Package ‘rainbow’

January 29, 2019

Type Package
Title Bagplots, Boxplots and Rainbow Plots for Functional Data
Version 3.6
Date 2019-1-29
Depends R (>= 3.4.0), MASS, pcaPP
Suggests forecast
Imports hdrcde, cluster, colorspace, ks
LazyLoad yes
LazyData yes
ByteCompile TRUE
Maintainer Han Lin Shang <han.lin.shang@anu.edu.au>
Description Visualizing functional data and identifying functional outliers.
License GPL-3
NeedsCompilation no
Author Han Lin Shang [aut, cre, cph] (<https://orcid.org/0000-0003-1769-6430>),
Rob Hyndman [aut] (<https://orcid.org/0000-0002-2140-5352>)
Repository CRAN
Date/Publication 2019-01-29 16:50:03 UTC

R topics documented:

rainbow-package ......................................................... 2
Australiafertility ......................................................... 3
ElNino_ERSST_region_land2 ............................................. 4
ElNino_OISST_region_land2 ............................................ 6
fboxplot ................................................................. 7
fdepth ................................................................. 9
fds ................................................................. 10
foutliers .......................................................... 12
plot.fdepth ......................................................... 13
Description

This package presents the rainbow plots, bagplots and boxplots for functional data. The latter two can also be used to identify outliers, which have either the lowest depth or the lowest density, respectively.

Author(s)

Han Lin Shang and Rob J Hyndman

Maintainer: Han Lin Shang <hanlin.shang@anu.edu.au>

References


Australiafertility

Australian age-specific fertility data

Description

Age-specific fertility rates between ages 15 and 49 in Australia from 1921 to 2006.

The age-specific fertility rates can be smoothed using a weighted median smoothing B-splines, constrained to be concave.

Usage

data(Australiafertility)
data(Australiasmoothfertility)

Format

An object of class fts.
Details

Australian fertility rates and populations (1921-2006) for age groups (15-49) were obtained from the Australian Bureau of Statistics (Cat.No.3105.0.65.001, Table 38). These are defined as the number of live births during the calendar year, according to the age of the mother, per 1000 of the female resident population of the same age at 30 June.

Australia smooth fertility is the smoothed version of Australia fertility data. The smoothing technique is the penalized regression spline with concave constraint, described in Hyndman and Ullah (2007).

Author(s)

Han Lin Shang

Source

The Australian Demographic Data Bank (courtesy of Len Smith).

References


Examples

```r
plot(Australiabirths)
plot(Australiasmoothbirths)
```

---

*ElNino_ERSST_region_1and2*

*Sea surface temperature data set from January 1950 to December 2018*

---

Description

Sea surface temperature data set from January 1950 to December 2018 observed by the extended reconstructed sea surface temperature
Usage

data(ElNino_ERSST_region_1and2)
data(ElNino_ERSST_region_3)
data(ElNino_ERSST_region_4)
data(ElNino_ERSST_region_3and4)

Format

An object of class sfts.

Details

These averaged monthly sea surface temperatures are measured by the different moored buoys in the "Nino region" defined by the coordinates 0-10 degree South and 90-80 degree West.

Source

National Weather Service Climate Prediction Center website at http://www.cpc.ncep.noaa.gov/data/indices/sstoi.indices. The data is the third column with the title NINO1+2.

References


Examples

data(ElNino_ERSST_region_1and2)
data(ElNino_ERSST_region_3)
data(ElNino_ERSST_region_4)
data(ElNino_ERSST_region_3and4)
ElNino_OISST_region_1and2

Sea surface temperature data set from January 1982 to December 2018

Description

Monthly sea surface temperatures from January 1982 to December 2018 observed by the optimum interpolation sea surface temperature.

Usage

data(ElNino_OISST_region_1and2)
data(ElNino_OISST_region_3)
data(ElNino_OISST_region_4)
data(ElNino_OISST_region_3and4)

Format

An object of class sfts.

Details

These averaged monthly sea surface temperatures are measured by the different moored buoys in the "Nino region" defined by the coordinates 0-10 degree South and 90-80 degree West.

Source

National Weather Service Climate Prediction Center website at http://www.cpc.ncep.noaa.gov/data/indices/sstoi.indices. The data is the third column with the title NINO1+2.

References


Examples

plot(ElNino_OISST_region_1and2)
plot(ElNino_OISST_region_3)
plot(ElNino_OISST_region_4)
plot(ElNino_OISST_region_3and4)

Description

Compute bivariate bagplot, functional bagplot and bivariate HDR boxplot, functional HDR boxplot.

Usage

fboxplot(data, plot.type = c("functional", "bivariate"),
type = c("bag", "hdr"), alpha = c(0.05, 0.5), projmethod = c("PCAproj","rapca"),
factor = 1.96, na.rm = TRUE, xlab = data$xname, ylab = data$yname,
shadecols = gray((9:1)/10), pointcol = 1, plotlegend = TRUE,
legendpos = "topleft", ncol = 2, ...)

Arguments

data An object of class fds or fts.
plot.type Version of boxplot. When plot.type="functional", a functional plot is provided. When plot.type="bivariate", a square bivariate plot is provided.
type Type of boxplot. When type = "bag", a bagplot is provided. When type = "hdr", a HDR boxplot is provided.
alpha Coverage probability for the functional HDR boxplot. \( \alpha \) are the coverage percentages of the outliers and the central region.
factor When type = "bag", the outer region of a bagplot is the convex hull obtained by inflating the inner region by the bagplot factor.
na.rm Remove missing values.
xlab A title for the x axis.
ylab A title for the y axis.
shadecols Colors for shaded regions.
pointcol Color for outliers and mode.
plotlegend Add a legend to the graph.
legendpos Legend position. By default, it is the top right corner.
ncol Number of columns in the legend.
projmethod Method used for projection.
... Other arguments.
Details

The functional curves are first projected into a finite dimensional subspace via functional principal component decomposition. For simplicity, we choose the subspace as $R^2$. Based on Tukey (1974)’s halfspace bagplot and Hyndman (1996)’s HDR boxplot, we order each data point in $R^2$ by data depth and data density. Outliers are those that have either lowest depth (distance from the centre) or lowest density.

Value

Function produces a graphical plot.

Author(s)

Rob J Hyndman, Han Lin Shang. Please, report bugs and suggestions to hanlin.shang@anu.edu.au

References


See Also

SVDplot

Examples

fboxplot(data = ElNino_OISST_region_1and2, plot.type = "functional", type = "bag", projmethod="PCAproj")
fboxplot(data = ElNino_OISST_region_1and2, plot.type = "bivariate", type = "bag", projmethod="PCAproj")
fdepth

Compute functional depth.

Description

Compute functional depth.

Usage

fdepth(data, type = c("FM", "mode", "RP", "RDP", "radius"), trim = 0.25, alpha, weight)

Arguments

data An object of class fds or fts.
type Type of functional depth. See detail section below.
trim Percentage of trimming.
alpha Tuning parameter used when type = "radius".
weight Hard-thresholding or soft-thresholding when type = "radius".

Details

If type = "FM", it computes the functional depth of Fraiman and Muniz (2001), which is considered as the first functional depth.

If type = "mode", it computes the functional depth of Cuevas et al. (2006). A functional mode is defined as the curve most densely surrounded by the rest of curves of the dataset.

If type = "RP" and type = "RDP", it computes random projection functional depth of Cuevas et al. (2007). Cuevas et al. (2007) considered the random projection depth based on measuring the depth of the functional data under projections and taking additional information of their derivatives. The basic idea is to project each functional curve, along a random direction, defining a point in $R^2$. A data depth in $R^2$ provides an order of the projected points.

If type = "radius", it ranks the curves according to alpha-radius. Then using the hard thresholding or soft thresholding, trimmed mean and median can be computed.

The argument trim = 0.25 first order curves by depth, and then trim 25 percent curves that have comparably lower depth.

Value

A list containing the following components is returned.

median Median curve (highest depth).
med Index of median curve.
ltrim Indexes of the trimmed curves.
prof Functional depth for each curve.
mtrim Mean of trimmed curves.
weight Weight values for all observations, when type = "radius".
Author(s)
Han Lin Shang

References


Examples

fdepth(data = ElNino_OISST_region_land2, type = "FM")
fdepth(data = ElNino_OISST_region_land2, type = "mode")
fdepth(data = ElNino_OISST_region_land2, type = "RP")
fdepth(data = ElNino_OISST_region_land2, type = "RPD")
fdepth(data = ElNino_OISST_region_land2, type = "radius", trim = 0.25, alpha = 0.5, weight = "hard")
fdepth(data = ElNino_OISST_region_land2, type = "radius", trim = 0.25, alpha = 0.5, weight = "soft")

fds Create functional objects
Description

The function `fds` is used to create general independent and identically distributed (i.i.d.) functional objects that are not ordered by time. The function `fts` is used to create functional time series objects. The function `sfts` is used to create sliced functional time series objects, where the `x` variable is also a time variable.

Usage

```r
dfs(x, y, xname, yname)
fts(x, y, start = 1, frequency = 1, xname, yname)
sfts(data, period = frequency(data), start = tsp(data)[1],
     frequency = 1, xname, yname)
```

Arguments

- `x` Numeric vector of length `p`.
- `y` Matrix of size `p` × `n` representing `n` functions of `x` observed at points `1, ..., p`.
- `data` An object of class `ts`.
- `period` Time period of sliced functional data. For instance, `period = 12` is a monthly data.
- `start` The time of the first observation. Either a single number or a vector of two integers, which specify a natural time unit and a (1-based) number of samples into the time unit. See `ts` for details.
- `frequency` The number of observations per unit of time.
- `xname` Character string giving name of `x` vector. (optional)
- `yname` Character string giving name of `y` vector. (optional)

Value

An object of class `fds` or `fts` or `sfts`.

Author(s)

Rob J Hyndman and Han Lin Shang

Examples

```r
fds(x = 1:20, y = Simulation$data$y, xname = "x", yname = "Simulated value")
fts(x = 15:49, y = Australia$smoothfertility$y, xname = "Age",
    yname = "Fertility rate")
sfts(ts(as.numeric(EiNino_O1SST_region_land2$y), frequency = 12), xname = "Month",
    yname = "Sea surface temperature")
```
foutliers

Functional outlier detection methods.

Description

Functional outlier detection methods.

Usage

foutliers(data, method = c("robMah", "lrt", "depth.trim", "depth.pond", "HUoutliers"), dfunc = depth.mode, nb = 200, suav = 0.05, trim = 0.1, order = 2, lambda = 3.29,...)

Arguments

data An object of class fds or fts.
method Outlier detection method.
dfunc When method = "lrt" or method = "depth.trim" or method = "depth.pond", users can specify the type of depth functions with possible choices of depth.FM, depth.mode, depth.RP, depth.RPD.
 nb When method = "lrt", users can specify the number of bootstrap samples.
suav When method = "lrt", users can specify the smoothing parameter used in the smoothed bootstrap samples to determine the cutoff value.
trim When method = "lrt" or method = "depth.trim" or method = "depth.pond", users can specify the trimming percentage.
order When method = "HUoutliers", users can specify the number of principal components.
lambda When method = "HUoutliers", users can specify the value of tuning parameter.
... Other arguments.

Details

When method = "lrt", the outlier detection method corresponds to the approach of Febrero et al. (2007) using the likelihood ratio test.

When method = "depth.trim", the outlier detection method corresponds to the approach of Febrero et al. (2008) using the functional depth with trimmed curves.

When method = "depth.pond", the outlier detection method corresponds to the approach of Febrero et al. (2008) using the functional depth with all curves.

When method = "HUoutliers", the outlier detection method corresponds to the approach of Hyndman and Ullah (2008) using the integrated square forecast errors.

When method = "robMah", the outlier detection method corresponds to the approach of Rousseeuw and Leroy (1987) using the robust Mahalanobis distance.
plot.fdepth

Value
A list containing the following components is returned.

- **outliers**: Detected outliers.
- **cutoff**: Threshold value to separate outliers from non-outliers, when method = "lrt", method = "depth.trim", and method = "depth.pond".
- **depth.total**: Depth measure of each functional curve.
- **depth.out**: Depth measure of functional outliers.

Author(s)
Han Lin Shang

References


Examples
```routliers(data = ElNino_OISST_region_1and2, method = "depth.trim")
```

---

**plot.fdepth**

Plot functional depth

Description
Plot functional depth.

Usage
```r
## S3 method for class 'fdepth'
plot(x, show.legend = TRUE, pos.legend = "bottomleft", ...)
```
Arguments

x  
An object of class fdepth.

show.legend  
Is legend required in the plot?

pos.legend  
When show.legend = TRUE, users can specify the position of the legend.

...  
Other plotting parameters passed to par.

Value

Function produces a plot.

Author(s)

Rob J Hyndman, Han Lin Shang

References


See Also

fdepth

Examples

plot(fdepth(E1Nino_OISST_region_1and2))

Description

Plot functional objects.

Usage

## S3 method for class 'fds'
plot(x, plot.type = c("functions", "time", "depth", "density"),
     col = NULL, type = "l", lty = 1, xlab = x$name, ylab = y$name,
     pch = c(1:9, 0, letters, LETTERS), add = FALSE, index,
     colorchoice = c("rainbow", "heat.colors", "terrain.colors",
     "topo.colors", "cm.colors", "rainbow_hcl", "gray", "sequential_hcl",
     "heat_hcl", "terrain_hcl", "diverge_hcl"), plotlegend = FALSE,
     legendpos = "topright", ncol = 1, ...)

## S3 method for class 'fds'
lines(x, type = "l", index, ...)

## S3 method for class 'fds'
points(x, type = "p", index, ...)
Arguments

x An object of class fds or fts.
plot.type Type of plot. See details for more explanations.
col Colors to use in plot. Default in plot.fds is to use a rainbow color palette with the number of colors equal to the number of functions.
type 1-character string giving the type of plot desired.
lty The line type.
xlab A title for x axis.
ynlab A title for y axis.
pch Either an integer specifying a symbol or a single character to be used as the default in plotting points.
add If add = TRUE, it plots a line or points.
index Index of a specific curve that is plotted as a line or points.
colorchoice Color palette used for drawing the rainbow plot.
plotlegend Add a legend to the graph.
legendpos Position of legend.
ncol Number of column in the legend.
... Other plotting parameters passed to par.

Details

If plot.type = "functions", then functions are plotted using a rainbow color palette so the first few functions are shown in red, followed by orange, yellow, green, blue and indigo with the last few functions plotted in violet.
If plot.type = "time", then each value of x is shown as a separate time series in a time plot.
If plot.type = "depth", then functions are first ordered by depth and then plotted using a rainbow color palette.
If plot.type = "density", then functions are first ordered by density and then plotted using a rainbow color palette.

Value

Function produces a plot.

Author(s)

Rob J Hyndman, Han Lin Shang. Please, report bugs and suggestions to hanlin.shang@anu.edu.au

References

See Also

`fds`, `lines.fds`, `points.fds`

Examples

```r
plot(x = ElNino_OISST_region_land2, plot.type = "functions", legend = TRUE)
lines(x = ElNino_OISST_region_land2, plot.type = "functions", index = 3)
points(x = ElNino_OISST_region_land2, plot.type = "functions", index = 3)
```

**Simulationdata**

*Simulated data*

**Description**

Simulated data used in Hyndman and Shang (2008).

**Usage**

```r
data(Simulationdata)
```

**Format**

An object of class `fds`.

**References**


**Examples**

```r
plot(Simulationdata, col = rainbow(100))
lines(Simulationdata, index = 991:1000, col = "black")
```

**summaryfunction**

*Summary statistics for functional data*

**Description**

Display summary statistics (minimum, 1st quantile, median, mean, 3rd quantile, maximum) and quantiles of functional data

**Usage**

```r
summaryfunction(ftsdata, plot.type = c("summarystats", "quantilestats"),
quantilepercent = seq(0.1, 0.9, by = 0.1), plot.legend = FALSE,
legendpos = "topright", cex = 0.9, lwd = 1, lty = 1, ncol = 2)
```
Args

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ftsdata</td>
<td>An object of class fds.</td>
</tr>
<tr>
<td>plot.type</td>
<td>Summary statistics or quantiles.</td>
</tr>
<tr>
<td>quantilepercent</td>
<td>Percentage of quantiles.</td>
</tr>
<tr>
<td>plot.legend</td>
<td>Is the legend required in the plot?</td>
</tr>
<tr>
<td>legendpos</td>
<td>When the plot.legend = TRUE, position of the legend.</td>
</tr>
<tr>
<td>cex</td>
<td>When the plot.legend = TRUE, point size.</td>
</tr>
<tr>
<td>lwd</td>
<td>When the plot.legend = TRUE, width of line.</td>
</tr>
<tr>
<td>lty</td>
<td>When the plot.legend = TRUE, line type.</td>
</tr>
<tr>
<td>ncol</td>
<td>When the plot.legend = TRUE, number of columns in the legend.</td>
</tr>
</tbody>
</table>

Details

A function for displaying summary statistics or quantiles of functional data.

Value

Return a plot of summary statistics of functional data or a plot of quantiles of functional data.

Author(s)

Han Lin Shang. Please, report bugs and suggestions to hanlin.shang@anu.edu.au

See Also

fds

Examples

```
summaryfunction(Australiasmoothfertility, plot.type = "summarystats")
summaryfunction(Australiasmoothfertility, plot.type = "quantilestats", plot.legend = TRUE)
```

svdplot

Singular value decomposition plot

Description

The singular value decomposition (SVD) plot of Zhang et al. (2007) captures the changes in the singular columns as the number of curves gradually increases.

Usage

```
svdplot(object, order = 3, plot = TRUE, plot.type = c("fts", "image"),
mfrow = c(2, 3))
```
**Arguments**

- **object** An object of `fds`.
- **order** Number of Singular Value Decomposition (SVD) components. The maximum order is 4.
- **plot** Is graphical display required?
- **plot.type** Plot functional time series or images?
- **mfrow** Grid of graphics.

**Details**

By using the SVD, Zhang et al. (2007) proposed a plot for visualizing patterns of functional time series. They considered a set of curves as a two-way \((p \times n)\) data matrix, where \(p\) is the total number of covariates and \(n\) is the total number of curves.

The main advantage of this dynamic plot is to visualize both column and row information of a two-way matrix simultaneously, relate the matrix to the corresponding curves, show local variations, and highlight interactions between columns and rows of a two-way matrix.

**Value**

- When `plot = TRUE`, it returns a plot.
- When `plot = FALSE`, it returns the following:
  - **svds** A number of singular value decomposition ordered by the amount of explained variation.
  - **reconstruction** Reconstruction of the original data using the SVD.
  - **residual** Residual of the original data.

**Note**


Using the animate package of Grahn(2011), a set of dynamic movies can be created to visualize the changes in singular rows and singular columns.

**Author(s)**

Han Lin Shang. Please, report bugs and suggestions to hanlin.shang@anu.edu.au

**References**


See Also

- fboxplot.svd

Examples

```r
SVDplot(ElNino_OISST_region_land2)
SVDplot(ElNino_OISST_region_land2, plot.type = "image")
```
Index

*Topic **datasets**
  * Australiafertility, 3
  * ELNino_ERSST_region_1and2, 4
  * ELNino_OISST_region_1and2, 6
  * Simulationdata, 16

*Topic **distribution**
  * fdepth, 9

*Topic **hplot**
  * plot.fdepth, 13
  * plot fds, 14

*Topic **methods**
  * summaryfunction, 16

*Topic **multivariate**
  * fboxplot, 7
  * foutliers, 12
  * SVDplot, 17

*Topic **package**
  * rainbow-package, 2

*Topic **ts**
  * fds, 10

Australiafertility, 3
Australiasmoothfertility
  (Australiafertility), 3

ELNino_ERSST_region_1and2, 4
ELNino_ERSST_region_3
  (ELNino_ERSST_region_1and2), 4
ELNino_ERSST_region_3and4
  (ELNino_ERSST_region_1and2), 4
ELNino_ERSST_region_4
  (ELNino_ERSST_region_1and2), 4
ELNino_OISST_region_1and2, 6
ELNino_OISST_region_3
  (ELNino_OISST_region_1and2), 6
ELNino_OISST_region_3and4
  (ELNino_OISST_region_1and2), 6
ELNino_OISST_region_4
  (ELNino_OISST_region_1and2), 6

fboxplot, 7, 19
fdepth, 9, 14
fds, 7, 9, 10, 12, 15–18
foutliers, 12
fts (fds), 10
lines fds, 16
lines fds (plot fds), 14
par, 14, 15
plot fdepth, 13
plot fds, 14
points fds, 16
points fds (plot fds), 14
rainbow (rainbow-package), 2
rainbow-package, 2
sfts (fds), 10
Simulationdata, 16
summaryfunction, 16
svd, 19
SVDplot, 8, 17
ts, 11