Package ‘frechet’

June 19, 2020

Type Package

Title Statistical Analysis for Random Objects and Non-Euclidean Data

URL https://github.com/functionaldata/tFrechet

BugReports https://github.com/functionaldata/tFrechet/issues

Version 0.1.0

Encoding UTF-8

Date 2020-06-09

Maintainer Yaqing Chen <yaqchen@ucdavis.edu>

Description Provides implementation of statistical methods for random objects lying in various metric spaces, which are not necessarily linear spaces. The core of this package is Fréchet regression for random objects with Euclidean predictors, which allows one to perform regression analysis for non-Euclidean responses under some mild conditions. Examples include distributions in L^2-Wasserstein space, covariance matrices endowed with power metric (with Frobenius metric as a special case), Cholesky and log-Cholesky metrics. References: Petersen, A., & Müller, H.-G. (2019) <doi:10.1214/17-AOS1624>.

License BSD_3_clause + file LICENSE

LazyData false

Imports corrplot, fdadensity, fdapace (>= 0.5.3), Matrix, methods, pracma, rosqp

Suggests Rcpp (>= 0.11.5), testthat

RoxygenNote 7.1.0

NeedsCompilation no

Author Yaqing Chen [aut, cre],
Alvaro Gajardo [aut],
Jianing Fan [aut],
Qixian Zhong [aut],
Paromita Dubey [aut],
Kyunghee Han [aut],
Satarupa Bhattacharjee [aut],
Hans-Georg Müller [cph, ths, aut]

Repository CRAN

Date/Publication 2020-06-19 10:00:07 UTC

R topics documented:

color.bar *Generate color bar/scale.*

Description

Generate color bar/scale.

Usage

```
color.bar(
  colVal = NULL,
  colBreaks = NULL,
  min = NULL,
  max = NULL,
  lut = NULL,
  nticks = 5,
  ticks = NULL,
  title = NULL
)
```

Arguments

`colVal` A numeric vector giving the variable values to which each color is corresponding. It overrides `min` (and `max`) if `min` > `min(colVal)` (`max` < `max(colVal)`).

`colBreaks` A numeric vector giving the breaks dividing the range of variable into different colors. It overrides `min` and `max`.

`min` A scalar giving the minimum value of the variable represented by colors.

`max` A scalar giving the maximum value of the variable represented by colors.

`lut` Color vector. Default is `colorRampPalette(colors = c("pink","royalblue"))(length(colBreaks)-1)`.

`nticks` An integer giving the number of ticks used in the axis of color bar.

`ticks` A numeric vector giving the locations of ticks used in the axis of color bar; it overrides `nticks`.

`title` A character giving the label of the variable according to which the color bar is generated.

Value

No return value.

`CovFMean` *Fréchet mean of covariance matrices*

Description

Fréchet mean computation for covariance matrices.

Usage

```
CovFMean(M = NULL, optns = list())
```

Arguments

`M` A q by q by n array (resp. a list of q by q matrices) where `M[,,i]` (resp. `M[[i]]`) contains the i-th covariance matrix of dimension q by q.

`optns` A list of options control parameters specified by `list(name=value)`. See 'Details'.

Details

Available control options are

metric Metric type choice, `"frobenius"`, `"power"`, `"log_cholesky"`, `"cholesky"` - default: `"frobenius"` which corresponds to the power metric with `alpha` equal to 2.

alpha The power parameter for the power metric, which can be any non-negative integer. Default is 2 which corresponds to Frobenius metric.

Value

A list containing the following fields:

`Mout` A list containing the Fréchet mean of the covariance matrices in `M`.

`optns` A list containing the `optns` parameters utilized.

References

- *Petersen, A. and Müller, H.-G. (2019). Fréchet regression for random objects with Euclidean predictors. The Annals of Statistics, 47(2), 691–719.*
- *Petersen, A., Deoni, S. and Müller, H.-G. (2019). Fréchet estimation of time-varying covariance matrices from sparse data, with application to the regional co-evolution of myelination in the developing brain. The Annals of Applied Statistics, 13(1), 393–419.*
- *Lin, Z. (2019). Riemannian geometry of symmetric positive definite matrices via Cholesky decomposition. Siam. J. Matrix. Anal, A. 40, 1353–1370.*

Examples

```
#Example M input
n=10 #sample size
m=5 # dimension of covariance matrices
M <- array(0,c(m,m,n))
for (i in 1:n){
 y0=rnorm(m)
 aux<-diag(m)+y0%*%t(y0)
 M[,,i]<-aux
}
Fmean=CovFMean(M=M,optns=list(metric="frobenius"))
```

CreateCovRegPlot *Plots for Fréchet regression for covariance matrices.*

Description

Plots for Fréchet regression for covariance matrices.

Usage

```
CreateCovRegPlot(x, optns = list())
```

Arguments

x	A covReg object obtained from CovFMean, GloCovReg or LocCovReg.
optns	A list of control options specified by `list(name=value)`. See 'Details'.

Details

Available control options are

ind.xout A vector holding the indices of elements in `x$Mout` at which the plots will be made. Default is

- `1:length(x$Mout)` when `x$Mout` is of length no more than 3;

- c(1,round(length(x$Mout)/2),length(x$Mout)) when x$Mout is of length greater than 3.

nrow An integer — default: 1; subsequent figures will be drawn in an optns$nrow-by-ceiling(length(ind.xout)/optns$nrow) array.

plot.type Character with two choices, "continuous" and "categorical". The former plots the correlations in a continuous scale of colors by magnitude while the latter categorizes the positive and negative entries into two different colors. Default is "continuous"

plot.clust Character, the ordering method of the correlation matrix. "original" for original order (default); "AOE" for the angular order of the eigenvectors; "FPC" for the first principal component order; "hclust" for the hierarchical clustering order, drawing 4 rectangles on the graph according to the hierarchical cluster; "alphabet" for alphabetical order.

plot.method Character, the visualization method of correlation matrix to be used. Currently, it supports seven methods, named "circle" (default), "square", "ellipse", "number", "pie", "shade" and "color".

CorrOut Logical, indicating if output is shown as correlation or covariance matrix. Default is FALSE and corresponds to a covariance matrix.

plot.display Character, "full" (default), "upper" or "lower", display full matrix, lower triangular or upper triangular matrix.

Value

No return value.

Examples

```
#Example y input
n=20             # sample size
t=seq(0,1,length.out=100)       # length of data
x = matrix(runif(n),n)
theta1 = theta2 = array(0,n)
for(i in 1:n){
 theta1[i] = rnorm(1,x[i],x[i]^2)
 theta2[i] = rnorm(1,x[i]/2,(1-x[i])^2)
}
y = matrix(0,n,length(t))
phi1 = sqrt(3)*t
phi2 = sqrt(6/5)*(1-t/2)
y = theta1%*%t(phi1) + theta2 %*% t(phi2)
xout = matrix(c(0.25,0.5,0.75),3)
Cov_est=GloCovReg(x=x,y=y,xout=xout,optns=list(corrOut = FALSE, metric="power",alpha=3))
CreateCovRegPlot(Cov_est, optns = list(ind.xout = 2, plot.method = "shade"))

CreateCovRegPlot(Cov_est, optns = list(plot.method = "color"))
```

CreateDensity *Create density functions from raw data, histogram objects or frequency tables with bins*

Description

Create kernel density estimate along the support of the raw data using the HADES method.

Usage

```
CreateDensity(
  y = NULL,
  histogram = NULL,
  freq = NULL,
  bin = NULL,
  optns = list()
)
```

Arguments

`y`	A vector of raw readings.
`histogram`	A `histogram` object in R. Use this option when histogram object is only available, but not the raw data `y`. The default is `NULL`.
`freq`	A frequency vector. Use this option when frequency table is only available, but not the raw sample or the histogram object. The corresponding `bin` should be provided together. The default is `NULL`.
`bin`	A bin vector having its length with `length(freq)+1`. Use this option when frequency table is only available, but not the raw sample or the histogram object. The corresponding `freq` should be provided together.The default is `NULL`.
`optns`	A list of options control parameters specified by `list(name=value)`. See 'Details'.

Details

Available control options are

userBwMu The bandwidth value for the smoothed mean function; positive numeric - default: determine automatically based on the data-driven bandwidth selector proposed by Sheather and Jones (1991)

nRegGrid The number of support points the KDE; numeric - default: 101.

delta The size of the bin to be used; numeric - default: `diff(range(y))/1000`. It only works when the raw sample is available.

kernel smoothing kernel choice, `"rect"`, `"gauss"`, `"epan"`, `"gausvar"`, `"quar"` - default: `"gauss"`.

infSupport logical if we expect the distribution to have infinite support or not; logical - default: `FALSE`.

outputGrid User defined output grid for the support of the KDE, it overrides `nRegGrid`; numeric - default: `NULL`.

Value

A list containing the following fields:

`bw`	The bandwidth used for smoothing.
`x`	A vector of length `nRegGrid` with the values of the KDE's support points.
`y`	A vector of length `nRegGrid` with the values of the KDE at the support points.

References

- *HG Müller, JL Wang and WB Capra (1997). "From lifetables to hazard rates: The transformation approach." Biometrika 84, 881–892.*
- *SJ Sheather and MC Jones (1991). "A reliable data-based bandwidth selection method for kernel density estimation." JRSS-B 53, 683–690.*
- *HG Müller, U Stadtmüller, and T Schmitt. (1987) "Bandwidth choice and confidence intervals for derivatives of noisy data." Biometrika 74, 743–749.*

Examples

```
### compact support case

# input: raw sample
set.seed(100)
n <- 100
x0 <-seq(0,1,length.out=51)
Y <- rbeta(n,3,2)
f1 <- CreateDensity(y=Y,optns = list(outputGrid=x0))

# input: histogram
histY <- hist(Y)
f2 <- CreateDensity(histogram=histY,optns = list(outputGrid=x0))

# input: frequency table with unequally spaced (random) bins
binY <- c(0,sort(runif(9)),1)
freqY <- c()
for (i in 1:(length(binY)-1)) {
  freqY[i] <- length(which(Y>binY[i] & Y<=binY[i+1]))
}
f3 <- CreateDensity(freq=freqY, bin=binY,optns = list(outputGrid=x0))

# plot
plot(f1$x,f1$y,type='l',col=2,lty=2,lwd=2,
     xlim=c(0,1),ylim=c(0,2),xlab='domain',ylab='density')
points(f2$x,f2$y,type='l',col=3,lty=3,lwd=2)
points(f3$x,f3$y,type='l',col=4,lty=4,lwd=2)
points(x0,dbeta(x0,3,2),type='l',lwd=2)
legend('topleft',
       c('true','raw sample','histogram','frequency table (unequal bin)'),
       col=1:4,lty=1:4,lwd=3,bty='n')
```

```
### infinite support case

# input: raw sample
set.seed(100)
n <- 200
x0 <-seq(-3,3,length.out=101)
Y <- rnorm(n)
f1 <- CreateDensity(y=Y,optns = list(outputGrid=x0))

# input: histogram
histY <- hist(Y)
f2 <- CreateDensity(histogram=histY,optns = list(outputGrid=x0))

# input: frequency table with unequally spaced (random) bins
binY <- c(-3,sort(runif(9,-3,3)),3)
freqY <- c()
for (i in 1:(length(binY)-1)) {
  freqY[i] <- length(which(Y>binY[i] & Y<=binY[i+1]))
}
f3 <- CreateDensity(freq=freqY, bin=binY,optns = list(outputGrid=x0))

# plot
plot(f1$x,f1$y,type='l',col=2,lty=2,lwd=2,
     xlim=c(-3,3),ylim=c(0,0.5),xlab='domain',ylab='density')
points(f2$x,f2$y,type='l',col=3,lty=3,lwd=2)
points(f3$x,f3$y,type='l',col=4,lty=4,lwd=2)
points(x0,dnorm(x0),type='l',lwd=2)
legend('topright',
       c('true','raw sample','histogram','frequency table (unequal bin)'),
       col=1:4,lty=1:4,lwd=3,bty='n')
```

DenFMean *Fréchet means of densities.*

Description

Obtain Fréchet means of densities with respect to L^2-Wasserstein distance.

Usage

```
DenFMean(yin = NULL, hin = NULL, qin = NULL, optns = list())
```

Arguments

yin A matrix or list holding the sample of observations of the response. If yin is a matrix, each row holds the observations of the response corresponding to a row in xin.

hin A list holding the histograms of the response corresponding to each row in xin.

`qin` A matrix or list holding the quantile functions of the response. If `qin` is a matrix, each row holds the quantile function of the response taking values on `optns$qSup` corresponding to a row in `xin`.

`optns` A list of options control parameters specified by `list(name=value)`.

Details

Available control options are `qSup`, `nqSup`, `lower`, `upper`, `bwDen`, `nRegGrid`, `delta`, `kernelDen`, `infSupport`, `outputGrid`. See `LocDenReg` for details.

Value

A list containing the following components:

`dout` A matrix or list holding the output densities corresponding to `xout`. If `dout` is a matrix, each row gives a density and the domain grid is given in `dSup`. If `dout` is a list, each element is a list of two components, `x` and `y`, giving the domain grid and density function values, respectively.

`dSup` A numeric vector giving the domain grid of `dout` when it is a matrix.

`qout` A matrix holding the quantile functions of the output densities. Each row corresponds to a value in `xout`.

`qSup` A numeric vector giving the domain grid of `qout`.

`optns` A list of control options used.

Examples

```
xin = seq(0,1,0.05)
yin = lapply(xin, function(x) {
  rnorm(100, rnorm(1,x + x^2,0.005), 0.05)
})
res <- DenFMean(yin=yin)
plot(res)
```

`frechet` *frechet: Statistical Analysis for Random Objects and Non-Euclidean Data*

Description

Provides implementation of statistical methods for random objects lying in various metric spaces, which are not necessarily linear spaces. The core of this package is Fréchet regression for random objects with Euclidean predictors, which allows one to perform regression analysis for non-Euclidean responses under some mild conditions. Examples include distributions in L^2-Wasserstein space, covariance matrices endowed with power metric (with Frobenius metric as a special case), Cholesky and log-Cholesky metrics. References: Petersen, A., & Müller, H.-G. (2019) <doi:10.1214/17-AOS1624>.

GloCovReg *Global Fréchet regression of covariance matrices*

Description

Global Fréchet regression of covariance matrices with Euclidean predictors.

Usage

```
GloCovReg(x, y = NULL, M = NULL, xout, optns = list())
```

Arguments

`x`	An n by p matrix of predictors.
`y`	An n by l matrix, each row corresponds to an observation, l is the length of time points where the responses are observed. See 'metric' option in 'Details' for more details.
`M`	A q by q by n array (resp. a list of q by q matrices) where `M[,,i]` (resp. `M[[i]]`) contains the i-th covariance matrix of dimension q by q. See 'metric' option in 'Details' for more details.
`xout`	An m by p matrix of output predictor levels.
`optns`	A list of options control parameters specified by `list(name=value)`. See 'Details'.

Details

Available control options are

corrOut Boolean indicating if output is shown as correlation or covariance matrix. Default is `FALSE` and corresponds to a covariance matrix.

metric Metric type choice, `"frobenius"`, `"power"`, `"log_cholesky"`, `"cholesky"` - default: `"frobenius"` which corresponds to the power metric with `alpha` equal to 2. For power (and Frobenius) metrics, either `y` or `M` must be input; `y` would override `M`. For Cholesky and log-Cholesky metrics, `M` must be input and `y` does not apply.

alpha The power parameter for the power metric, which can be any non-negative integer. Default is 2 which corresponds to Frobenius metric.

Value

A `covReg` object — a list containing the following fields:

`xout`	An m by p matrix of output predictor levels.
`Mout`	A list of estimated conditional covariance or correlation matrices at `xout`.
`optns`	A list containing the `optns` parameters utilized.

References

- *Petersen, A. and Müller, H.-G. (2019). Fréchet regression for random objects with Euclidean predictors. The Annals of Statistics, 47(2), 691–719.*
- *Petersen, A., Deoni, S. and Müller, H.-G. (2019). Fréchet estimation of time-varying covariance matrices from sparse data, with application to the regional co-evolution of myelination in the developing brain. The Annals of Applied Statistics, 13(1), 393–419.*
- *Lin, Z. (2019). Riemannian geometry of symmetric positive definite matrices via Cholesky decomposition. Siam. J. Matrix. Anal, A. 40, 1353–1370.*

Examples

```
#Example y input
n=50             # sample size
t=seq(0,1,length.out=100)       # length of data
x = matrix(runif(n),n)
theta1 = theta2 = array(0,n)
for(i in 1:n){
 theta1[i] = rnorm(1,x[i],x[i]^2)
 theta2[i] = rnorm(1,x[i]/2,(1-x[i])^2)
}
y = matrix(0,n,length(t))
phi1 = sqrt(3)*t
phi2 = sqrt(6/5)*(1-t/2)
y = theta1%*%t(phi1) + theta2 %*% t(phi2)
xout = matrix(c(0.25,0.5,0.75),3)
Cov_est=GloCovReg(x=x,y=y,xout=xout,optns=list(corrOut=FALSE,metric="power",alpha=3))
#Example M input
n=10 #sample size
m=5 # dimension of covariance matrices
M <- array(0,c(m,m,n))
for (i in 1:n){
 y0=rnorm(m)
 aux<-diag(m)+y0%*%t(y0)
 M[,,i]<-aux
}
x=cbind(matrix(rnorm(n),n),matrix(rnorm(n),n)) #vector of predictor values
xout=cbind(runif(3),runif(3)) #output predictor levels
Cov_est=GloCovReg(x=x,M=M,xout=xout,optns=list(corrOut=FALSE,metric="power",alpha=3))
```

`GloDenReg` *Global density regression.*

Description

Global Fréchet regression for densities with respect to L^2-Wasserstein distance.

Usage

```
GloDenReg(
  xin = NULL,
  yin = NULL,
  hin = NULL,
  qin = NULL,
  xout = NULL,
  optns = list()
)
```

Arguments

`xin` An n by p matrix or a vector of length n (if p=1) with input measurements of the predictors.

`yin` A matrix or list holding the sample of observations of the response. If `yin` is a matrix, each row holds the observations of the response corresponding to a row in `xin`.

`hin` A list holding the histograms of the response corresponding to each row in `xin`.

`qin` A matrix or list holding the quantile functions of the response. If `qin` is a matrix, each row holds the quantile function of the response taking values on `optns$qSup` corresponding to a row in `xin`.

`xout` A k by p matrix or a vector of length k (if p=1) with output measurements of the predictors. Default is `xin`.

`optns` A list of options control parameters specified by `list(name=value)`.

Details

Available control options are `qSup`, `nqSup`, `lower`, `upper`, `Rsquared`, `bwDen`, `nRegGrid`, `delta`, `kernelDen`, `infSupport`, `outputGrid`. `Rsquared` is explained as follows and see `LocDenReg` for the other options.

Rsquared A logical variable indicating whether R squared would be returned. Default is `FALSE`.

Value

A list containing the following components:

`xout` Input `xout`.

`dout` A matrix or list holding the output densities corresponding to `xout`. If `dout` is a matrix, each row gives a density and the domain grid is given in `dSup`. If `dout` is a list, each element is a list of two components, `x` and `y`, giving the domain grid and density function values, respectively.

`dSup` A numeric vector giving the domain grid of `dout` when it is a matrix.

`qout` A matrix holding the quantile functions of the output densities. Each row corresponds to a value in `xout`.

`qSup` A numeric vector giving the domain grid of `qout`.

`xin` Input `xin`.

`din` Densities corresponding to the input `yin`, `hin` or `qin`.

`qin` Quantile functions corresponding to the input `yin`, `hin` or `qin`.

`Rsq` A scalar giving the R squared value if `optns$Rsquared` = `TRUE`.

`optns` A list of control options used.

References

Petersen, A., & Müller, H.-G. (2019). "Fréchet regression for random objects with Euclidean predictors." The Annals of Statistics, 47(2), 691–719.

Examples

```
xin = seq(0,1,0.05)
yin = lapply(xin, function(x) {
  rnorm(100, rnorm(1,x,0.005), 0.05)
})
qSup = seq(0,1,0.02)
xout = seq(0,1,0.25)
res1 <- GloDenReg(xin=xin, yin=yin, xout=xout, optns = list(qSup = qSup))
plot(res1)

hin = lapply(yin, function(y) hist(y, breaks = 50, plot=FALSE))
res2 <- GloDenReg(xin=xin, hin=hin, xout=xout, optns = list(qSup = qSup))
plot(res2)
```

`LocCovReg` *Local Fréchet regression of covariance matrices*

Description

Local Fréchet regression of covariance matrices with Euclidean predictors.

Usage

```
LocCovReg(x, y = NULL, M = NULL, xout, optns = list())
```

Arguments

`x` An n by p matrix of predictors.

`y` An n by l matrix, each row corresponds to an observation, l is the length of time points where the responses are observed. See 'metric' option in 'Details' for more details.

`M` A q by q by n array (resp. a list of q by q matrices) where `M[,,i]` (resp. `M[[i]]`) contains the i-th covariance matrix of dimension q by q. See 'metric' option in 'Details' for more details.

`xout` An m by p matrix of output predictor levels.

`optns` A list of options control parameters specified by `list(name=value)`. See 'Details'.

Details

Available control options are

corrOut Boolean indicating if output is shown as correlation or covariance matrix. Default is `FALSE` and corresponds to a covariance matrix.

metric Metric type choice, `"frobenius"`, `"power"`, `"log_cholesky"`, `"cholesky"` - default: `"frobenius"` which corresponds to the power metric with `alpha` equal to 2. For power (and Frobenius) metrics, either `y` or `M` must be input; `y` would override `M`. For Cholesky and log-Cholesky metrics, `M` must be input and `y` does not apply.

alpha The power parameter for the power metric, which can be any non-negative integer. Default is 2 which corresponds to Frobenius metric.

bwMean A vector of length p holding the bandwidths for conditional mean estimation if `y` is provided. If `bwMean` is not provided, it is chosen by cross validation.

bwCov A vector of length p holding the bandwidths for conditional covariance estimation. If `bwCov` is not provided, it is chosen by cross validation.

kernel Name of the kernel function to be chosen from `"rect"`, `"gauss"`, `"epan"`, `"gausvar"`, `"quar"`. Default is `"gauss"`.

Value

A `covReg` object — a list containing the following fields:

`xout` An m by p matrix of output predictor levels.

`Mout` A list of estimated conditional covariance or correlation matrices at `xout`.

`optns` A list containing the `optns` parameters utilized.

References

- *Petersen, A. and Müller, H.-G. (2019). Fréchet regression for random objects with Euclidean predictors. The Annals of Statistics, 47(2), 691–719.*
- *Petersen, A., Deoni, S. and Müller, H.-G. (2019). Fréchet estimation of time-varying covariance matrices from sparse data, with application to the regional co-evolution of myelination in the developing brain. The Annals of Applied Statistics, 13(1), 393–419.*
- *Lin, Z. (2019). Riemannian geometry of symmetric positive definite matrices via Cholesky decomposition. Siam. J. Matrix. Anal, A. 40, 1353–1370.*

Examples

```
#Example y input
n=30            # sample size
t=seq(0,1,length.out=100)       # length of data
x = matrix(runif(n),n)
```

```
theta1 = theta2 = array(0,n)
for(i in 1:n){
 theta1[i] = rnorm(1,x[i],x[i]^2)
 theta2[i] = rnorm(1,x[i]/2,(1-x[i])^2)
}
y = matrix(0,n,length(t))
phi1 = sqrt(3)*t
phi2 = sqrt(6/5)*(1-t/2)
y = theta1%*%t(phi1) + theta2 %*% t(phi2)
xout = matrix(c(0.25,0.5,0.75),3)
Cov_est=LocCovReg(x=x,y=y,xout=xout,optns=list(corrOut=FALSE,metric="power”,alpha=3))

#Example M input
n=30 #sample size
m=30 #dimension of covariance matrices
M <- array(0,c(m,m,n))
for (i in 1:n){
 y0=rnorm(m)
 aux<-15*diag(m)+y0%*%t(y0)
 M[,,i]<-aux
}
x=matrix(rnorm(n),n)
xout = matrix(c(0.25,0.5,0.75),3) #output predictor levels
Cov_est=LocCovReg(x=x,M=M,xout=xout,optns=list(corrOut=FALSE,metric="power”,alpha=0))
```

`LocDenReg` *Local density regression.*

Description

Local Fréchet regression for densities with respect to L^2-Wasserstein distance.

Usage

```
LocDenReg(
  xin = NULL,
  yin = NULL,
  hin = NULL,
  qin = NULL,
  xout = NULL,
  optns = list()
)
```

Arguments

`xin` A numeric vector holding the observations of the predictor.

`yin` A matrix or list holding the sample of observations of the response. If `yin` is a matrix, each row holds the observations of the response corresponding to a predictor value in `xin`.

`hin` A list holding the histograms of the response corresponding to each predictor value in `xin`.

`qin` A matrix or list holding the quantile functions of the response. If `qin` is a matrix, each row holds the quantile function corresponding to a predictor value in `xin`.

`xout` A numeric vector holding the output grid for the predictor. Default is `xin`.

`optns` A list of options control parameters specified by `list(name=value)`. See 'Details'.

Details

Available control options are

bwReg A scalar used as the bandwidth or a character indicating the selection method: `"CV"` (default) or `"GCV"`.

kernelReg A character holding the type of kernel functions for local Fréchet regression for densities; `"rect"`, `"gauss"`, `"epan"`, `"gausvar"`, `"quar"` - default: `"gauss"`.

qSup A numeric vector holding the grid on [0,1] quantile functions take value on. Default is an equidistant grid.

nqSup A scalar giving the length of `qSup`. Default is 201.

lower A scalar with the lower bound of the support of the distribution. Default is `NULL`.

upper A scalar with the upper bound of the support of the distribution. Default is `NULL`.

bwRange Bandwidth selection range if `bwReg` equals `"CV"` or `"GCV"`.

bwDen The bandwidth value for the smoothed mean function; positive numeric - default: determine automatically based on the data-driven bandwidth selector proposed by Sheather and Jones (1991).

ndSup The number of support points the kernel density estimation; numeric - default: 101.

dSup User defined output grid for the support of kernel density estimation, it overrides `nRegGrid`; numeric - default: `NULL`

delta The size of the bin to be used; numeric - default: `diff(range(y))/1000`. It only works when the raw sample is available.

kernelDen A character holding the type of kernel functions for density estimation; `"rect"`, `"gauss"`, `"epan"`, `"gausvar"`, `"quar"` - default: `"gauss"`.

infSupport logical if we expect the distribution to have infinite support or not; logical - default: `TRUE`

denLowerThreshold `FALSE` or a positive value giving the lower threshold of the densities; default: `0.001 * mean(qin[,ncol(qin)] -qin[,1])`.

Value

A list containing the following components:

`xout` Input `xout`.

`dout` A matrix or list holding the output densities corresponding to `xout`. If `dout` is a matrix, each row gives a density and the domain grid is given in `dSup`. If `dout` is a list, each element is a list of two components, `x` and `y`, giving the domain grid and density function values, respectively.

`dSup` A numeric vector giving the domain grid of `dout` when it is a matrix.

`qout` A matrix holding the quantile functions of the output densities. Each row corresponds to a value in `xout`.

`qSup` A numeric vector giving the domain grid of `qout`.

`xin` Input `xin`.

`din` Densities corresponding to the input `yin`, `hin` or `qin`.

`qin` Quantile functions corresponding to the input `yin`, `hin` or `qin`.

`optns` A list of control options used.

References

Petersen, A., & Müller, H.-G. (2019). "Fréchet regression for random objects with Euclidean predictors." The Annals of Statistics, 47(2), 691–719.

Examples

```
xin = seq(0,1,0.05)
yin = lapply(xin, function(x) {
  rnorm(100, rnorm(1,x + x^2,0.005), 0.05)
})
qSup = seq(0,1,0.02)
xout = seq(0,1,0.2)
res1 <- LocDenReg(xin=xin, yin=yin, xout=xout, optns = list(bw = 0.12, qSup = qSup))
plot(res1)

xout <- xin
hin = lapply(yin, function(y) hist(y, breaks = 50))
res2 <- LocDenReg(xin=xin, hin=hin, xout=xout, optns = list(qSup = qSup))
plot(res2)
```

`plot.denReg` *Plots for Fréchet regression for univariate densities.*

Description

Plots for Fréchet regression for univariate densities.

Usage

```
## S3 method for class 'denReg'
plot(
  x,
  obj = NULL,
  prob = NULL,
  xlab = NULL,
  ylab = NULL,
  main = NULL,
  ylim = NULL,
  xlim = NULL,
  col.bar = TRUE,
  widrt = 4,
  col.lab = NULL,
  nticks = 5,
  ticks = NULL,
  add = FALSE,
  pos.prob = 0.9,
  colPalette = NULL,
  ...
)
```

Arguments

`x` A denReg object, result of `DenFMean`, `GloDenReg` or `LocDenReg`.

`obj` An integer indicating which output to be plotted; 1, 2, 3, 4, and 5 for `dout`, `qout`, `din`, `qin`, and reference chart for `qout`, respectively - default: 1.

`prob` A vector specifying the probability levels for reference chart if `obj` is set to 5. Default: `c(0.05,0.25,0.5,0.75,0.95)`.

`xlab` Character holding the label for x-axis; default: `"Probability"` when `obj` is 2 or 4, `""` when `obj` is 1 or 3, `"x"` when `obj` is 5.

`ylab` Character holding the label for y-axis; default: `"Quantile"` when `obj` is 2, 4, or 5, and `"Density"` when `obj` is 1 or 3.

`main` Character holding the plot title; default: `NULL`.

`ylim` A numeric vector of length 2 holding the range of the y-axis to be drawn; default: automatically determined by the input `x`.

`xlim` A numeric vector of length 2 holding the range of the x-axis to be drawn; default: automatically determined by the input `x`.

`col.bar` A logical variable indicating whether a color bar is presented on the right of the plot - default: `TRUE`.

`widrt` A scalar giving the width ratio between the main plot and the color bar - default: 4.

`col.lab` A character giving the color bar label.

`nticks` An integer giving the number of ticks used in the axis of color bar.

`ticks` A numeric vector giving the locations of ticks used in the axis of color bar; it overrides `nticks`.

`add` Logical; only works when `obj` is 5. If `TRUE` add to an already existing plot. Taken as `FALSE` (with a warning if a different value is supplied) if no graphics device is open.

`pos.prob` `FALSE` or a scalar less than 0 or larger than 1. FALSE: no probability levels will be labeled on the quantile curves; a scalar between 0 and 1: indicating where to put the probability levels along the curves on growth charts: 0 and 1 correspond to left and right ends, respectively. Default: 0.9.

`colPalette`	A function that takes an integer argument (the required number of colors) and returns a character vector of colors interpolating the given sequence (e.g., `heat.colors`, `terrain.colors` and functions created by `colorRampPalette`). Default is `colorRampPalette(colors = c("pink","royalblue"))` for more than one curves and `"black"` otherwise.
`...`	Can set up `lty`, `lwd`, etc.

Value

No return value.

Note

see `DenFMean`, `GloDenReg` and `LocDenReg` for example code.

Index