

Package ‘frechet’

December 16, 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 0.2.0

Encoding UTF-8

Date 2020-12-15

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 (>= 0.5.5), Matrix, methods, pracma, osqp

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 2020-12-16 07:00:07 UTC

R topics documented:

color.bar	2
CovFMean	3
CreateCovRegPlot	4
CreateDensity	6
DenFMean	8
dist4cov	9
dist4den	10
frechet	11
GloCovReg	12
GloDenReg	13
LocCovReg	15
LocDenReg	17
plot.denReg	20

Index 22

color.bar	<i>Generate color bar/scale.</i>
-----------	----------------------------------

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.

<code>min</code>	A scalar giving the minimum value of the variable represented by colors.
<code>max</code>	A scalar giving the maximum value of the variable represented by colors.
<code>lut</code>	Color vector. Default is <code>colorRampPalette(colors = c("pink", "royalblue"))(length(colBreaks)-1)</code> .
<code>nticks</code>	An integer giving the number of ticks used in the axis of color bar.
<code>ticks</code>	A numeric vector giving the locations of ticks used in the axis of color bar; it overrides <code>nticks</code> .
<code>title</code>	A character giving the label of the variable according to which the color bar is generated.

Value

No return value.

CovFMean	<i>Fréchet mean of covariance matrices</i>
----------	--

Description

Fréchet mean computation for covariance matrices.

Usage

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

Arguments

<code>M</code>	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 .
<code>optns</code>	A list of options control parameters specified by <code>list(name=value)</code> . 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 1.

alpha The power parameter for the power metric, which can be any non-negative number. Default is 1 which corresponds to Frobenius metric.

Value

A list containing the following fields:

<code>Mout</code>	A list containing the Fréchet mean of the covariance matrices in <code>M</code> .
<code>optns</code>	A list containing the <code>optns</code> 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	<i>Create density functions from raw data, histogram objects or frequency tables with bins</i>
---------------	--

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

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

- H.-G. Müller, J.L. Wang and W.B. Capra (1997). "From lifetables to hazard rates: The transformation approach." *Biometrika* 84, 881–892.
- S.J. Sheather and M.C. Jones (1991). "A reliable data-based bandwidth selection method for kernel density estimation." *JRSS-B* 53, 683–690.
- H.-G. 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 measurements for the observed distributions. If yin is a matrix, each row holds the measurements for one distribution.
hin	A list holding the histograms of an observed distribution.

qin	A matrix or list holding the quantile functions of the response. If qin is a matrix, each row holds the quantile function of an observed distribution taking values on optns\$qSup. Note that only one of the three yin, hin, and qin needs to be input. If more than one of them are specified, yin overwrites hin, and hin overwrites qin.
optns	A list of options control parameters specified by list(name=value).

Details

Available control options are qSup, nqSup, bwDen, ndSup, dSup, delta, kernelDen, infSupport, and denLowerThreshold. See [LocDenReg](#) for details.

Value

A list containing the following components:

dout	A numeric vector holding the density of the Fréchet mean.
dSup	A numeric vector giving the domain grid of dout when it is a matrix.
qout	A numeric vector holding the quantile function of the Fréchet mean.
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)
```

dist4cov *Distance between covariance matrices*

Description

Distance computation between two covariance matrices

Usage

```
dist4cov(A = NULL, B = NULL, optns = list())
```

Arguments

A	an p by p matrix
B	an p by p matrix
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" and "cholesky" - default: "frobenius", which corresponds to the power metric with alpha equal to 1.

alpha The power parameter for the power metric, which can be any non-negative number. Default is 1 which corresponds to Frobenius metric.

Value

A list containing the following fields:

dist the distance between covariance matrices A and B.
optns A list containing the optns parameters utilized.

References

- Petersen, A. and Müller, H.-G. (2016). *Fréchet integration and adaptive metric selection for interpretable covariances of multivariate functional data*. *Biometrika*, 103, 103–120.
- 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.

Examples

```
# M input as array
m <- 5 # dimension of covariance matrices
M <- array(0,c(m,m,2))
for (i in 1:2) {
  y0 <- rnorm(m)
  aux <- diag(m) + y0 %*% t(y0)
  M[, ,i] <- aux
}
A <- M[, ,1]
B <- M[, ,2]
frobDist <- dist4cov(A=A, B=B, optns=list(metric="frobenius"))
```

dist4den

L^2 Wasserstein distance between two distributions.

Description

L^2 Wasserstein distance between two distributions.

Usage

```
dist4den(d1 = NULL, d2 = NULL, fctn_type = NULL, optns = list())
```

Arguments

d1, d2	Lists holding the density functions or quantile functions of the two distributions. Each list consists of two numeric vectors x and y of the same length, where x holds the support grid and y holds the values of the function. Note that the type of functions representing the distributions in $d1$ and $d2$ should be the same—either both are density functions, or both are quantile functions. If both are quantile functions, all elements in $d1\$x$ and $d2\$x$ must be between 0 and 1. $d1\$x$ and $d2\$x$ may have different lengths.
fctn_type	Character vector of length 1 holding the function type in $d1$ and $d2$ representing the distributions: "density" (default), "quantile".
optns	A list of control parameters specified by <code>list(name=value)</code> .

Details

Available control options are:

nqSup A scalar giving the length of the support grid of quantile functions based on which the L^2 Wasserstein distance (i.e., the L^2 distance between the quantile functions) is computed. Default is 201.

Value

A scalar holding the L^2 Wasserstein distance between $d1$ and $d2$.

Examples

```
d1 <- list(x = seq(-6,6,0.01))
d1$y <- dnorm(d1$x)
d2 <- list(x = d1$x + 1)
d2$y <- dnorm(d2$x, mean = 1)
dist <- dist4den(d1 = d1,d2 = d2)
```

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 <code>list(name=value)</code> . 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 1. 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. Default is 1 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))
```

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

<code>xin</code>	An n by p matrix or a vector of length n (if $p=1$) with input measurements of the predictors.
<code>yin</code>	A matrix or list holding the sample of observations of the response. If <code>yin</code> is a matrix, each row holds the observations of the response corresponding to a row in <code>xin</code> .
<code>hin</code>	A list holding the histograms of the response corresponding to each row in <code>xin</code> .
<code>qin</code>	A matrix or list holding the quantile functions of the response. If <code>qin</code> is a matrix, each row holds the quantile function of the response taking values on <code>optns\$qSup</code> corresponding to a row in <code>xin</code> . Note that only one of the three <code>yin</code> , <code>hin</code> , and <code>qin</code> needs to be input. If more than one of them are specified, <code>yin</code> overwrites <code>hin</code> , and <code>hin</code> overwrites <code>qin</code> .
<code>xout</code>	A k by p matrix or a vector of length k (if $p=1$) with output measurements of the predictors. Default is <code>xin</code> .
<code>optns</code>	A list of control parameters specified by <code>list(name=value)</code> .

Details

Available control options are `qSup`, `nqSup`, `lower`, `upper`, `Rsquared`, `bwDen`, `ndSup`, `dSup`, `delta`, `kernelDen`, `infSupport`, and `denLowerThreshold`. `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:

<code>xout</code>	Input <code>xout</code> .
<code>dout</code>	A matrix or list holding the output densities corresponding to <code>xout</code> . If <code>dout</code> is a matrix, each row gives a density and the domain grid is given in <code>dSup</code> . If <code>dout</code> is a list, each element is a list of two components, <code>x</code> and <code>y</code> , giving the domain grid and density function values, respectively.
<code>dSup</code>	A numeric vector giving the domain grid of <code>dout</code> when it is a matrix.
<code>qout</code>	A matrix holding the quantile functions of the output densities. Each row corresponds to a value in <code>xout</code> .

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 <code>list(name=value)</code> . 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 1. 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. Default is 1 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 <code>xout</code> .
optns	A list containing the <code>optns</code> 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.*, 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

<code>xin</code>	An n by p matrix or a vector of length n if $p=1$ holding the n observations of the predictor.
<code>yin</code>	A matrix or list holding the sample of observations of the response. If <code>yin</code> is a matrix, each row holds the observations of the response corresponding to a predictor value in the corresponding row of <code>xin</code> .
<code>hin</code>	A list holding the histograms of the response corresponding to each predictor value in the corresponding row of <code>xin</code> .
<code>qin</code>	A matrix or list holding the quantile functions of the response. If <code>qin</code> is a matrix, the support of the quantile functions should be the same (i.e., <code>optns\$qSup</code>), and each row of <code>qin</code> holds the quantile function corresponding to a predictor value in the corresponding row of <code>xin</code> . If the quantile functions are evaluated on different grids, then <code>qin</code> should be a list, each element consisting of two components <code>x</code> and <code>y</code> holding the support grid and the corresponding values of the quantile functions, respectively. Note that only one of the three <code>yin</code> , <code>hin</code> , and <code>qin</code> needs to be input. If more than one of them are specified, <code>yin</code> overwrites <code>hin</code> , and <code>hin</code> overwrites <code>qin</code> .
<code>xout</code>	An m by p matrix or a vector of length m if $p=1$ holding the m output predictor values. Default is <code>xin</code> .
<code>optns</code>	A list of control parameters specified by <code>list(name=value)</code> . See 'Details'.

Details

Available control options are

- bwReg** A vector of length p used as the bandwidth for the Fréchet regression or "CV" (default), i.e., a data-adaptive selection done by cross-validation.
- 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** A 2 by p matrix whose columns contain the bandwidth selection range for each corresponding dimension of the predictor `xin` for the case when `bwReg` equals "CV". Default is NULL and is automatically chosen by a data-adaptive method.
- bwDen** The bandwidth value used in `CreateDensity()` for density estimation; 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 uses in `CreateDensity()`; numeric - default: 101.
- dSup** User defined output grid for the support of kernel density estimation used in `CreateDensity()`, it overrides `nRegGrid`; numeric - default: NULL

- delta** The size of the bin to be used in `CreateDensity()`; numeric - default: `diff(range(y))/1000`. It only works when the raw sample is available.
- kernelDen** A character holding the type of kernel functions used in `CreateDensity()` for density estimation; "rect", "gauss", "epan", "gausvar", "quar" - default: "gauss".
- infSupport** logical if we expect the distribution to have infinite support or not, used in `CreateDensity()` for density estimation; logical - default: FALSE
- denLowerThreshold** FALSE or a positive value giving the lower threshold of the densities used in `CreateDensity()`; default: `0.001 * mean(qin[,ncol(qin)] - qin[,1])`.

Value

A list containing the following components:

<code>xout</code>	Input <code>xout</code> .
<code>dout</code>	A matrix or list holding the output densities corresponding to <code>xout</code> . If <code>dout</code> is a matrix, each row gives a density and the domain grid is given in <code>dSup</code> . If <code>dout</code> is a list, each element is a list of two components, <code>x</code> and <code>y</code> , giving the domain grid and density function values, respectively.
<code>dSup</code>	A numeric vector giving the domain grid of <code>dout</code> when it is a matrix.
<code>qout</code>	A matrix holding the quantile functions of the output densities. Each row corresponds to a value in <code>xout</code> .
<code>qSup</code>	A numeric vector giving the domain grid of <code>qout</code> .
<code>xin</code>	Input <code>xin</code> .
<code>din</code>	Densities corresponding to the input <code>yin</code> , <code>hin</code> or <code>qin</code> .
<code>qin</code>	Quantile functions corresponding to the input <code>yin</code> , <code>hin</code> or <code>qin</code> .
<code>optns</code>	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(0,1,0.1)
res1 <- LocDenReg(xin=xin, yin=yin, xout=xout, optns = list(bwReg = 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 <code>dout</code> , <code>qout</code> , <code>din</code> , <code>qin</code> , and reference chart for <code>qout</code> , respectively - default: 1.
prob	A vector specifying the probability levels for reference chart if <code>obj</code> is set to 5. Default: <code>c(0.05, 0.25, 0.5, 0.75, 0.95)</code> .
xlab	Character holding the label for x-axis; default: "Probability" when <code>obj</code> is 2 or 4, "" when <code>obj</code> is 1 or 3, "x" when <code>obj</code> is 5.
ylab	Character holding the label for y-axis; default: "Quantile" when <code>obj</code> is 2, 4, or 5, and "Density" when <code>obj</code> 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 <code>x</code> .

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 <code>colorRampPalette(colors = c("pink", "royalblue"))</code> 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

`color.bar`, 2
`colorRampPalette`, 21
`CovFMean`, 3, 4
`CreateCovRegPlot`, 4
`CreateDensity`, 6

`DenFMean`, 8, 20, 21
`dist4cov`, 9
`dist4den`, 10

`frechet`, 11

`GloCovReg`, 4, 12
`GloDenReg`, 13, 20, 21

`heat.colors`, 21

`LocCovReg`, 4, 15
`LocDenReg`, 9, 14, 17, 20, 21

`plot.denReg`, 20

`terrain.colors`, 21