidence interval as a standard polygon on a plot.

Usage

```
polygon.conint(x, ...)
```

Arguments

`x`	object of class `conint`
`...`	further arguments to `polygon`

Details

This is intended for plotting confidence intervals produced by the output of `simconboot` or `rconint`.

Author(s)

Wesley Graybill, Mingli Chen, Victor Chernozhukov, Ivan Fernandez-Val, Alfred Galichon

See Also

`polygon`

Examples

```
## Not run: data(GrowthChart)
attach(GrowthChart)

nage <- 2 * pi * (age - min(age)) / (max(age) - min(age))
formula <- height~I(sin(nage))+I(cos(nage))+I(sin(2*nage))+I(cos(2*nage))+
            I(sin(3*nage))+I(cos(3*nage))+I(sin(4*nage))+I(cos(4*nage))
j<-simconboot(nage,height,lm,formula)
plot(nage,height,pch=21,bg='gray',cex=0.5,xlab="Age (years)",
     ylab="Height (cms)",col='gray',xaxt='n')
axis(1, at = seq(-2*pi*min(age)/(max(age)-min(age)),
        2*pi+1,by=5*2*pi/(max(age)-min(age))), label = seq(0, max(age)+1, by=5))
polygon.conint(j, border=NA, col='darkgray')

detach(GrowthChart)
## End(Not run)
```

---

rconint *Rearrangement of Simultaneous Confidence Intervals*

Description

Uses rearrangement to apply the rearrangement operator to objects of class conint.

Usage

```
rconint(x, n = 100, stochastic = FALSE, avg = TRUE)
```

Arguments

x	object of class conint
n	an integer denoting the number of sample points desired
stochastic	logical. If TRUE, stochastic sampling will be used
avg	logical. If TRUE, the average rearrangement will be computed and outputed

Details

Implements the rearrangement operator of rearrangement on simultaneous confidence intervals. Intended for use on output of simconboot.

Value

An object of class conint with the following elements:

x	the original x data
y	the original y data

sortedx	the original x data, sorted with repeated elements removed
Lower	the rearranged lower end-point function. Represented as a vector of values corresponding to sortedx
Upper	the rearranged upper end-point function. Represented as a vector of values corresponding to sortedx
cef	the corresponding estimates

Author(s)

Wesley Graybill, Mingli Chen, Victor Chernozhukov, Ivan Fernandez-Val, Alfred Galichon

References

Chernozhukov, V., I. Fernandez-Val, and a. Galichon. 2009. Improving point and interval estimators of monotone functions by rearrangement. Biometrika 96 (3): 559-575.

Chernozhukov, V., I. Fernandez-Val, and a. Galichon. 2010. Quantile and Probability Curves Without Crossing. Econometrica 78(3): 1093-1125.

See Also

simconboot, rearrangement

Examples

```
## Not run:
data(GrowthChart)
attach(GrowthChart)

nage <- 2 * pi * (age - min(age)) / (max(age) - min(age))
formula <- height ~ I(sin(nage))+I(cos(nage))+I(sin(2*nage))+I(cos(2*nage))+
            I(sin(3*nage))+I(cos(3*nage))+I(sin(4*nage))+I(cos(4*nage))
j <- simconboot(nage,height,lm,formula)
k <- rconint(j)
plot(k, border=NA, col='darkgray',xlab = 'Age (years)',ylab = 'Height (cms)',xaxt = "n")
axis(1, at = seq(-2*pi*min(age)/(max(age)-min(age)),
        2*pi+1,by=5*2*pi/(max(age)-min(age))), label = seq(0, max(age)+1, by=5))
polygon.conint(j, border=NA, col='lightgray')
polygon.conint(k, border=NA, col='darkgray', density=50)
points(nage,height,col='gray80')
legend(min(nage),max(height),c("95% CI Original",
       "95% CI  Rearranged"),lty=c(1,1),lwd=c(2,2),
       col=c("lightgray","darkgray"),bty="n");
detach(GrowthChart)

## End(Not run)
```

`rearrangement` *Rearrangement*

Description

Monotonize a step function by rearrangement. Returns a matrix or array of points which are monotonic, or a monotonic function performing linear (or constant) interpolation.

Usage

```
rearrangement(x,y,n=1000,stochastic=FALSE,avg=TRUE,order=1:length(x))
```

Arguments

`x`	a list or data frame, the entries of which are vectors containing the x values corresponding to the fitted y values
`y`	a vector, matrix, or three-dimensional array containing the fitted values of a model, typically the result of a regression
`n`	an integer denoting the number of sample points desired
`stochastic`	logical. If TRUE, stochastic sampling will be used
`avg`	logical. If TRUE, the average rearrangement will be computed and outputted
`order`	a vector containing the desired permutation of the elements of 1:length(x). The rearrangement will be performed in the order specified if avg= FALSE, otherwise all the possible orderings are computed and the average rearrangement is reported

Details

This function applies this rearrangement operator of Hardy, Littlewood, and Polya (1952) to the estimate of a monotone function.

Note: `rearrangement` currently only operates on univariate, bivariate, and trivariate regressions (that is, length(x)<=3).

Value

`rearrangement` returns a matrix or array of equivalent dimension and size to y that is monotonically increasing in all of its dimensions.

Author(s)

Wesley Graybill, Mingli Chen, Victor Chernozhukov, Ivan Fernandez-Val, Alfred Galichon

References

Chernozhukov, V., I. Fernandez-Val, and a. Galichon. 2009. Improving point and interval estimators of monotone functions by rearrangement. Biometrika 96 (3): 559-575.

Chernozhukov, V., I. Fernandez-Val, and a. Galichon. 2010. Quantile and Probability Curves Without Crossing. Econometrica 78(3): 1093-1125.

Hardy, G.H., J.E. Littlewood, and G. Polya, Inequalities,2nd ed, Cambridge U. Press,1952

See Also

rconint, quantile

Examples

```
##Univariate example:
library(splines)
data(GrowthChart)
attach(GrowthChart)

ages <- unique(sort(age))
aknots <- c(3, 5, 8, 10, 11.5, 13, 14.5, 16, 18)
splines_age <- bs(age,kn=aknots)
sformula <- height~splines_age
sfunc <- approxfun(age,lm(sformula)$fitted.values)
splreg <- sfunc(ages)
rsplreg <- rearrangement(list(ages),splreg)
plot(age,height,pch=21,bg='gray',cex=.5,xlab="Age (years)",ylab="Height (cms)",
     main="CEF (Regression Splines)",col='gray')
lines(ages,splreg,col='red',lwd=3)
lines(ages,rsplreg,col='blue',lwd=2)
legend("topleft",c('Original','Rearranged'),lty=1,col=c('red','blue'),bty='n')
detach(GrowthChart)

##Bivariate example:
## Not run: library(quantreg)
data(GrowthChart)
attach(GrowthChart)

ages <- unique(sort(age))
taus <- c(1:999)/1000
nage <- 2 * pi * (age - min(age)) / (max(age) - min(age))
nages <- 2 * pi * (ages - min(ages)) / (max(ages) - min(ages))
fform <- height ~ I(sin(nage))+I(cos(nage))+I(sin(2*nage))+I(cos(2*nage))+
          I(sin(3*nage))+I(cos(3*nage))+I(sin(4*nage))+I(cos(4*nage))
ffit <- rq(fform, tau = taus)
fcoefs <- t(ffit$coef)
freg <- rbind(1, sin(nages), cos(nages), sin(2*nages),
              cos(2*nages),sin(3*nages), cos(3*nages), sin(4*nages), cos(4*nages) )
fcqf <- crossprod(t(fcoefs),freg)
rrfcqf <- rearrangement(list(taus,ages),fcqf, avg=TRUE)
tdom <-c(1,10*c(1:99),999)
adom <-c(1,5*c(1:floor(length(ages)/5)), length(ages))
```

```
par(mfrow=c(2,1))
persp(taus[tdom],ages[adom],rrfcqf[tdom,adom],xlab='quantile',
      ylab='age',zlab='height',col='lightgreen',theta=315,phi=25,shade=.5)
title("CQP: Average Quantile/Age Rearrangement")
contour(taus,ages,rrfcqf,xlab='quantile',ylab='age',col='green',
        levels=10*c(ceiling(min(fcqf)/10):floor(max(fcqf)/10)))
title("CQP: Contour (RR-Quantile/Age)")

detach(GrowthChart)

## End(Not run)
```

`simconboot` *Simultaneous Confidence Interval Estimation using Bootstrap*

Description

`simconboot` obtains a simultaneous confidence interval for a function. It estimates the lower and upper endpoint functions of the interval by bootstrap.

Usage

```
simconboot(x,y,estimator,formula,B=200,alpha=0.05,sampsize=length(x),
           seed=8,colInt=c(5:39)/2,...)
```

Arguments

`x`	a numerical vector of x values
`y`	a numerical vector of y values
`estimator`	estimator to be used in regression
`formula`	formula to be used in the estimator
`B`	an integer with the number of bootstrap repetitions
`alpha`	a real number between 0 and 1 reflecting the desired confidence level
`sampsize`	an integer with the sample size of each bootstrap repetition
`seed`	if desired, seed to be set for the random number generator
`colInt`	the points to be evaluated when ploting
`...`	further arguments to be passed to the estimator

Details

`estimator` can be any of a set of standard regression models, most commonly `lm` or `rq` (from package **quantreg**) for global estimators and the built-in functions `lclm`, `lplm`, `lcrq2`, `lprq2` for local estimators.

Note: formula=0 for all the local estimators.

Value

An object of class `conint` with the following elements:

`x`	the original x data
`y`	the original y data
`sortedx`	the original x data, sorted with repeated elements removed
`Lower`	the lower endpoint function. Represented as a vector of values corresponding to sortedx
`Upper`	the upper endpoint function. Represented as a vector of values corresponding to sortedx
`cef`	the corresponding estimates

Author(s)

Wesley Graybill, Mingli Chen, Victor Chernozhukov, Ivan Fernandez-Val, Alfred Galichon

See Also

`rconint`

Examples

```
data(GrowthChart)
attach(GrowthChart)

nage <- 2 * pi * (age - min(age)) / (max(age) - min(age))
nages <- unique(sort(nage))
formula <- height~I(sin(nage))+I(cos(nage))+I(sin(2*nage))+I(cos(2*nage))+
            I(sin(3*nage))+I(cos(3*nage))+I(sin(4*nage))+I(cos(4*nage))
j <- simconboot(nage,height,lm,formula)
plot(j, border=NA, col='darkgray',xlab = 'Age (years)',ylab = 'Height (cms)',xaxt = "n")
axis(1, at = seq(-2*pi*min(age)/(max(age)-min(age)), 2*pi+1,
        by=5*2*pi/(max(age)-min(age))), label = seq(0, max(age)+1, by=5))
points(nage,height)
lines(nages, j$cef, lty=2, col='green')

detach(GrowthChart)
```

Index