Package ‘Rfssa’

February 11, 2019

Type Package

Title Functional Singular Spectrum Analysis

Version 0.0.1

Maintainer Hossein Haghbin <haghbinh@gmail.com>

URL https://github.com/haghbinh/Rfssa.git

Description Methods and tools for implementing functional singular spectrum analysis for functional time series
as described in Haghbin H., Najibi, S.M., Mahmoudvand R., Maadooliat M. (2019). Functional singular spectrum Analysis. Manuscript submitted for publication.

License GPL (>= 2)

Encoding UTF-8

LazyData TRUE

RoxygenNote 6.1.1

Imports Rcpp, fda, lattice, plot3D,

LinkingTo Rcpp, RcppArmadillo,

Suggests knitr,

Depends R (>= 2.10)

NeedsCompilation yes

Author Hossein Haghbin [aut, cre],
Seyed Morteza Najibi [aut]

Repository CRAN

Date/Publication 2019-02-11 14:13:16 UTC

R topics documented:

`Callcenter` *Number of calls for a bank.*

Description

This dataset is a small call center for an anonymous bank (Brown et al., 2005). This dataset provides the exact time of the calls that were connected to the call center from January 1 to December 31 in the year 1999. The data are aggregated into time intervals to obtain a data matrix. More precisely, the *(i,j)*'th element of the data matrix contains the call volume during the *j*th time interval on day *i*. This dataset has been analyzed in several prior studies; e.g. Brown et al. (2005), Shen and Huang (2005), Huang et al. (2008), and Maadooliat et al. (2015). Here, the data are aggregated into time intervals 6 minutes.

Usage

```
Callcenter
```

Format

A dataframe with 87600 rows and 5 variables:

calls The number of calls in 6 minutes aggregated interval.

u a numeric vector to show the aggregated interval.

Date Date time when the calls counts are recorded.

Day Weekday associated with Date.

Month Month associated with Date.

References

1. Brown, L., Gans, N., Mandelbaum, A., Sakov, A., Shen, H., Zeltyn, S., & Zhao, L. (2005). Statistical analysis of a telephone call center: A queueing-science perspective. *Journal of the American statistical association*, **100**(469), 36-50.
2. Shen, H., & Huang, J. Z. (2005). Analysis of call center arrival data using singular value decomposition. Applied Stochastic Models in Business and Industry, 21(3), 251-263.
3. Huang, J. Z., Shen, H., & Buja, A. (2008). Functional principal components analysis via penalized rank one approximation. *Electronic Journal of Statistics*, **2**, 678-695.
4. Maadooliat, M., Huang, J. Z., & Hu, J. (2015). Integrating data transformation in principal components analysis. *Journal of Computational and Graphical Statistics*, **24**(1), 84-103.

See Also

Deathrate, fssa

Deathrate *The Age-specific death rates dataset.*

Description

The death rates for years 1921 to 2014 and age range 0 to 100.

Usage

```
Deathrate
```

Format

A dataframe with 10434 rows and 5 variables:

Year The year.

Age The age.

Female The female death rate.

Male The male death rate.

Total The total death rate.

Source

http://www.mortality.org/

See Also

Callcenter

---

freconstruct *Reconstruction sage of FSSA*

Description

This is a function for the reconstructing functional time series from functional singular spectrum objects (including Grouping and Hankelization steps). The output is a list of functional time series corresponds to each group.

Usage

```
freconstruct(U, group = as.list(1L:10L))
```

Arguments

`U`	an object of class `fssa`
`group`	a list of numeric vectors, each vector includes indices of such elementary components of a group used for reconstruction.

Value

a named list of reconstructed functional time series in each groups and a numeric vector of eigenvalues.

See Also

`fssa`

`fssa` *Functional Singular Spectrum Analysis*

Description

fssa is a function for decomposition stage (including embedding and functional SVD steps) of a functional time series.

Usage

```
fssa(Y, L = floor(dim(Y$coefs)[2L]/2L))
```

Arguments

`Y`	A functional time series.
`L`	Window length.

Value

An object of class fssa, which is a list of multivariate functional objects and the following components:

`values`	A numeric vector of eigenvalues.
`L`	Window length.
`N`	Length of time series.
`Y`	The original functional time series.

Examples

```
## Not run:
## Call Center Data
data("Callcenter")
require(fda)
require(Rfssa)
D <- matrix(sqrt(Callcenter$calls),nrow = 240)
N <- ncol(D)
time <- 1:N
K <- nrow(D)
u <- seq(0,K,length.out =K)
d <- 22 #Optimal Number of basises
basis <- create.bspline.basis(c(min(u),max(u)),d)
Ysmooth <- smooth.basis(u,D,basis)
Y <- Ysmooth$fd
## fssa decomposition
L <- 28
U <- fssa(Y,L)
plot(U,d=13)
plot(U,d=9,type="efunctions")
plot(U,d=9,type="efunctions2")
plot(U,d=9,type="vectors")
plot(U,d=10,type="meanvectors")
plot(U,d=10,type="paired")
plot(U,d=10,type="meanpaired")
plot(U,d=10,type="wcor")
## fssa reconstruction
gr <- list(1,2:3,4:5,6:7,8:20)
Q <- freconstruct(U, gr)

cols3 <- rainbow(N)

layout(matrix(c(1,1,2,3,4,5,6,6),nr=2))
par(mar=c(2,1,2,2))
plot(Y,lty=1,xlab="",main="Call Numbers(Observed)"
    ,ylab="",col=cols3)
    plot(Q[[1]],lty=1,xlab="",main="1st Component"
    ,ylab="",lwd=1,col=cols3)
    plot(Q[[2]],lty=1,xlab="",main="2nd Component"
    ,ylab="",lwd=1,col=cols3)
    plot(Q[[3]],lty=1,xlab="",main="3rd Component"
    ,ylab="",lwd=1,col=cols3)
    plot(Q[[4]],lty=1,xlab="",main="4th Component"
    ,ylab="",lwd=1,col=cols3)
    plot(Q[[5]],lty=1,xlab="",main="5th Component(Noise)"
    ,ylab="",lwd=1,col=cols3)


layout(matrix(c(1,1,2,3,4,5,6,6),nr=2))
par(mar=c(2,1,2,2))
ftsplot(u,time,Y,space = 0.2,type=3,ylab = "",xlab = "Day",main = "Call Numbers(Observed)")
```

```
ftsplot(u,time,Q[[1]],space = 0.2,type=3,ylab = "",xlab = "Day",main = "1st Component")
ftsplot(u,time,Q[[2]],space = 0.2,type=3,ylab = "",xlab = "Day",main = "2nd Component")
ftsplot(u,time,Q[[3]],space = 0.2,type=3,ylab = "",xlab = "Day",main = "3rd Component")
ftsplot(u,time,Q[[4]],space = 0.2,type=3,ylab = "",xlab = "Day",main = "4th Component")
ftsplot(u,time,Q[[5]],space = 0.2,type=3,ylab = "",xlab = "Day",main = "5th Component(Noise)")

## End(Not run)
```

---

ftsplot *Functional Time Series Plots*

Description

Novel different type of plots to visualize the functional time series data objects.

Usage

```
ftsplot(x, y, X, type = 2, zlab = NULL, xlab = NULL, ylab = NULL,
  space = 0.1, main = NULL)
```

Arguments

x	a numeric vector of coordinate x.
y	a numeric vector of coordinate y.
X	a functional time series object of class fd.
type	what type of plot should be drawn. Possible types are "type=1" for ribbon3D plot with curtain, "type=2" for ribbon3D plot without curtain, "type=3" for image2D plot.
zlab	The label of z axis.
xlab	The label of x axis.
ylab	The label of y axis.
space	The amount of space (as a fraction of the average ribbon width) left between ribbons.
main	The main title.

Examples

```
data("Callcenter")
library(fda)
D <- matrix(sqrt(Callcenter$calls),nrow = 240)
N <- ncol(D)
time <- 1:30
K <- nrow(D)
u <- seq(0,K,length.out =K)
d <- 22 #Optimal Number of basises
basis <- create.bspline.basis(c(min(u),max(u)),d)
```

```
Ysmooth <- smooth.basis(u,D,basis)
Y <- Ysmooth$fd

par(mar=c(2,1,2,2),mfrow=c(1,3))
ftsplot(u,time,Y[1:30],space = 0.4,type=1,ylab = "",xlab = "Day",main = "Typ1=1")
ftsplot(u,time,Y[1:30],space = 0.4,type=2,ylab = "",xlab = "Day",main = "Typ1=2")
ftsplot(u,time,Y[1:30],space = 0.4,type=3,ylab = "",xlab = "Day",main = "Typ1=3")
```

`fwcor` *W-correlation matrix*

Description

This function evaluate the Wcorrelation plot for fssa

Usage

```
fwcor(U, d)
```

Arguments

`U`	in the input is an object of class `fssa`.
`d`	is the number of elementary components in pairwise W-correlations matrix.

Value

A squared matrix of W-correlation values.

See Also

`fssa`

`plot.fssa` *Plotting fssa Objects*

Description

Plotting method for objects inheriting from class `fssa`.

Usage

```
## S3 method for class 'fssa'
plot(x, d = length(x$values), type = "values", ...)
```

Arguments

x
: a functional singular value decomposition object, time series objects, usually inheriting from class "fssa".

d
: an integer which is the number of elementary components in the plot.

type
: what type of plot should be drawn. Possible types are:

- `"values"` plot the square-root of singular values (default).
- `"paired"` plot the pairs of eigenfunction's coefficients. (useful for the detection of periodic components).
- `"wcor"` plot the W-correlation matrix for the reconstructed objects.
- `"vectors"` plot the eigenfunction's coefficients.(useful for the detection of period length).
- `"meanvectors"` plot the mean of eigenfunction's coefficients.(useful for the detection of period length).
- `"meanpaired"` plot the pairs of mean of eigenfunction's coefficients. (useful for the detection of periodic components).
- `"efunctions"` heatmap plot of eigenfunctions.(useful for the detection of period length).
- `"efunctions2"` plot the eigenfunctions.(useful for the detection of meaningful patterns).

...
: Arguments to be passed to methods, such as graphical parameters.

See Also

fssa, ftsplot

Examples

```
require(Rfssa)
require(fda)
n <- 50 # Number of points in each function.
d <- 9
N <- 60
sigma <- 0.5
set.seed(110)
E <- matrix(rnorm(N*d,0,sigma/sqrt(d)),ncol = N, nrow = d)
basis <- create.fourier.basis(c(0, 1), d)
Eps <- fd(E,basis)
om1 <- 1/10
om2 <- 1/4
f0 <- function(tau, t) 2*exp(-tau*t/10)
f1 <- function(tau, t) 0.2*exp(-tau^3) * cos(2 * pi * t * om1)
f2 <- function(tau, t) -0.2*exp(-tau^2) * cos(2 * pi * t * om2)
tau <- seq(0, 1, length = n)
t <- 1:N
f0_mat <- outer(tau, t, FUN = f0)
f0_fd <- smooth.basis(tau, f0_mat, basis)$fd
f1_mat <- outer(tau, t, FUN = f1)
f1_fd <- smooth.basis(tau, f1_mat, basis)$fd
```

```
f2_mat <- outer(tau, t, FUN = f2)
f2_fd <- smooth.basis(tau, f2_mat, basis)$fd
Y_fd <- f0_fd+f1_fd+f2_fd
L <-10
U <- fssa(Y_fd,L)
plot(U)
plot(U,d=4,type="efunctions")
plot(U,d=4,type="vectors")
plot(U,d=5,type="paired")
plot(U,d=5,type="wcor")
plot(U,d=5,type="meanvectors")
plot(U,d=5,type="efunctions2")
plot(U,d=5,type="meanpaired")
```

Rfssa *Rfssa: A package for functional singular spectrum analysis.*

Description

The Rfssa package provides the collections of necessary functions to implement Functional Singular Spectrum Analysis (FSSA) for analysing Functional Time Series (FTS). FSSA is a novel non-parametric method to perform decomposition and reconstruction of FTS.

Details

Typically the use of the package starts with the decomposition of the functional time series using `fssa`. Then, a suitable grouping of the elementary time series is required. This can be done heuristically, for example, via looking at the plots of the decomposition (`plot`). Alternatively, one can examine the so-called w-correlation matrix (`fwcor`). Next step includes the reconstruction of the time-series using the selected grouping (`freconstruct`).

Typically the use of the package starts with the decomposition of the functional time series using `fssa`. Then, a suitable grouping of the elementary time series is required. This can be done heuristically, for example, via looking at the plots of the decomposition (`plot`). Alternatively, one can examine the so-called w-correlation matrix (`fwcor`). Next step includes the reconstruction of the time-series using the selected grouping (`freconstruct`).

References

Haghbin H., Najibi, S.M., Mahmoudvand R., Maadooliat M. (2019). Functional singular spectrum Analysis. Manuscript submitted for publication.

See Also

`fssa`, `freconstruct`, `fwcor`, `wplot`, `ftsplot`

`fssa`, `freconstruct`, `fwcor`, `wplot`, `ftsplot`

`wplot` *Plot of W-correlations*

Description

Display W-correlation matrix of a single components.

Usage

```
wplot(W)
```

Arguments

`W`	a W-correlation matrix.

See Also

`fssa`

Index