Package ‘fdaPOIFD’

August 4, 2021

Type Package
Title Partially Observed Integrated Functional Depth
Version 1.0.2
Maintainer Antonio Elías <antonioeff91@gmail.com>
License GPL-3
Encoding UTF-8
LazyData true
RoxygenNote 7.1.1
Depends R (>= 3.5.0)
Suggests knitr, rmarkdown
VignetteBuilder knitr
Imports ggplot2, tibble, magrittr, reshape2, patchwork, MASS, fdapace, FastGP, stats
URL https://github.com/aefdz/fdaPOIFD
BugReports https://github.com/aefdz/fdaPOIFD
NeedsCompilation no
Author Antonio Elías [aut, cre],
Raúl Jiménez [ctb],
Anna M. Paganoni [ctb],
Laura M. Sangalli [ctb]
Repository CRAN
Date/Publication 2021-08-04 06:10:10 UTC
**boxplotPOFD**

**Functional Boxplot for Partially Observed Functional Data**

**Description**

Plots the Functional Boxplot for PoFD and returns the magnitude and domain outliers. Magnitude outliers in blue, a dotted red indicates that the outlier situation occurs in a region with less than $f_{dom}$ proportion of the central region.

**Usage**

```r
boxplotPOFD(data, centralRegion = 0.5, fmag = 1.5, fdom = 0)
```

**Arguments**

- `data`: matrix p by n, being n the number of functions and p the number of grid points.
- `centralRegion`: number between 0 and 1 determining the proportion of the deepest functions that builds the central region.
- `fmag`: factor to enhance the functional central region and determine the functional whiskers. Default is equal to 1.5. The whiskers provide the rule to unmask magnitude outliers.
- `fdom`: factor that provides the maximum proportion of observed functions in the central region to consider a magnitude outlier as a domain outlier also. A value equals to 0 means that domain outliers are those functions that are observed on the domain where any of the functions building the central region are observed. A value equals to 1 determine as domain outlier any magnitude outlier out of the region where the central region is completely observed.

**Value**

a list with the functional boxplot for PoDF the magnitude outliers and the domain outliers.

**References**

### Examples

```r
data(exampleData)
boxplotPOFD(exampleData$PoFDextremes_outliers, centralRegion = 0.5, fmag = 1.5, fdom = 0)
```

---

### Description

Generates samples of functions observed in a common domain in the center part of the domain. See Elías et al (2020).

### Usage

```r
commmondomainPOFD(data, observability = NULL, pIncomplete = NULL)
```

### Arguments

- **data**: functional data completely observed. pxn matrix being n the number of curves and p the number of evaluation points.
- **observability**: mean observed proportion of the domain where each function is observed.
- **pIncomplete**: number between 0 and 1 related to the proportion of curves that suffers partially observability. The default is 1 meaning that all the sample curves are partially observed.

### Value

A list containing two elements 1) a functional sample and 2) the same sample of functions but partially observed following one of the schemes described in the argument type.

### References


### Examples

```r
data <- sapply(1:100, function(x) runif(1)*sin(seq(0, 2*pi, length.out = 200)) +
runif(1)*cos(seq(0, 2*pi, length.out = 200)))
data_pofd <- commmondomainPOFD(data, observability = 0.5, pIncomplete = 1)
```
**Description**

An illustrative Functional Gaussian processes with different partially observed patterns with outliers and without outliers.

**Usage**

```r
data(exampleData)
pplotPOFD(exampleData$PoFDintervals)
```

**Format**

A list with three data sets (functions by columns):

- **PoFDintervals** Partially observed functional data in intervals
- **PoFDextremes** Partially Observed functional data with missing intervals at the extremes
- **PoFDextremes_outliers** Same as above but including two magnitude and shape outliers

**References**


**Examples**

```r
data(exampleData)
pplotPOFD(exampleData$PoFDintervals)
```

---

**intervalPOFD**

*Random Interval Observability*

**Description**

Generates samples of functions observed in different intervals. See Elías et al (2020).

**Usage**

```r
intervalPOFD(data, observability = NULL, ninterval = NULL, pIncomplete = NULL)
```
outliergramPOFD

Arguments

data functional data completely observed. p\*n matrix being n the number of curves and p the number of grid evaluation points.
observability mean observed proportion of the domain where each function is observed.
ninterval if type = "interval", n_interval is an integer with the number of observed intervals 1, 2, 3... Large values of this parameter requires a large parameter p to guarantee the observability level.
pIncomplete number between 0 and 1 related to the proportion of curves that suffers partially observability. The default is 1 meaning that all the sample curves are partially observed.

Value

a list containing two elements 1) a functional sample and 2) the same sample of functions but partially observed following one of the schemes described in the argument type.

References


Examples

data <- sapply(1:100, function(x) runif(1)*sin(seq(0, 2*pi, length.out = 200)) + runif(1)*cos(seq(0, 2*pi, length.out = 200)))
data_pofd <- intervalPOFD(data, observability = 0.5, ninterval = 2, pIncomplete = 1)

outliergramPOFD  Outliergram for Partially Observed Functional Data

Description

Plots the Outliergram for PoFD and returns the shape outliers.

Usage

outliergramPOFD(data, fshape = 1.5, p1 = 1, p2 = 0)

Arguments

data matrix p by n, being n the number of functions and p the number of grid points.
fshape inflation of the outliergram that determine the shape outlier rule.
p1 parameter of the outliergram for resampling method. Default = 1.
p2 parameter of the outliergram for resampling method. Default = 0.
Value

a list with the functional outliergram for PoDF and the shape outliers.

References


Examples

data(exampleData)
outliergramPOFD(exampleData$PoFDextremes_outliers, fshape = 1.5, p1 = 1, p2 = 0)

plotPOFD

Plot Partially Observed Functional Data

Description

Plot the sample of partially observed curves and the proportion of observed functions.

Usage

plotPOFD(data)

Arguments

data matrix p by n, being n the number of functions and p the number of grid points.

Value

Plot of the partially observed functional data and the proportion of observed functions at each time point.

Examples

data(exampleData)
plotPOFD(exampleData$PoFDextremes)
**Description**

Compute the depth measures of a partially observed functional data set evaluated in a common grid.

**Usage**

```r
POIFD(data, type = c("MBD", "FMD", "MHRD"), phi)
```

**Arguments**

- `data`: matrix p by n, being n the number of functions and p the number of grid points. Rownames are the dense grid x and colnames the identifier of each functional data.
- `type`: chosen depth measure. Fraiman and Muniz depth ("FMD"), Modified band depth ("MBD") or Modified Half Region Depth and Modified Epigraph/Hipograph Index "MHRD")
- `phi`: phi function of weights for the POIFD. The default value is as in the paper, i.e. the proportion of observed functions at each time point.

**Value**

Ordered vector of depths from the deepest to outward. The names are the functions names (if provided) or the column position.

**Examples**

```r
data(exampleData)
data <- exampleData$PoFDintervals
poifd <- POIFD(data, type = c("MBD"))
```

---

**sparsePOFD**

**Sparse Observability**

**Description**


**Usage**

```r
sparsePOFD(data, observability = NULL, pIncomplete = NULL)
```
Arguments

data: functional data completely observed. \( p \times n \) matrix being \( n \) the number of curves and \( p \) the number of evaluation points.

observability: observed proportion of the domain where each function is observed.

pIncomplete: number between 0 and 1 related to the proportion of curves that suffers partially observability. The default is 1 meaning that all the sample curves are partially observed.

Value

a list containing two elements 1) a functional sample and 2) the same sample of functions but partially observed following one of the schemes described in the argument type.

References


Examples

data <- sapply(1:100, function(x) runif(1)*sin(seq(0, 2*pi, length.out = 200)) + runif(1)*cos(seq(0, 2*pi, length.out = 200)))
data_pofd <- sparsePOFD(data, observability = 0.5, pIncomplete = 1)
Index

* datasets
  exampleData, 4
boxplotPOFD, 2
commondomainPOFD, 3
eexampleData, 4
intervalPOFD, 4
outliergramPOFD, 5
plotPOFD, 6
POIFD, 7
sparsePOFD, 7