Package ‘hdftsa’

May 8, 2026

Type Package

Title High-Dimensional Functional Time Series Analysis

Version 1.1

Date 2026-03-31

Depends R (>= 3.5.0), ftsa, forecast

Imports methods, MASS, pdfCluster

Encoding UTF-8

LazyLoad yes

LazyData yes

LazyDataCompression xz

ByteCompile TRUE

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

Description Offers methods for visualising, modelling, and forecasting high-dimensional functional time series, also known as functional panel data. Documentation about 'hdftsa' is initially provided via the paper by Cristian F. Jimenez-Varon, Ying Sun and Han Lin Shang (2024, Journal of Computational and Graphical Statistics).

License GPL-3

NeedsCompilation no

Author Han Lin Shang [aut, cre] (ORCID: <https://orcid.org/0000-0003-1769-6430>)

Repository CRAN

Date/Publication 2026-03-31 07:10:02 UTC

Contents

`hdftsa-package` *High-dimensional Functional Time Series Analysis*

Description

Offers methods for visualising, modelling, and forecasting high-dimensional functional time series, also known as functional panel data. Documentation for 'hdftsa' is provided in the paper by Cristian F. Jimenez-Varon, Ying Sun, and Han Lin Shang (2024, Journal of Computational and Graphical Statistics).

Author(s)

Han Lin Shang [aut, cre] (ORCID: <https://orcid.org/0000-0003-1769-6430>)

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

References

D. Li, R. Li and H. L. Shang (2024) Detection and estimation of structural breaks in high-dimensional functional time series, *Annals of Statistics*, **52**(4), 1716-1740.

C. F. Jimenez-Varon, Y. Sun and H. L. Shang (2024) Forecasting high-dimensional functional time series: Application to sub-national age-specific mortality, *Journal of Computational and Graphical Statistics*, **33**(4), 1160-1174.

C. F. Jimenez-Varon, Y. Sun and H. L. Shang (2025) Forecasting density-valued functional panel data, *Australian and New Zealand Journal of Statistics*, **67**(3), 401-415.

H. Wang, T. Guan and H. L. Shang (2025) Interpretable additive model for analyzing functional panel data, *Journal of Multivariate Analysis*, **in press**.

H. L. Shang (2025) Forecasting a time series of Lorenz curves: one-way functional analysis of variance, *Journal of Applied Statistics*, **52**(15), 2924-2940.

C. Tang, H. L. Shang, Y. Yang and Y. Yang (2025) Forecasting high-dimensional functional time series with dual-factor structures, *Journal of the Royal Statistical Society: Series A*, **in press**.

C. Leng, D. Li, H. L. Shang and Y. Xia (2026) Covariance function estimation for high-dimensional functional time series with dual factor structures, *Journal of Business and Economic Statistics*, **in press**.

H. L. Shang (2026) Conformal prediction for high-dimensional functional time series: Applications to subnational mortality. https://arxiv.org/abs/2603.10674.

H. L. Shang and C. F. Jimenez-Varon (2026) Interpretable models for forecasting high-dimensional functional time series.

`all_hmd_female_data` *The US female log-mortality rate from 1959-2020 and 3 states (New York, California, Illinois).*

Description

We generate for the female population in the US. The functional time series corresponding to the log mortality data in each of the 3 states. Each functional time series comprises the ages from 0 to 100+.

Usage

```
data("all_hmd_male_data")
```

Format

A n x p matrix with n=186 observations on the following p=101 ages from 0 to 100+.

Details

The data generated corresponds to the FTS for the female US log-mortality. The matrix contains 186 FTS stacked by rows. They correspond to 62 (number of years) times 3 (states). Each FTS contains 101 functional values.

References

United States Mortality Database (2023). University of California, Berkeley (USA). Department of Demography at the University of California, Berkeley. Available at usa.mortality.org (data downloaded on March 15, 2023).

Examples

```
data(all_hmd_male_data)
```

`all_hmd_male_data` *The US male log-mortality rate from 1959-2020 and 3 states (New York, California, Illinois).*

Description

We generate for the male population in the US. The functional time series corresponding to the log mortality data in each of the 3 states. Each functional time series comprises the ages from 0 to 100+.

Usage

```
data("all_hmd_male_data")
```

Format

A n x p matrix with n=186 observations on the following p=101 ages from 0 to 100+.

Details

The data generated corresponds to the FTS for the male US log-mortality. The matrix contains 186 FTS stacked by rows. They correspond to 62 (number of years) times 3 (states). Each FTS contains 101 functional values.

References

United States Mortality Database (2023). University of California, Berkeley (USA). Department of Demography at the University of California, Berkeley. Available at usa.mortality.org (data downloaded on March 15, 2023).

Examples

```
data(all_hmd_male_data)
```

`dmfpca` *Dynamic multilevel functional principal component analysis*

Description

Functional principal component analysis is used to decompose multiple functional time series. This function uses a functional panel data model to reduce dimensions for multiple functional time series.

Usage

```
dmfpca(y, M = NULL, J = NULL, N = NULL, tstart = 0, tlength = 1)
```

Arguments

`y`	A data matrix containing functional responses. Each row contains measurements from a function at a set of grid points, and each column contains measurements of all functions at a particular grid point
`M`	Number of `fts` obejcts
`J`	Number of functions in each object
`N`	Number of grid points per function
`tstart`	Start point of the grid points
`tlength`	Length of the interval that the functions are evaluated at

Value

`K1`	Number of components for the common time-trend
`K2`	Number of components for the residual component
`lambda1`	A vector containing all common time-trend eigenvalues in non-increasing order
`lambda2`	A vector containing all residual component eigenvalues in non-increasing order
`phi1`	A matrix containing all common time-trend eigenfunctions. Each row contains an eigenfunction evaluated at the same set of grid points as the input data. The eigenfunctions are in the same order as the corresponding eigenvalues
`phi2`	A matrix containing all residual component eigenfunctions. Each row contains an eigenfunction evaluated at the same set of grid points as the input data. The eigenfunctions are in the same order as the corresponding eigenvalues.
`scores1`	A matrix containing estimated common time-trend principal component scores. Each row corresponds to the common time-trend scores for a particular subject in a cluster. The number of rows is the same as that of the input matrix y. Each column contains the scores for a common time-trend component for all subjects.
`scores2`	A matrix containing estimated residual component principal component scores. Each row corresponds to the level 2 scores for a particular subject in a cluster. The number of rows is the same as that of the input matrix y. Each column contains the scores for a residual component for all subjects.
`mu`	A vector containing the overall mean function.
`eta`	A matrix containing the deviation from the overall mean function to the country-specific mean function. The number of rows is the number of countries.

Author(s)

Chen Tang and Han Lin Shang

References

Rice, G. and Shang, H. L. (2017) "A plug-in bandwidth selection procedure for long-run covariance estimation with stationary functional time series", *Journal of Time Series Analysis*, **38**, 591-609.

Shang, H. L. (2016) "Mortality and life expectancy forecasting for a group of populations in developed countries: A multilevel functional data method", *The Annals of Applied Statistics*, **10**, 1639-1672.

Di, C.-Z., Crainiceanu, C. M., Caffo, B. S. and Punjabi, N. M. (2009) "Multilevel functional principal component analysis", *The Annals of Applied Statistics*, **3**, 458-488.

See Also

mftsc

Examples

```
## The following takes about 10 seconds to run ##
## Not run:
y <- do.call(rbind, sim_ex_cluster)
MFPCA.sim <- dmfpca(y, M = length(sim_ex_cluster), J = nrow(sim_ex_cluster[[1]]),
    N = ncol(sim_ex_cluster[[1]]), tlength = 1)

## End(Not run)
```

`forecast.hdfpca` *Forecasting via a high-dimensional functional principal component regression*

Description

Forecast a high-dimensional functional principal component model.

Usage

```
## S3 method for class 'hdfpca'
forecast(object, h = 3, level = 80, B = 50, ...)
```

Arguments

`object`	An object of class 'hdfpca'
`h`	Forecast horizon
`level`	Prediction interval level, the default is 80 percent
`B`	Number of bootstrap replications
`...`	Other arguments passed to forecast routine.

Details

The low-dimensional factors are forecasted separately using autoregressive integrated moving-average (ARIMA) models. The forecast functions are then calculated using the forecast factors. Bootstrap prediction intervals are constructed by resampling from the forecast residuals of the ARIMA models.

Value

forecast	A list containing the h-step-ahead forecast functions for each population
upper	Upper confidence bound for each population
lower	Lower confidence bound for each population

Author(s)

Y. Gao and H. L. Shang

References

Y. Gao, H. L. Shang and Y. Yang (2018) High-dimensional functional time series forecasting: An application to age-specific mortality rates, *Journal of Multivariate Analysis*, **forthcoming**.

See Also

hdfpca, hd_data

Examples

```
## Not run:
hd_model = hdfpca(hd_data, order = 2, r = 2)
hd_model_fore = forecast.hdfpca(object = hd_model, h = 1)

## End(Not run)
```

---

hdfpca — *High-dimensional functional principal component analysis*

Description

Fit a high-dimensional functional principal component analysis model to a multiple-population of functional time series data.

Usage

```
hdfpca(y, order, q = sqrt(dim(y[[1]])[2]), r)
```

Arguments

y	A list, where each item is a population of functional time series. Each item is a data matrix of dimension p by n, where p is the number of discrete points in each function and n is the sample size
order	The number of principal component scores to retain in the first step dimension reduction
q	The tuning parameter used in the first step of dimension reduction, by default it is equal to the square root of the sample size
r	The number of factors to retain in the second step dimension reduction

Details

In the first step, dynamic functional principal component analysis is performed on each population, and then in the second step, factor models are fitted to the resulting principal component scores. The high-dimensional functional time series are thus reduced to low-dimensional factors.

Value

`y`	The input data
`p`	The number of discrete points in each function
`fitted`	A list containing the fitted functions for each population
`m`	The number of populations
`model`	Model 1 includes the first step dynamic functional principal component analysis models, model 2 includes the second step high-dimensional principal component analysis models
`order`	Input order
`r`	Input r

Author(s)

Y. Gao and H. L. Shang

References

Y. Gao, H. L. Shang and Y. Yang (2019) High-dimensional functional time series forecasting: An application to age-specific mortality rates, *Journal of Multivariate Analysis*, **170**, 232-243.

See Also

`forecast.hdfpca`, `hd_data`

Examples

```
hd_model = hdfpca(hd_data, order = 2, r = 2)
```

`hd_data` *Simulated high-dimensional functional time series*

Description

We generate N populations of functional time series. For each $i \in \{1, \dots, N\}$, the ith function at time $t \in \{1, \dots, T\}$ is given by

$$X_t^{(i)}(u) = \sum_{p=1}^{2} \beta_{p,t}^{(i)} \gamma_p^{(i)}(u) + \theta_t^{(i)}(u),$$

where $\theta_t^{(i)}(u) = \sum_{p=3}^{\infty} \beta_{p,t}^{(i)} \gamma_p^{(i)}(u)$.

Usage

```
data("hd_data")
```

Details

The coefficients $\beta_{p,t}^{(i)}$ for all N populations are combined and generated, for all $p \in N$, by

$$\boldsymbol{\beta}_{p,t} = \boldsymbol{A}_p \boldsymbol{f}_{p,t},$$

where $\boldsymbol{\beta}_{p,t} = \{\beta_{p,t}^1, \dots, \beta_{p,t}^N\}$. Here, $\boldsymbol{A}_p$ is an $N \times N$ matrix, and $\boldsymbol{f}_{p,t}$ is an $N \times 1$ vector. Furthermore, we assume that the $\beta_{p,t}^{(i)}$s have mean 0 and variance 0 when $p > 3$, so we only construct the coefficients $\boldsymbol{\beta}_{p,t}$ for $p \in \{1, 2, 3\}$.

The first set of coefficients $\boldsymbol{\beta}_{1,t}$ for N populations are generated with $\boldsymbol{\beta}_{1,t} = \boldsymbol{A}_1 \boldsymbol{f}_{1,t}$. Each element in the matrix $\boldsymbol{A}_1$ is generated by $a_{ij} = N^{-1/4} \times b_{ij}$, where $b_{ij} \sim N(2, 4)$.

The factors $\boldsymbol{f}_{1,t}$ are generated using an autoregressive model of order 1, i.e., AR(1). Define the ith element in vector $\boldsymbol{f}_{1,t}$ as $f_{1,t}^{(i)}$. Then, $f_{1,t}^1$ is generated by $f_{1,t}^1 = 0.5 \times f_{1,t-1}^1 + \omega_t$, where ω_t are independent $N(0, 1)$ random variables. We generate $f_{1,t}^{(i)}$ for all $i \in \{2, \dots, N\}$ by $f_{1,t}^{(i)} = (1/N) \times g_t^{(i)}$, where $g_t^{(2)}, \dots, g_t^{(N)}$ are also AR(1) and follow $g_t^{(i)} = 0.2 \times g_{t-1}^{(i)} + \omega_t$. It is then ensured that most of the variance of $\boldsymbol{\beta}_{1,t}$ can be explained by one factor. The second coefficient $\boldsymbol{\beta}_{2,t}$ are constructed the same way as $\boldsymbol{\beta}_{1,t}$.

We also generate the third functional principal component scores $\boldsymbol{\beta}_{3,t}$ but with small values. Moreover, $\boldsymbol{A}_3$ is generated by $a_{ij} = N^{-1/4} \times b_{ij}$, where $b_{ij} \sim N(0, 0.04)$. The factors $bmf_{3,t}$ are generated as $\boldsymbol{f}_{1,t}$.

The three basis functions are constructed by $\gamma_1^{(i)}(u) = \sin(2\pi u + \pi i/2)$, $\gamma_2^{(i)}(u) = \cos(2\pi u + \pi i/2)$ and $\gamma_3^{(i)}(u) = \sin(4\pi u + \pi i/2)$, where $u \in [0, 1]$. Finally, the functional time series for the ith population is constructed by

$$\boldsymbol{X}_t^{(i)}(u) = \boldsymbol{\beta}_{1,t}\gamma_1^{(i)}(u) + \boldsymbol{\beta}_{2,t}\gamma_2^{(i)}(u) + \boldsymbol{\beta}_{3,t}\gamma_3^{(i)}(u),$$

where $(\cdot)_i$ denotes the ith element of the vector.

References

Y. Gao, H. L. Shang and Y. Yang (2019) High-dimensional functional time series forecasting: An application to age-specific mortality rates, *Journal of Multivariate Analysis*, **170**, 232-243.

See Also

hdfpca, forecast.hdfpca

Examples

```
data(hd_data)
```

MFPCA *Multilevel functional principal component analysis for clustering*

Description

A multilevel functional principal component analysis for performing clustering analysis

Usage

```
MFPCA(y, M = NULL, J = NULL, N = NULL)
```

Arguments

`y`	A data matrix containing functional responses. Each row contains measurements from a function at a set of grid points, and each column contains measurements of all functions at a particular grid point
`M`	Number of countries
`J`	Number of functional responses in each country
`N`	Number of grid points per function

Value

`K1`	Number of components at level 1
`K2`	Number of components at level 2
`K3`	Number of components at level 3
`lambda1`	A vector containing all level 1 eigenvalues in non-increasing order
`lambda2`	A vector containing all level 2 eigenvalues in non-increasing order
`lambda3`	A vector containing all level 3 eigenvalues in non-increasing order
`phi1`	A matrix containing all level 1 eigenfunctions. Each row contains an eigenfunction evaluated at the same set of grid points as the input data. The eigenfunctions are in the same order as the corresponding eigenvalues
`phi2`	A matrix containing all level 2 eigenfunctions. Each row contains an eigenfunction evaluated at the same set of grid points as the input data. The eigenfunctions are in the same order as the corresponding eigenvalues
`phi3`	A matrix containing all level 3 eigenfunctions. Each row contains an eigenfunction evaluated at the same set of grid points as the input data. The eigenfunctions are in the same order as the corresponding eigenvalues
`scores1`	A matrix containing estimated level 1 principal component scores. Each row corresponds to the level 1 scores for a particular subject in a cluster. The number of rows is the same as that of the input matrix `y`. Each column contains the scores for a level 1 component for all subjects

`scores2`	A matrix containing estimated level 2 principal component scores. Each row corresponds to the level 2 scores for a particular subject in a cluster. The number of rows is the same as that of the input matrix `y`. Each column contains the scores for a level 2 component for all subjects.
`scores3`	A matrix containing estimated level 3 principal component scores. Each row corresponds to the level 3 scores for a particular subject in a cluster. The number of rows is the same as that of the input matrix `y`. Each column contains the scores for a level 3 component for all subjects.
`mu`	A vector containing the overall mean function
`eta`	A matrix containing the deviation from overall mean function to country-specific mean function. The number of rows is the number of countries
`Rj`	Common trend
`Uij`	Country-specific mean function

Author(s)

Chen Tang, Yanrong Yang and Han Lin Shang

References

T. Chen, H. L. Shang, Y. Yang and Y. Yang (2026) Forecasting high-dimensional functional time series with dual-factor structures, *Journal of the Royal Statistical Society: Series A*, **in press**.

See Also

`mftsc`

`mftsc` *Multiple functional time series clustering*

Description

Clustering the multiple functional time series. The function uses the functional panel data model to cluster different time series into subgroups

Usage

```
mftsc(X, alpha)
```

Arguments

`X`	A list of sets of smoothed functional time series to be clustered, for each object, it is a p x q matrix, where p is the sample size and q is the number of grid points of the function
`alpha`	A value input for adjusted rand index to measure similarity of the memberships with last iteration, can be any value big than 0.9

Details

As an initial step, conventional k-means clustering is performed on the dynamic FPC scores, then an iterative membership updating process is applied by fitting the MFPCA model.

Value

`iteration`	the number of iterations until convergence
`memebership`	a list of all the membership matrices at each iteration
`member.final`	the final membership

Author(s)

Chen Tang, Yanrong Yang and Han Lin Shang

References

T. Chen, H. L. Shang, Y. Yang and Y. Yang (2026) Forecasting high-dimensional functional time series with dual-factor structures, *Journal of the Royal Statistical Society: Series A*, **in press**.

See Also

MFPCA

Examples

```
## Not run:
data(sim_ex_cluster)
cluster_result<-mftsc(X=sim_ex_cluster, alpha=0.99)
cluster_result$member.final

## End(Not run)
```

`One_way_mean` *One-way functional analysis of variance based on means*

Description

Decomposition by one-way functional analysis of variance based on means.

Usage

```
One_way_mean(data_pop1, year = 1959:2020, age = 0:100, n_prefectures = 51)
```

Arguments

data_pop1	The multivariate functional data, which are a matrix with dimension n by 2p, where n is the sample size, and p is the dimensionality.
year	Vector with the years considered in each population.
age	Vector with the ages considered in each year.
n_prefectures	Number of prefectures.

Value

GE_mean	Grand_effect, a vector of dimension p.
FRE_mean	Row_effect, a matrix of dimension length(row_partition_index) by p.
Deterministic	Deterministic component = Grand effect + Row effect

Author(s)

Cristian Felipe Jimenez Varon, Han Lin Shang

References

C. F. Jimenez Varon, Y. Sun and H. L. Shang (2024) “Forecasting high-dimensional functional time series: Application to sub-national age-specific mortality", *Journal of Computational and Graphical Statistics*, **33**(4), 1160-1174.

H. L. Shang (2025) “Forecasting a time series of Lorenz curves: One-way functional analysis of variance", *Journal of Applied Statistics*, **52**(15), 2924-2940.

See Also

One_way_mean_residuals, One_way_median_polish, One_way_median_polish_residuals

Examples

```
# The US mortality data  1959-2020, for one population (female)
# and 3 states (New York, California, Illinois)
# first define the parameters and the row partitions.
# Define some parameters.
year = 1959:2020
age = 0:100
n_prefectures = 3

#Load the US data. Make sure it is a matrix.
Y <-  all_hmd_female_data
FMP <- One_way_mean(t(Y), year=1959:2020, age=0:100, n_prefectures=3)
```

One_way_mean_residuals

High-dimensional functional time series decomposition into deterministic and functional residual components.

Description

Decomposition of high-dimensional functional time series into deterministic and functional residuals

Usage

```
One_way_mean_residuals(data_pop1, n_prefectures, n_year, n_age)
```

Arguments

data_pop1	The multivariate functional data, which is a matrix with dimension n by 2p, where n is the sample size, and p is the dimensionality.
n_prefectures	Number of prefectures.
n_year	Vector with the years considered in each population.
n_age	Vector with the ages considered in each year.

Value

Residuals	Residual component
Reconstructed_Data	Reconstructed data
Reconstruction_OK	Indicator if reconstruction equals original data

Note

One_way_mean

Author(s)

Cristian Felipe Jimenez Varon and Han Lin Shang

References

C. F. Jimenez Varon, Y. Sun and H. L. Shang (2024) "Forecasting high-dimensional functional time series: Application to sub-national age-specific mortality", *Journal of Computational and Graphical Statistics*, **33**(4), 1160-1174.

Examples

```
# The US mortality data  1959-2020, for one population (female)
# and 3 states (New York, California, Illinois)
# first define the parameters and the row partitions.
# Define some parameters.
year = 1959:2020
age = 0:100
n_prefectures = 3

#Load the US data. Make sure it is a matrix.
Y <- all_hmd_female_data
# The results
# Compute the functional residuals.
FANOVA_mean_residuals <- One_way_mean_residuals(t(Y), n_prefectures=3,
n_year=length(year), n_age=length(age))
```

One_way_median_polish *One-way functional median polish from Sun and Genton (2012)*

Description

Decomposition by one-way functional median polish.

Usage

```
One_way_median_polish(Y, n_prefectures=51, year=1959:2020, age=0:100)
```

Arguments

Y	The multivariate functional data, which are a matrix with dimension n by 2p, where n is the sample size, and p is the dimensionality.
year	Vector with the years considered in each population.
n_prefectures	Number of prefectures.
age	Vector with the ages considered in each year.

Value

grand_effect	Grand_effect, a vector of dimension p.
row_effect	Row_effect, a matrix of dimension length(row_partition_index) by p.

Author(s)

Cristian Felipe Jimenez Varon, Ying Sun, Han Lin Shang

References

C. F. Jimenez Varon, Y. Sun and H. L. Shang (2024) “Forecasting high-dimensional functional time series: Application to sub-national age-specific mortality", *Journal of Computational and Graphical Statistics*, **33**(4), 1160-1174. \ Sun, Ying, and Marc G. Genton (2012) “Functional Median Polish", Journal of Agricultural, Biological, and Environmental Statistics 17(3), 354-376.

See Also

One_way_median_polish_residuals, Two_way_median_polish, Two_way_median_polish_residuals

Examples

```
# The US mortality data  1959-2020, for one population (female)
# and 3 states (New York, California, Illinois)
# first define the parameters and the row partitions.
# Define some parameters.
year = 1959:2020
age = 0:100
n_prefectures = 3

#Load the US data. Make sure it is a matrix.
Y <-  all_hmd_female_data
# Compute the functional median polish decomposition.
FMP <- One_way_median_polish(Y,n_prefectures=3,year=1959:2020,age=0:100)
# The results
##1. The functional grand effect
FGE <- FMP$grand_effect
##2. The functional row effect
FRE <- FMP$row_effect
```

One_way_median_polish_residuals

High-dimensional functional time series decomposition into deterministic (from functional median polish of Sun and Genton (2012)), and functional residual components.

Description

Decomposition of high-dimensional functional time series into deterministic (from functional median polish), and functional residuals

Usage

```
One_way_median_polish_residuals(Y, n_prefectures = 51, year = 1959:2020, age = 0:100)
```

Arguments

Y	The multivariate functional data, which is a matrix with dimension n by 2p, where n is the sample size, and p is the dimensionality.
n_prefectures	Number of prefectures.
year	Vector with the years considered in each population.
age	Vector with the ages considered in each year.

Value

A matrix of dimension n by p.

Author(s)

Cristian Felipe Jimenez Varon, Ying Sun, Han Lin Shang

References

C. F. Jimenez Varon, Y. Sun and H. L. Shang (2024) “Forecasting high-dimensional functional time series: Application to sub-national age-specific mortality", *Journal of Computational and Graphical Statistics*, **33**(4), 1160-1174. \ Y. Sun and M. G. Genton (2012) “Functional median polish", Journal of Agricultural, Biological, and Environmental Statistics, 17(3), 354-376.

See Also

One_way_median_polish, One_way_mean, One_way_mean_residuals

Examples

```
# The US mortality data  1959-2020, for one population (female)
# and 3 states (New York, California, Illinois)
# first define the parameters and the row partitions.
# Define some parameters.
year = 1959:2020
age = 0:100
n_prefectures = 3

#Load the US data. Make sure it is a matrix.
Y <- all_hmd_female_data
# The results
# Compute the functional residuals.
FMP_residuals <- One_way_median_polish_residuals(Y, n_prefectures=3, year=1959:2020, age=0:100)
```

sim_ex_cluster *Simulated multiple sets of functional time series*

Description

We generate 2 groups of m functional time series. For each i in {1, ..., m} in a given cluster c, c in {1,2}, the t th function, t in {1,..., T}, is given by

$$Y_{it}^{(c)}(x) = \mu^{(c)}(x) + \sum_{k=1}^{2} \xi_{tk}^{(c)} \rho_k^{(c)}(x) + \sum_{l=1}^{2} \zeta_{itl}^{(c)} \psi_l^{(c)}(x) + \upsilon_{it}^{(c)}(x)$$

Usage

```
data("sim_ex_cluster")
```

Details

The mean functions for each of these two clusters are set to be $\mu^{(1)}(x) = 2(x-0.25)^2$ and $\mu^{(2)}(x) = 2(x-0.4)^2 + 0.1$.

While the variates $\xi_{\mathbf{tk}}^{(\mathbf{c})} = (\xi_{1k}^{(c)}, \xi_{2k}^{(c)}, \ldots, \xi_{Tk}^{(c)})^\top$ for both clusters, are generated from autoregressive of order 1 with parameter 0.7, while the variates $\zeta_{it1}^{(c)}$ and $\zeta_{it2}^{(c)}$ for both clusters, are generated from independent and identically distributed $N(0, 0.5)$ and $N(0, 0.25)$, respectively.

The basis functions for the common-time trend for the first cluster, $\rho_k^{(1)}(x)$, for k in {1,2} are $sqrt(2) * sin(\pi * (0 : 200/200))$ and $sqrt(2) * cos(\pi * (0 : 200/200))$ respectively; and the basis functions for the common-time trend for the second cluster, $\rho_k^{(2)}(x)$, for k in {1,2} are $sqrt(2) * sin(2\pi * (0 : 200/200))$ and $sqrt(2) * cos(2\pi * (0 : 200/200))$ respectively.

The basis functions for the residual for the first cluster, $\psi_l^{(1)}(x)$, for l in {1,2} are $sqrt(2) * sin(3\pi * (0 : 200/200))$ and $sqrt(2) * cos(3\pi * (0 : 200/200))$ respectively; and the basis functions for the residual for the second cluster, $\psi_l^{(2)}(x)$, for l in {1,2} are $sqrt(2) * sin(4\pi * (0 : 200/200))$ and $sqrt(2) * cos(4\pi * (0 : 200/200))$ respectively.

The measurement error υ_{it} for each continuum x is generated from independent and identically distributed $N(0, 0.2^2)$

References

T. Chen, H. L. Shang, Y. Yang and Y. Yang (2026) Forecasting high-dimensional functional time series with dual-factor structures, *Journal of the Royal Statistical Society: Series A*, **in press**.

Examples

```
data(sim_ex_cluster)
```

Two_way_mean *Functional analysis of variance fitted by means.*

Description

Decomposition by functional analysis of variance fitted by means.

Usage

```
Two_way_mean(data_pop1, data_pop2, year=1959:2020, age= 0:100,
   n_prefectures=51, n_populations=2)
```

Arguments

data_pop1	It's a p by n matrix
data_pop2	It's a p by n matrix
year	Vector with the years considered in each population.
n_prefectures	Number of prefectures
age	Vector with the ages considered in each year.
n_populations	Number of populations.

Value

FGE_mean	FGE_mean, a vector of dimension p
FRE_mean	FRE_mean, a matrix of dimension length(row_partition_index) by p.
FCE_mean	FCE_mean, a matrix of dimension length(column_partition_index) by p.

Author(s)

Cristian Felipe Jimenez Varon, Ying Sun, Han Lin Shang

References

C. F. Jimenez Varon, Y. Sun and H. L. Shang (2023) “Forecasting high-dimensional functional time series: Application to sub-national age-specific mortality".

Ramsay, J. and B. Silverman (2006). Functional Data Analysis. Springer Series in Statistics. Chapter 13. New York: Springer

See Also

Two_way_median_polish, Two_way_median_polish_residuals

Examples

```
# The US mortality data  1959-2020 for two populations and three states
# (New York, California, Illinois)
# Compute the functional ANOVA decomposition fitted by means.
FANOVA_means <- Two_way_mean(data_pop1 = t(all_hmd_male_data),
                 data_pop2 = t(all_hmd_female_data),
                 year = 1959:2020, age =  0:100,
                 n_prefectures = 3, n_populations = 2)

##1. The functional grand effect
FGE = FANOVA_means$FGE_mean
##2. The functional row effect
FRE = FANOVA_means$FRE_mean
##3. The functional column effect
FCE = FANOVA_means$FCE_mean
```

Two_way_mean_residuals   *Functional time series decomposition into deterministic (functional analysis of variance fitted by means), and time-varying components (functional residuals).*

Description

Decomposition of functional time series into deterministic (by functional analysis of variance fitted by means), and time-varying components (functional residuals)

Usage

```
Two_way_mean_residuals(data_pop1, data_pop2, year, age, n_prefectures, n_populations)
```

Arguments

data_pop1	A p by n matrix
data_pop2	A p by n matrix
year	Vector with the years considered in each population.
n_prefectures	Number of prefectures
age	Vector with the ages considered in each year.
n_populations	Number of populations.

Value

residuals1	A matrix with dimension n by p.
residuals2	A matrix with dimension n by p.
rd	A two-dimensional logic vector proving that the decomposition sums up the data.

R	A matrix of dimension as n by 2p. This represents the time-varying component in the decomposition.
Fixed_comp	A matrix of dimension as n by 2p. This represents the deterministic component in the decomposition.

Author(s)

Cristian Felipe Jimenez Varon, Ying Sun, Han Lin Shang

References

C. F. Jimenez Varon, Y. Sun and H. L. Shang (2023) "Forecasting high-dimensional functional time series: Application to sub-national age-specific mortality".

Ramsay, J. and B. Silverman (2006). Functional Data Analysis. Springer Series in Statistics. Chapter 13. New York: Springer.

See Also

Two_way_median_polish_residuals

Examples

```
# The US mortality data  1959-2020, for two populations
# and three states (New York, California, Illinois)
# Compute the functional ANOVA decomposition fitted by means.
FANOVA_means_residuals <- Two_way_mean_residuals(data_pop1=t(all_hmd_male_data),
                            data_pop2=t(all_hmd_female_data), year = 1959:2020,
                            age = 0:100, n_prefectures = 3, n_populations = 2)

# The results
##1. The functional residuals from population 1
Residuals_pop_1=FANOVA_means_residuals$residuals1
##2. The functional residuals from population 2
Residuals_pop_2=FANOVA_means_residuals$residuals2
##3. A logic vector whose components indicate whether the sum of deterministic
##  and time-varying components recovers the original FTS.
Construct_data=FANOVA_means_residuals$rd
##4. Time-varying components for all the populations. The functional residuals
All_pop_functional_residuals <- FANOVA_means_residuals$R
##5. The deterministic components from the functional ANOVA decomposition
deterministic_comp <- FANOVA_means_residuals$Fixed_comp
```

Two_way_median_polish *Two-way functional median polish from Sun and Genton (2012)*

Description

Decomposition by two-way functional median polish

Usage

```
Two_way_median_polish(Y, year=1959:2020, age=0:100, n_prefectures=51, n_populations=2)
```

Arguments

`Y`	A matrix with dimension n by 2p. The functional data.
`year`	Vector with the years considered in each population.
`n_prefectures`	Number of prefectures
`age`	Vector with the ages considered in each year.
`n_populations`	Number of populations.

Value

`grand_effect`	grand_effect, a vector of dimension p
`row_effect`	row_effect, a matrix of dimension length(row_partition_index) by p.
`col_effect`	col_effect, a matrix of dimension length(column_partition_index) by p

Author(s)

Cristian Felipe Jimenez Varon, Ying Sun, Han Lin Shang

References

C. F. Jimenez Varon, Y. Sun and H. L. Shang (2023) “Forecasting high-dimensional functional time series: Application to sub-national age-specific mortality".

Sun, Ying, and Marc G. Genton (2012) “Functional Median Polish", Journal of Agricultural, Biological, and Environmental Statistics, 17(3), 354-376.

See Also

`Two_way_mean`

Examples

```
# The US mortality data  1959-2020 for two populations and three states
# (New York, California, Illinois)
# Compute the functional median polish decomposition.
FMP = Two_way_median_polish(cbind(all_hmd_male_data, all_hmd_female_data),
n_prefectures = 3, year = 1959:2020, age = 0:100, n_populations = 2)

##1. The functional grand effect
FGE = FMP$grand_effect
##2. The functional row effect
FRE = FMP$row_effect
##3. The functional column effect
FCE = FMP$col_effect
```

`Two_way_median_polish_residuals`
Functional time series decomposition into deterministic (from functional median polish from Sun and Genton (2012)), and time-varying components (functional residuals).

Description

Decomposition of functional time series into deterministic (from functional median polish), and time-varying components (functional residuals)

Usage

```
Two_way_median_polish_residuals(Y, n_prefectures, year, age, n_populations)
```

Arguments

`Y`	A matrix with dimension n by 2p. The functional data
`year`	Vector with the years considered in each population
`n_prefectures`	Number of prefectures
`age`	Vector with the ages considered in each year
`n_populations`	Number of populations

Value

`residuals1`	A matrix with dimension n by p
`residuals2`	A matrix with dimension n by p
`rd`	A two-dimensional logic vector that proves that the decomposition sums up to the data
`R`	A matrix with the same dimension as Y. This represent the time-varying component in the decomposition
`Fixed_comp`	A matrix with the same dimension as Y. This represent the deterministic component in the decomposition

Author(s)

Cristian Felipe Jimenez Varon, Ying Sun, Han Lin Shang

References

C. F. Jimenez Varon, Y. Sun and H. L. Shang (2023) "Forecasting high-dimensional functional time series: Application to sub-national age-specific mortality".

Sun, Ying, and Marc G. Genton (2012). "Functional Median Polish". Journal of Agricultural, Biological, and Environmental Statistics 17(3), 354-376.

See Also

Two_way_mean_residuals

Examples

```
# The US mortality data  1959-2020, for two populations
# and three states (New York, California, Illinois)
# Column binds the data from both populations
Y = cbind(all_hmd_male_data, all_hmd_female_data)
# Decompose FTS into deterministic (from functional median polish)
# and time-varying components (functional residuals).
FMP_residuals <- Two_way_median_polish_residuals(Y,n_prefectures=3,year=1959:2020,
                                   age=0:100,n_populations=2)
# The results
##1. The functional residuals from population 1
Residuals_pop_1=FMP_residuals$residuals1
##2. The functional residuals from population 2
Residuals_pop_2=FMP_residuals$residuals2
##3. A logic vector whose components indicate whether the sum of deterministic
##   and time-varying components recover the original FTS.
Construct_data=FMP_residuals$rd
##4. Time-varying components for all the populations. The functional residuals
All_pop_functional_residuals <- FMP_residuals$R
##5. The deterministic components from the functional median polish decomposition
deterministic_comp <- FMP_residuals$Fixed_comp
```

Index