Package 'SLBDD'

April 27, 2022

Type Package

Title Statistical Learning for Big Dependent Data

Version 0.0.4

Maintainer Antonio Elias <antonioefz91@gmail.com>

Description Programs for analyzing large-scale time series data. They include functions for automatic specification and estimation of univariate time series, for clustering time series, for multivariate outlier detections, for quantile plotting of many time series, for dynamic factor models and for creating input data for deep learning programs. Examples of using the package can be found in the Wiley book 'Statistical Learning with Big Dependent Data' by Daniel Peña and Ruey S. Tsay (2021). ISBN 9781119417385.

License GPL-3

Encoding UTF-8

LazyData true

LazyDataCompression xz

RoxygenNote 7.1.1

Depends R (>= 3.5.0)

Imports stats, glmnet, corpcor, forecast, gsarima, cluster, fGarch, imputeTS, methods, MASS, MTS, TSclust, tsoutliers, Matrix, matrixcalc, rnn

NeedsCompilation no

Author Angela Caro [aut],
Antonio Elias [aut, cre],
Daniel Peña [aut],
Ruey S. Tsay [aut]

Repository CRAN

Date/Publication 2022-04-27 11:20:05 UTC

R topics documented:

`arimaID` *Automatic Modeling of a Scalar Time Series*

Description

Automatic selection and estimation of a regular or possibly seasonal ARIMA model for a given time series.

Usage

```
arimaID(
  zt,
  maxorder = c(5, 1, 3),
  criterion = "bic",
  period = c(12),
  output = TRUE,
  method = "CSS-ML",
  pv = 0.01,
  spv = 0.01,
  transpv = 0.05,
  nblock = 0
)
```

Arguments

`zt`	T by 1 vector of an observed scalar time series without any missing values.
`maxorder`	Maximum order of (p, d, q) where p is the AR order, d the degree of differencing, and q the MA order. Default value is (5,1,4).
`criterion`	Information criterion used for model selection. Either AIC or BIC. Default is "bic".
`period`	Seasonal period. Default value is 12.
`output`	If TRUE it returns the differencing order, the selected order and the minimum value of the criterion. Default is TRUE.
`method`	Estimation method. See the arima command in R. Possible values are "CSS-ML", "ML", and "CSS". Default is "CSS-ML".
`pv`	P-value for unit-root test. Default value is 0.01.
`spv`	P-value for detecting seasonality. Default value is 0.01.
`transpv`	P-value for checking non-linear transformation. Default value is 0.05.
`nblock`	Number of blocks used in checking non-linear transformations. Default value is floor(sqrt(T)).

Details

The program follows the following steps:

- Check for seasonality: fitting a multiplicative ARIMA(p,0,0)(1,0,0)_s model to a scalar time series and testing if the estimated seasonal AR coefficient is significant.
- Check for non-linear transformation: the series is divided into a given number of consecutive blocks and in each of them the Mean Absolute Deviation (MAD) and the median is computed. A regression of the log of the MAD with respect to the log of the median is run and the slope defines the non-linear transformation.
- Select orders: maximum order of (p, d, q).

Value

A list containing:

- data - The time series. If any non-linear transformation is taken, "data" is the transformed series.
- order - Regular ARIMA order.
- sorder - Seasonal ARIMA order.
- period - Seasonal period.
- include.mean - Switch concerning the inclusion of mean in the model.

Examples

```
data(TaiwanAirBox032017)
fit <- arimaID(TaiwanAirBox032017[,1])
```

---

arimaSpec *Automatic Modeling of a Scalar Non-Seasonal Time Series*

Description

Select an ARIMA model for a non-seasonal scalar time series. It uses augmented Dickey-Fuller (ADF) test to check for unit roots. The maximum degree of differencing is 2.

Usage

```
arimaSpec(
  zt,
  maxorder = c(5, 1, 4),
  criterion = "bic",
  output = FALSE,
  method = "CSS-ML",
  pv = 0.01
)
```

Arguments

zt	T by 1 vector of an observed scalar time series without missing values.
maxorder	Maximum order of (p, d, q) where p is the AR order, d the degree of differencing, and q the MA order. Default value is (5,1,4).
criterion	Information criterion used for model selection. Either AIC or BIC. Default is "bic".
output	If TRUE it returns the differencing order, the selected order and the minimum value of the criterion. Default is TRUE.
method	Estimation method. See the arima command in R. Possible values are "CSS-ML", "ML", and "CSS". Default is "CSS-ML".
pv	P-value for unit-root test. Default is 0.01.

Details

Find the AR order by checking a pure AR model for the differenced series. The maximum AR order tried is min(default AR order and the order of pure AR model). Check the MA order by checking pure MA model using rank-based Ljung-Box statistics. The maximum MA order tried is the min(default MA order and the order of pure MA model). Finally, sequentially decreasing the AR order and increasing the MA order to obtain best models using the specified criterion function.

Value

A list containing:

- order - Regular ARIMA order.
- crit - Minimum criterion.
- include.mean - Switch about including mean in the model.

Examples

```
data(TaiwanAirBox032017)
fit <- arimaSpec(as.matrix(TaiwanAirBox032017[,1]))
```

---

chksea *Check the Seasonality of Each Component of a Multiple Time Series*

Description

Check the seasonality of each component of a multiple time series.

Usage

```
chksea(x, period = c(12), p = 0, alpha = 0.05, output = TRUE)
```

Arguments

x	T by k data matrix: T data points in rows with each row being data at a given time point, and k time series in columns.
period	seasonal period. Default value is 12.
p	Regular AR order. Default value is max(floor(log(T)),1).
alpha	Type-I error for the t-ratio of seasonal coefficients. Default value is 0.05.
output	If TRUE it returns if the series has seasonality. Default is TRUE.

Details

Check the seasonality fitting a seasonal AR(1) model and a regular AR(p) model to a scalar time series and testing if the estimated seasonal AR coefficient is significant.

Value

A list containing:

- Seasonal - TRUE or FALSE.
- period - Seasonal period.

Examples

```
data(TaiwanAirBox032017)
output <- chksea(TaiwanAirBox032017[,1])
```

---

chktrans *Check for Possible Non-linear Transformations of a Multiple Time Series*

Description

Check for possible non-linear transformations of a multiple time series, series by series.

Usage

```
chktrans(x, block = 0, output = FALSE, period = 1, pv = 0.05)
```

Arguments

x	T by k data matrix: T data points in rows with each row being data at a given time point, and k time series in columns.
block	Number of blocks used in the linear regression. Default value is floor(sqrt(T)).
output	If TRUE it returns the estimates, the code: log, sqrt and No-trans and the numbers of non-linear transformations. Default is TRUE.
period	Seasonal period.
pv	P-value = pv/log(1 + k) is used to check the significance of the coefficients. Default value is 0.05.

Details

Each series is divided into a given number of consecutive blocks and in each of them the mean absolute deviation (MAD) and the median are computed. A regression of the log of the MAD with respect to the log of the median is run and the slope defines the non-linear transformation.

Value

A list containing:

- lnTran - Column locations of series that require log-transformation.
- sqrtTran - Column locations of series that require square-root transformation.
- noTran - Column locations of series that require no transformation.
- tran - A vector indicating checking results, where 0 means no transformation, 1 means log-transformation, 2 means square-root transformation.
- tranX Transformed series. This is only provided if the number of series requiring transformation is sufficiently large, i.e. greater than $2kpv$.
- Summary Number of time series that require log-transformation, square-root transformation and no transformation.

Examples

```
data(TaiwanAirBox032017)
output <- chktrans(TaiwanAirBox032017[,1])
```

---

clothing *Cloth sales in China*

Description

Daily sales, in natural logarithms, of a clothing brand in 25 provinces in China from January 1, 2008, to December 9, 2012. The number of observations is 1812.

Usage

```
data(clothing)
```

Format

An object of class `"data.frame"`.

References

Chang, J., Guo, B., and Yao, Q. (2018). Principal component analysis for second-order stationary vector time series. *The Annals of Statistics*, 46(5), 2094-2124.

ClusKur — *Cluster Identification Procedure using Projections on Directions of Extreme Kurtosis Coefficient*

Description

Identification of groups using projections of a vector of features of each time series in directions of extreme kurtosis coefficient.

Usage

```
ClusKur(x)
```

Arguments

x	p by k data matrix: p features or variables for each time series and k time series in columns.

Value

A list containing:

- lbl - Cluster labels (possible outliers get negative labels).
- ncl - Number of clusters.

Examples

```
data(Stockindexes99world)
S <- Stockindexes99world[,-1]
v1 <- apply(S,2, mean)
v2 <- apply(S,2, sd)
M <- rbind(v1,v2)
out <- ClusKur(M)
```

CPIEurope200015 — *Price Indexes EUUS*

Description

Monthly consumer price indexes of European countries and the United States of America for the period January 2000 to October 2015. There are 33 indexes and the names of the countries are the columns names. The number of observations is 190.

Usage

```
data("CPIEurope200015")
```

Format

An object of class "data.frame".

Source

https://ec.europa.eu/eurostat

---

dfmpc *Dynamic Factor Model by Principal Components*

Description

The function estimates the Dynamic Factor Model by Principal Components and by the estimator of Lam et al. (2011).

Usage

```
dfmpc(x, stand = 0, mth = 4, r, lagk = 0)
```

Arguments

x	T by k data matrix: T data points in rows with each row being data at a given time point, and k time series in columns.
stand	Data standardization. The default is stand = 0 and x is not transformed, if stand = 1 each column of x has zero mean an if stand=2 also unit variance.
mth	Method to estimate the number of factors and the common component (factors and loadings): • mth = 0 - the number of factors must be given by the user and the model is estimated by Principal Components. • mth = 1 - the number of factors must be given by the user and the model is estimated using Lam et al. (2011) methodology. • mth = 2 - the number of factors is estimated using Bai and Ng (2002) ICP1 criterion and the model is estimated by Principal Components. • mth = 3 - the number of factors is estimated using Bai and Ng (2002) ICP1 criterion and the model is estimated using Lam et al. (2011) methodology. • mth = 4 - the number of factors is estimated by applying once the Lam and Yao (2012) criterion and the model is estimated using Lam et al. (2011) methodology (default method). • mth = 5 - the number of factors is estimated using Ahn and Horenstein (2013) test and the model is estimated by Principal Components. • mth = 6 - the number of factors is estimated using Caro and Peña (2020) test and the model is estimated using Lam et al. (2011) methodology with the combined correlation matrix.
r	Number of factors, default value is estimated by Lam and Yao (2012) criterion.
lagk	Maximum number of lags considered in the combined matrix. The default is lagk = 3.

Value

A list with the following items:

- r - Estimated number of common factors, if mth=0, r is given by the user.
- F - Estimated common factor matrix (T x r).
- L - Estimated loading matrix (k x r).
- E - Estimated noise matrix (T x k).
- VarF - Proportion of variability explained by the factor and the accumulated sum.
- MarmaF - Matrix giving the number of AR, MA, seasonal AR and seasonal MA coefficients for the Factors, plus the seasonal period and the number of non-seasonal and seasonal differences.
- MarmaE - Matrix giving the number of AR, MA, seasonal AR and seasonal MA coefficients for the noises, plus the seasonal period and the number of non-seasonal and seasonal differences.

References

Ahn, S. C. and Horenstein, A. R. (2013). Eigenvalue ratio test for the number of factors. *Econometrica*, 81(3):1203–1227.

Bai, J. and Ng, S. (2002). Determining the number of factors in approximate factor models. *Econometrica*, 70(1):191–221.

Caro, A. and Peña, D. (2020). A test for the number of factors in dynamic factor models. UC3M Working papers. Statistics and Econometrics.

Lam, C. and Yao, Q. (2012). Factor modeling for high-dimensional time series: inference for the number of factors. *The Annals of Statistics*, 40(2):694–726.

Lam, C., Yao, Q., and Bathia, N. (2011). Estimation of latent factors for high-dimensional time series. *Biometrika*, 98(4):901–918.

Examples

```
data(TaiwanAirBox032017)
dfm1 <- dfmpc(as.matrix(TaiwanAirBox032017[1:100,1:30]), mth=4)
```

DLdata *Create an input data matrix for a Deep learning program that uses time series data.*

Description

R command to setup the training and forecasting data for deep learning.

Usage

```
DLdata(x, forerate = 0.2, locY = 1, lag = 1)
```

Arguments

x	T by k data matrix: T data points in rows and k time series in columns.
forerate	Fraction of sample size to form the forecasting (or testing) sample.
locY	Locator for the dependent variable.
lag	Number of lags to be used to form predictors.

Value

A list containing:

- Xtrain - Standardized predictors matrix.
- Ytrain - Dependent variable in training sample.
- Xtest - Predictor in testing sample, standardized according to X_train.
- Ytest - Dependent variable in the testing sample.
- nfore - Number of forecasts.

Examples

```
x <- matrix(rnorm(7000), nrow=700, ncol=100)
m1 <- DLdata(x, forerate=c(200/nrow(x)), lag=6, locY=6)
```

---

draw.coef *Random Draw of Coefficients for AR Models and MA Models*

Description

Random draw of polynomial coefficients for stationary AR models or invertible MA models. The resulting polynomial has solutions outside the unit circle.

Usage

```
draw.coef(deg, delta = 0.02)
```

Arguments

deg	Degree of the polynomial. Maximum degree is 5.
delta	The minimum distance of a polynomial root from the boundary 1 or -1. The default is 0.02.

Value

$c = (c1, c2, ...)$ denotes the coefficients of $1 - c1 * x - c2 * x^2 - ....$

Examples

```
draw.coef(2)
```

edqplot *Plot the Observed Time Series and Selected EDQs (Empirical Dynamic Quantiles)*

Description

Plot the observed time series and selected empirical dynamic quantiles (EDQs) computed as in Peña, Tsay and Zamar (2019).

Usage

```
edqplot(
  x,
  prob = c(0.05, 0.5, 0.95),
  h = 30,
  loc = NULL,
  color = c("yellow", "red", "green")
)
```

Arguments

x	T by k data matrix: T data points in rows with each row being data at a given time point, and k time series in columns.
prob	Probability, the quantile series of which is to be computed. Default values are 0.05, 0.5, 0.95.
h	Number of time series used in the algorithm. Default value is 30.
loc	Locations of the EDQ. If loc is not null, then prob is not used.
color	Colors for plotting the EDQ. Default is "yellow", "red", and "green".

Value

The observed time series plot with the selected EDQs.

References

Peña, D. Tsay, R. and Zamar, R. (2019). Empirical Dynamic Quantiles for Visualization of High-Dimensional Time Series, *Technometrics*, 61:4, 429-444.

Examples

```
data(TaiwanAirBox032017)
edqplot(TaiwanAirBox032017[1:100,1:25])
```

edqts — *Empirical Dynamic Quantile for Visualization of High-Dimensional Time Series*

Description

Compute empirical dynamic quantile (EDQ) for a given probability "p" based on the weighted algorithm proposed in the article by Peña, Tsay and Zamar (2019).

Usage

```
edqts(x, p = 0.5, h = 30)
```

Arguments

x	T by k data matrix: T data points in rows with each row being data at a given time point, and k time series in columns.
p	Probability, the quantile series of which is to be computed. Default value is 0.5.
h	Number of time series observations used in the algorithm. The larger h is the longer to compute. Default value is 30.

Value

The column of the matrix x which stores the "p" EDQ of interest.

References

Peña, D. Tsay, R. and Zamar, R. (2019). Empirical Dynamic Quantiles for Visualization of High-Dimensional Time Series, *Technometrics*, 61:4, 429-444.

Examples

```
data(TaiwanAirBox032017)
edqts(TaiwanAirBox032017[,1:25])
```

FREDMDApril19 — *Federal Reserve Bank at St Louis.*

Description

Data obtained from the Federal Research Bank after process to remove missing values.

Usage

```
data("FREDMDApril19")
```

Format

An object of class "data.frame".

Source

https://www.stlouisfed.org/

---

gap.clus *Gap statistics*

Description

This function computes the gap and the number of groups using the gap statistics.

Usage

```
gap.clus(DistanceMatrix, Clusters, B)
```

Arguments

DistanceMatrix	Square matrix of GCC distances.
Clusters	Matrix of member labels.
B	Number of iterations for the bootstrap.

Value

A list containing:

- - optim.k: number of groups.
- - gap.values: gap values.

References

Alonso, A. M. and Peña, D. (2019). Clustering time series by linear dependency. *Statistics and Computing*, 29(4):655–676.

Examples

```
data(TaiwanAirBox032017)
library(TSclust)
z <- diff(as.matrix(TaiwanAirBox032017[1:50,1:8]))
Macf <- as.matrix(diss(t(z), METHOD = "ACF", lag.max = 5))
sc1 <- hclust(as.dist(Macf), method = "complete")
memb <- cutree(sc1, 1:8)

ngroups <- gap.clus(Macf, memb, 100)
```

GCCclus *Clustering of Time Series Using the Generalized Cross Correlation Measure of Linear dependency*

Description

Clustering of time series using the Generalized Cross Correlation (GCC) measure of linear dependency proposed in Alonso and Peña (2019).

Usage

```
GCCclus(x, lag, rs, thres, plot, printSummary = TRUE, lag.set, silh = 1)
```

Arguments

x	T by k data matrix: T data points in rows with each row being data at a given time point, and k time series in columns.
lag	Selected lag for computing the GCC between the pairs of series. Default value is computed inside the program.
rs	Relative size of the minimum group considered. Default value is 0.05.
thres	Percentile in the distribution of distances that define observations that are not considered outliers. Default value is 0.9.
plot	If the value is TRUE, a clustermatrix plot of distances and a dendogram are presented. Default is FALSE.
printSummary	If the value is TRUE, the function prints a summary table of the clustering. Default is TRUE.
lag.set	If lag is not specified and the user wants to use instead of lags from 1 to 'lag' a non consecutive set of lags they can be defined as lag.set = c(1, 4, 7).
silh	If silh = 1 standard silhoutte statistics and if silh = 2 modified procedure. Default value is 1.

Details

First, the matrix of Generalized Cross correlation (GCC) is built by using the subrutine GCCmatrix, then a hierarchical grouping is constructed and the number of clusters is selected by either the silhouette statistics or a modified silhouette statistics The modified silhouette statistics is as follows:

- (1) Series that join the groups at a distance larger than a given threshold of the distribution of the distances are disregarded.
- (2) A minimum size for the groups is defined by rs, relative size, groups smaller than rs are disregarded.
- (3) The final groups are obtained in two steps:
  - First the silhouette statistics is applied to the set of time series that verify conditions (1) and (2).

  - Second, the series disregarded in steps (1) and (2) are candidates to be assigned to its closest group. It is checked using the median and the MAD of the group if the point is or it is not an outlier with respect to the group. If it is an outlier it is included in a group 0 of outlier series. The distance between a series and a group is usually to the closest in the group (simple linkage) but could be to the mean of the group.

Value

A list containing:

- - Table of number of clusters found and number of observations in each cluster. Group 0 indicates the outlier group in the case it exists.
- - sal: A list with four objects
  - labels: assignments of time series to each of the groups.
  - groups: is a list of matrices. Each matrix corresponds to the set of time series that make up each group. For example, $groups[[i]] contains the set of time series that belong to the ith group.
  - matrix: GCC distance matrix.
  - gmatrix: GCC distance matrices in each group.
- Two plots are included (1) A clustermatrix plot with the distances inside each group in the diagonal boxes and the distances between series in two groups in off-diagonal boxes (2) the dendogram.

References

Alonso, A. M. and Peña, D. (2019). Clustering time series by linear dependency. *Statistics and Computing*, 29(4):655–676.

Examples

```
data(TaiwanAirBox032017)
output <- GCCclus(TaiwanAirBox032017[1:50,1:8])
```

`GCCmatrix` *Generalized Cross-Correlation Matrix*

Description

Built the GCC similarity matrix between time series proposed in Alonso and Peña (2019).

Usage

```
GCCmatrix(x, lag, model, lag.set)
```

Arguments

x	T by k data matrix: T data points in rows with each row being data at a given time point, and k time series in columns.
lag	Selected lag for computing the GCC between the pairs of series. Default value is computed inside the program.
model	Model specification. When the value lag is unknown for the user, the model specification is chosen between GARCH model or AR model. Default is ARMA model.
lag.set	If lag is not specified and the user wants to use instead of lags from 1 to 'lag' a non consecutive set of lags they can be defined as lag.set = c(1, 4, 7).

Value

A list containing:

- DM - A matrix object with the distance matrix.
- k_GCC - The lag used to calculate GCC measure.

References

Alonso, A. M. and Peña, D. (2019). Clustering time series by linear dependency. *Statistics and Computing*, 29(4):655–676.

Examples

```
data(TaiwanAirBox032017)
output <- GCCmatrix(TaiwanAirBox032017[,1:3])
```

---

gdpsimple6c8018 *Growth of GDP in 6 Countries*

Description

$([x(t) - x(t-1)]/x(t-1))$ of United States, United Kingdom, France, Australia, Germany, and Canada. Quarterly data from 1980 to 2018. The original data are downloaded from FRED (Federal Reserve Bank of St Louis). The GDP is based on expenditures.

Usage

```
data(gdpsimple6c8018)
```

Format

An object of class `"data.frame"`.

Source

https://fred.stlouisfed.org/

i.plot *Plot a Selected Time Series Using Quantile as the Background*

Description

Plot a selected time series using quantile as the background.

Usage

```
i.plot(x, idx = 1, prob = c(0.25, 0.5, 0.75), xtime = NULL)
```

Arguments

x	T by k data matrix: T data points in rows with each row being data at a given time point, and k time series in columns.
idx	Selected time series.
prob	Probability, the quantile series of which is to be computed. Default values are 0.25, 0.5, 0.75.
xtime	A vector with the values for the x labels. Default values are 1, 2, 3, ...

Value

standardized - Matrix containing the standardized time series.

Examples

```
data(TaiwanAirBox032017)
output <- i.plot(TaiwanAirBox032017[,1:3])
```

i.qplot *Plot the Closest Series to a Given Timewise Quantile Series*

Description

Use sum of absolute deviations to select the individual time series that is closest to a given timewise quantile series.

Usage

```
i.qplot(x, prob = 0.5, box = 3, xtime = NULL)
```

Arguments

x	T by k data matrix: T data points in rows with each row being data at a given time point, and k time series in columns.
prob	Probability, the quantile series of which is to be computed. Default value is 0.5.
box	Number of boxplots for the difference series between the selected series and the timewise quantile with the given probability. The differences are divided into blocks. Default value is 3.
xtime	A vector with the values for the x labels. Default values are 1, 2, 3, ...

Value

A list containing:

- standardized - A matrix containing standardized time series.
- qts - The timewise quantile of order prob.
- selected - The closest time series to the given timewise quantile series.

Examples

```
data(TaiwanAirBox032017)
output <-  i.qplot(TaiwanAirBox032017[,1:3])
```

---

i.qrank *Rank Individual Time Series According to a Given Timewise Quantile Series*

Description

Use sum of absolute deviations to select the individual time series that is closest to a given timewise quantile series.

Usage

```
i.qrank(x, prob = 0.5)
```

Arguments

x	T by k data matrix: T data points in rows with each row being data at a given time point, and k time series in columns.
prob	Probability, the quantile series of which is to be computed. Default value is 0.5.

Value

A list containing:

- standardized - A matrix containing standardized time series.
- qts - The timewise quantile of order prob.
- ranks - Rank of the individual time series according to a the given timewise quantile series.
- crit - Sum of absolute deviations of each individual series. Distance of each series to the quantile.

Examples

```
data(TaiwanAirBox032017)
output <-  i.qrank(TaiwanAirBox032017[,1:3])
```

Lambda.sel *Select the Penalty Parameter of LASSO-type Linear Regression*

Description

Use out-of-sample Root Mean Square Error to select the penalty parameter of LASSO-type linear regression.

Usage

```
Lambda.sel(X, y, newX, newY, family = "gaussian", alpha = 1)
```

Arguments

X	Matrix of predictors of the estimation sample.
y	Dependent variables of the estimation sample.
newX	Design matrix in the forecasting subsample.
newY	Dependent variable in the forecasting subsample.
family	Response type. See the glmnet command in R. Possible types are "gaussian", "binomial", "poisson", "multinomial", "cox", "mgaussian". Default is "gaussian".
alpha	The elasticnet mixing parameter, with $0 \leq \alpha \leq 1$. See the glmnet command in R. Default value is 1.

Value

A list containing:

- lambda.min - lambda that achieves the minimum mean square error.
- beta - estimated coefficients for lambda.min.
- mse - mean squared error.
- lambda - the actual sequence of lambda values used.

Examples

```
X <- cbind(rnorm(200),rnorm(200,2,1),rnorm(200,4,1))
y <- rnorm(200)
newX <- cbind(rnorm(200),rnorm(200,2,1),rnorm(200,4,1))
newy <- rnorm(200)
output <- Lambda.sel(X, y, newX, newy)
```

---

locations032017 *Locations of Air-Box Devices in Taiwan*

Description

Latitude and longitude of the Air Boxes used in TaiwanAirBox032017.

Usage

```
data(locations032017)
```

Format

An object of class `"data.frame"`.

References

Chen, L.J. et al. (2017). Open framework for participatory PM2.5 monitoring in smart cities. *IEEE Access Journal*, Vol. 5, pp. 14441-14454.

---

mdec1to5 *Monthly Simple Returns of United States Market Portfolios.*

Description

Based on Market Cap. The first 5 (smallest 10 percentage, next 10 percentage, etc.). The time span is from January 1962 to December 2018. The data is from CRSP (Center of Research for the Security Prices). The first column is calendar time.

Usage

```
data("mdec1to5")
```

Format

An object of class `"data.frame"`.

Source

https://www.crsp.org/

mexpimpcnus *Monthly Exports and Imports of China and United States.*

Description

The original data are from FRED (Federal Reserve Bank of St Louis) and the unit is US Dollars. The time span is from January 1992 to December 2018.

Usage

```
data("mexpimpcnus")
```

Format

An object of class "data.frame".

Source

https://fred.stlouisfed.org/

mts.plot *Plot Multiple Time Series in One Frame*

Description

Plot multiple time series in one frame and return standardized time series.

Usage

```
mts.plot(x, title = "mts plot", scaling = TRUE, xtime = NULL)
```

Arguments

x	T by k data matrix: T data points in rows with each row being data at a given time point, and k time series in columns.
title	Character with the title of the plot. Default title is "mts plot".
scaling	If scaling = TRUE (default), then each series is standardized based on its own range. If scaling = FALSE, then the original series is used.
xtime	A vector with the values for the x labels. Default values are 1, 2, 3, ...

Value

standardized - Matrix containing the standardized time series.

Examples

```
data(TaiwanAirBox032017)
output <- mts.plot(TaiwanAirBox032017[,1:5])
```

`mts.qplot` *Plot Timewise Quantiles in One Frame*

Description

Plot timewise quantiles in one frame.

Usage

```
mts.qplot(
  x,
  title = "mts quantile plot",
  prob = c(0.25, 0.5, 0.75),
  scaling = TRUE,
  xtime = NULL,
  plot = TRUE
)
```

Arguments

`x`	T by k data matrix: T data points in rows with each row being data at a given time point, and k time series in columns.
`title`	Character with the title of the plot. Default title is "mts quantile plot".
`prob`	Probability, the quantile series of which is to be computed. Default values are 0.25, 0.5, 0.75.
`scaling`	If scaling = TRUE (default), then each series is standardized based on its own range. If scaling = FALSE, then the original series is used.
`xtime`	A vector with the values for the x labels. Default values are 1, 2, 3, ...
`plot`	Receives TRUE or FALSE values. If the value is TRUE, a quantile plot is presented. Defaults is TRUE.

Value

A list containing:

- standardized - A matrix containing standardized time series.
- qseries - Matrix of timewise quantiles series of order prob.

Examples

```
data(TaiwanAirBox032017)
output <-  mts.qplot(TaiwanAirBox032017[,1:5])
```

`outlier.plot` *Find Outliers Using an Upper and a Lower Timewise Quantile Series*

Description

Use an upper and a lower timewise quantile series to highlight the possible outliers in a collection of time series.

Usage

```
outlier.plot(x, prob = 0.05, percent = 0.05, xtime = NULL)
```

Arguments

`x`	T by k data matrix: T data points in rows with each row being data at a given time point, and k time series in columns.
`prob`	Tail probability. That is, the two quantile series is (prob, 1-prob). prob is restricted to be in (0,0.15). Default value is 0.05.
`percent`	The number of possible outliers in each side is T*k*prob*percent.
`xtime`	A vector with the values for the x labels. Default values are 1, 2, 3, ...

Value

A list containing:

- standardized - A matrix containing standardized time series.
- qts - The timewise quantile of order prob.
- minseries - The timewise minimum of the standardized time series.
- maxseries - The timewise maximum of the standardized time series.

Examples

```
data(TaiwanAirBox032017)
output <-  outlier.plot(TaiwanAirBox032017[,1:3])
```

`outlierLasso` *Outliers LASSO*

Description

Use LASSO estimation to identify outliers in a set of time series by creating dummy variables for every time point.

Usage

```
outlierLasso(
  zt,
  p = 12,
  crit = 3.5,
  family = "gaussian",
  standardize = TRUE,
  alpha = 1,
  jend = 3
)
```

Arguments

`zt`	T by 1 vector of an observed scalar time series without missing values.
`p`	Seasonal period. Default value is 12.
`crit`	Criterion. Default is 3.5.
`family`	Response type. See the glmnet command in R. Possible types are "gaussian", "binomial", "poisson", "multinomial", "cox", "mgaussian". Default is "gaussian".
`standardize`	Logical flag for zt variable standardization. See the glmnet command in R. Default is TRUE.
`alpha`	Elasticnet mixing parameter, with $0 \leq \alpha \leq 1$. See the glmnet command in R. Default value is 1.
`jend`	Number of first and last observations assumed to not be level shift outliers. Default value is 3.

Value

A list containing:

- nAO - Number of additive outliers.
- nLS - Number of level shifts.

Examples

```
data(TaiwanAirBox032017)
output <- outlierLasso(TaiwanAirBox032017[1:100,1])
```

`outliers.hdts` *Multivariate Outlier Detection*

Description

Outlier detection in high dimensional time series by using projections as in Galeano, Peña and Tsay (2006).

Usage

```
outliers.hdts(x, r.max, type)
```

Arguments

`x`	T by k data matrix: T data points in rows with each row being data at a given time point, and k time series in columns.
`r.max`	The maximum number of factors including stationary and non-stationary.
`type`	The type of series, i.e., 1 if stationary or 2 if nonstationary.

Value

A list containing:

- x.clean - The time series cleaned at the end of the procedure (n x m).
- P.clean - The estimate of the loading matrix if the number of factors is positive.
- Ft.clean - The estimated dynamic factors if the number of factors is positive.
- Nt.clean - The idiosyncratic residuals if the number of factors is positive.
- times.idi.out - The times of the idiosyncratic outliers.
- comps.idi.out - The components of the noise affected by the idiosyncratic outliers.
- sizes.idi.out - The sizes of the idiosyncratic outliers.
- stats.idi.out - The statistics of the idiosyncratic outliers.
- times.fac.out - The times of the factor outliers.
- comps.fac.out - The dynamic factors affected by the factor outliers.
- sizes.fac.out - The sizes of the factor outliers.
- stats.fac.out - The statistics of the factor outliers.
- x.kurt - The time series cleaned in the kurtosis sub-step (n x m).
- times.kurt - The outliers detected in the kurtosis sub-step.
- pro.kurt - The projection number of the detected outliers in the kurtosis sub-step.
- n.pro.kurt - The number of projections leading to outliers in the kurtosis sub-step.
- x.rand - The time series cleaned in the random projections sub-step (n x m).
- times.rand - The outliers detected in the random projections sub-step.

- x.uni - The time series cleaned after the univariate substep (n x m).
- times.uni - The vector of outliers detected with the univariate substep.
- comps.uni - The components affected by the outliers detected with the univariate substep.
- r.rob - The number of factors estimated (1 x 1).
- P.rob - The estimate of the loading matrix (m x r.rob).
- V.rob - The estimate of the orthonormal complement to P (m x (m - r.rob)).
- I.cov.rob - The matrix (V'GnV)^-1 used to compute the statistics to detect the idiosyncratic outliers.
- IC.1 - The values of the information criterion of Bai and Ng.

References

Galeano, P., Peña, D., and Tsay, R. S. (2006). Outlier detection in multivariate time series by projection pursuit. *Journal of the American Statistical Association*, 101(474), 654-669.

Examples

```
data(TaiwanAirBox032017)
output <- outliers.hdts(as.matrix(TaiwanAirBox032017[1:100,1:3]), r.max = 1, type =2)
```

`PElectricity1344` *Electricity Prices in New England and USA*

Description

Weakly series of electricity price each hour of each day during 678 weeks in the 8 regions in New England. We have 1344 series corresponding to each of the seven days, one of the 24 hours and for one of the regions ($7x24x8 = 1344$). The first series correspond to the price in the first region at 1 am CT of Thursday 01/01/2004, the second to 2 am, same day and so on. Thus the first 192 series (24 hours x 8 regions) are the price of all the hours of Thursday in the eight regions, the next 192 are for Friday and so on. These series were used in the articles Alonso and Peña (2019), and Peña, Tsay and Zamar (2019). The series have been corrected of missing values at days of changing time in saving energy days.

Usage

```
data(PElectricity1344)
```

Format

An object of class `"data.frame"`.

References

Alonso, A. M. and Peña, D. (2019). Clustering time series by linear dependency. *Statistics and Computing*, 29(4):655–676.

Peña, D. Tsay, R. and Zamar, R. (2019). Empirical Dynamic Quantiles for Visualization of High-Dimensional Time Series, *Technometrics*, 61:4, 429-444.

quantileBox *Quantile Boxplot*

Description

Boxplots of selected quantiles of each time series (Column-wise operations).

Usage

```
quantileBox(x, prob = c(0.25, 0.5, 0.75))
```

Arguments

x	T by k data matrix: T data points in rows with each row being data at a given time point, and k time series in columns.
prob	Probability, the quantile series of which is to be computed. Default values are 0.25, 0.5, 0.75.

Value

Boxplot.

Examples

```
data(TaiwanAirBox032017)
quantileBox(TaiwanAirBox032017[,1:3])
```

rnnStream *Setup the Input and Output for a Recurrent Neural Network*

Description

R command to setup the input and output for a Recurrent Neural Network. It is used in the Wiley book *Statistical Learning with Big Dependent Data* by Daniel Peña and Ruey S. Tsay (2021).

Usage

```
rnnStream(z, h = 25, nfore = 200)
```

Arguments

z	Input in integer values.
h	Number of lags used as input.
nfore	Data points in the testing subsample.

Value

A list containing:

- Xfit - Predictor in training sample (binary).
- Yfit - Dependent variable in the training sample (binary).
- yp - Dependent variable in testing sample.
- Xp - Predictor in the testing sample (binary).
- X - Predictor in the training sample.
- yfit - Dependent variable in the training sample.
- newX - Predictor in the testing sample.

Examples

```
output <- rnnStream(rnorm(100), h=5, nfore=20)
```

---

sarimaSpec *Automatic Modeling of a Scalar Seasonal Time Series*

Description

Auto-model specification of a scalar seasonal time series. The period should be given.

Usage

```
sarimaSpec(
  zt,
  maxorder = c(2, 1, 3),
  maxsea = c(1, 1, 1),
  criterion = "bic",
  period = 12,
  output = FALSE,
  method = "CSS-ML",
  include.mean = TRUE
)
```

Arguments

zt	T by 1 vector of an observed scalar time series without missing values.
maxorder	Maximum order of (p, d, q). p is the AR order, d the degree of differencing, and q The MA order. Default value is (2,1,3).
maxsea	Maximum order of (P, D, Q). P is the seasonal AR order, D the degree of seasonal differencing, and Q the seasonal MA order. Default value is (1,1,1).
criterion	Information criterion used for model selection. Either AIC or BIC. Default is "bic".

period	Seasonal period. The default is 12.
output	If TRUE it returns the differencing order, the selected order and the minimum value of the criterion. Default is TRUE.
method	Estimation method. See the arima command in R. Possible values are "CSS-ML", "ML", and "CSS". Default is "CSS-ML".
include.mean	Should the model include a mean/intercept term? Default is TRUE.

Details

ADF unit-root test is used to assess seasonal and regular differencing. For seasonal unit-root test, critical value associated with pv = 0.01 is used.

Value

A list containing:

- data - The time series. If any transformation is taken, "data" is the transformed series.
- order - Regular ARIMA order.
- sorder - Seasonal ARIMA order.
- period - Seasonal period.
- include.mean - Switch about including mean in the model.

Examples

```
data(TaiwanAirBox032017)
output <- sarimaSpec(TaiwanAirBox032017[1:100,1])
```

---

scatterACF *Scatterplot of Two Selected-lag Autocorrelation Functions*

Description

Scatterplot of two selected-lag ACFs.

Usage

```
scatterACF(x, lags = c(1, 2))
```

Arguments

x	T by k data matrix: T data points in rows with each row being data at a given time point, and k time series in columns.
lags	Set of lags. Default values are 1, 2.

Value

A list containing:

- acf1 - Autocorrelation function of order lags[1].
- acf2 - Autocorrelation function of order lags[2].

Examples

```
data(TaiwanAirBox032017)
output <- scatterACF(TaiwanAirBox032017[,1:100])
```

`SelectedSeries` *Identified the Series with the Given Order*

Description

To be used after the command "SummaryModel". The input "M" must be an output from "SummaryModel". Selected time series of a given order (p, d, q).

Usage

```
SelectedSeries(M, order = c(1, 0, 1))
```

Arguments

`M`	Matrix that is an output from "SummaryModel" command, that is, M1, M2 or M3.
`order`	Specification of the non-seasonal part of the ARIMA model: the three integer components (p, d, q) are p the AR order, d the degree of differencing, and q the MA order. Default value is c(1, 0, 1).

Value

The number of series with the given order and the names of the resulting series.

Examples

```
data(TaiwanAirBox032017)
outputSummaryModel <- SummaryModel(TaiwanAirBox032017[,1:3])
SelectedSeries(outputSummaryModel$M1, order = c(2,0,0))
```

silh.clus *Find the Number of Clusters by the Standard Silhouette Statistics*

Description

Find the number of clusters by the standard Silhouette statistics. The cluster is hierarchical.

Usage

```
silh.clus(nClus, distanceMatrix, method)
```

Arguments

nClus	Maximum number of groups.
distanceMatrix	Matrix of distances.
method	Hierarchical method "single", "average","complete".

Value

A list containing:

- nClus - Number of groups
- list - Silhouette statistics for each value of nclus.

Examples

```
data(TaiwanAirBox032017)
output_gcc <- GCCmatrix(TaiwanAirBox032017[1:100,1:10])
output <- silh.clus(nClus=3,distanceMatrix=output_gcc$DM ,method="complete")
```

sim.urarima *Generate Unit-root ARIMA Possibly Seasonal Time Series*

Description

Generate Unit-root ARIMA, possibly, seasonal time series.

Usage

```
sim.urarima(
  T = 300,
  ar = c(0.5),
  ma = c(-0.5),
  d = 1,
  sar = NULL,
  sma = NULL,
  D = 0,
  period = 12,
  ini = 200,
  df = 50
)
```

Arguments

T	Number of observations.
ar	Vector with the autoregressive coefficients. Default value is 0.5.
ma	Vector with the moving average coefficients. Default value is -0.5.
d	Order of first-differencing. Default value is 1.
sar	Seasonal autoregressive coefficients. Default is NULL.
sma	Seasonal moving average coefficients. Default is NULL.
D	Order of seasonal differencing. Default value is 0.
period	Seasonal period. Default value is 12.
ini	Length of 'burn-in' period. Default value is 200.
df	If df ≥ 50 random generation for the Normal distribution, if df < 50 random generation for the t distribution with df degrees of freedom. Default value is 50.

Value

A time series vector.

Examples

```
x <- sim.urarima()
```

---

sSelectedSeries *Selected Seasonal Time Series*

Description

To be used after the command "sSummaryModel". The input "M" must be output from "sSummaryModel". Selected seasonal time series of a given order $(p, d, q) * (P, D, Q)$.

Usage

```
sSelectedSeries(M, order = c(0, 1, 1, 0, 1, 1))
```

Arguments

M	Matrix that is an output from "sSummaryModel" command, that is, M1, M2, M3, M4, M5, or M6.
order	order of ARIMA model $(p, d, q) * (P, D, Q)$. Default values is (0, 1, 1, 0, 1, 1).

Value

A list with the series names and count.

Examples

```
data(TaiwanAirBox032017)
outputSummaryModel <- sSummaryModel(TaiwanAirBox032017[,1:3])
sSelectedSeries(outputSummaryModel$M1)
```

---

sSummaryModel *Collects All Models Specified by "sarimaSpec"*

Description

Models specified by "sarimaSpec".

Usage

```
sSummaryModel(
  x,
  maxorder = c(3, 1, 2),
  period = 12,
  criterion = "bic",
  method = "CSS"
)
```

Arguments

x	T by k data matrix: T data points in rows with each row being data at a given time point, and k time series in columns.
maxorder	Maximum order of ARIMA model (p, d, q) where p is the AR order, d the degree of differencing, and q the MA order. Default value is (3,1,2).
period	Seasonal period. The default is 12.
criterion	Information criterion used for model selection. Either AIC or BIC. Default is "bic".
method	Estimation method. See the arima command in R. Possible values are "CSS-ML", "ML", and "CSS". Default is "CSS".

Value

A list containing:

- Order - Order of ARIMA model (p, d, q, P, D, Q) of each series. A matrix of (ncol(x),6). The six columns are "p","d","q", "P", "D", "Q".
- Mean - A logical vector indicating whether each series needs a constant (or mean).
- M1 - Contains orders the stationary series.
- M2 - Contains orders of series with (d=1) and (D=0).
- M3 - Contains orders of series with (d=2) and (D=0).
- M4 - Contains orders of series with (d=0) and (D=1).
- M5 - Contains orders of series with (d=1) and (D=1).
- M6 - Contains orders of series with (d=2) and (D=1).

Examples

```
data(TaiwanAirBox032017)
summary <- sSummaryModel(TaiwanAirBox032017[,1:3])
```

---

stepp *Stepp*

Description

To compute and plot the observed and simulated distances for measuring similarity between time series. The distance can be computed using ACF, PACF, AR-coefficients, or Periodogram.

Usage

```
stepp(
  x,
  M = 100,
  lmax = 5,
  alpha = 0.95,
  dismethod = "ACF",
  clumethod = "complete"
)
```

Arguments

x	T by k data matrix: T data points in rows with each row being data at a given time point, and k time series in columns.
M	Number of simulation realizations. Default value is 100.
lmax	Number of lags used (for ACF, PACF, AR-coefficient). Default value is 5.

alpha	Quantile used in the plotting. Default value is 0.95.
dismethod	Summary statistics of each time series to be used in computing distance. Choices include “ACF”, “PACF”, “AR.PIC” and “PER”. Default is "ACF".
clumethod	Hierarchical clustering method: choices include “single”, “average”, and “complete”. Default is “complete”.

Details

The Empirical Dynamic Quantile of the series is obtained, a set of Txk series is generated and the heights in the dendrogram are obtained. This is repeated M times and the alpha quantile of the results of the M simulations are reported. Both dendrogram's heights and steps (differences) of these heights are compared.

Value

Two plots are given in output:

The first plot shows the “height” of the dendrogram. Solid line is the observed height. The points denote the alpha quantile of heights based on the simulated series.

The second plot shows the “step” of the dendrogam (increments of heights). Solid line is the observed increments and the points are those of selected quantile for the simulated series.

A list containing:

- mh - alpha quantile of heights based on the simulated series.
- mdh - increments of selected quantile for the simulated series.
- hgt - observed height.
- hgtincre - observed increments.
- Mh - the alpha quantile of the results of the M simulations are reported.

Examples

```
data(TaiwanAirBox032017)
output <- stepp(as.matrix(TaiwanAirBox032017[,1:50]), M = 2)
```

Stockindexes99world *World Stock Indexes*

Description

Standardized daily stock indexes of the 99 most important financial markets around the world from January 3, 2000, to December 16, 2015, with a total of 4163 observations. The first column contains the dates and the names of the indexes are the columns names.

Usage

```
data(Stockindexes99world)
```

Format

An object of class "data.frame".

References

Refence or source

Summaryccm *Summary Statistics of Cross-Correlation Matrices*

Description

Compute and plot summary statistics of cross-correlation matrices (CCM) for high-dimensional time series.

Usage

```
Summaryccm(x, max.lag = 12)
```

Arguments

x	T by k data matrix: T data points in rows with each row being data at a given time point, and k time series in columns.
max.lag	The number of lags for CCM.

Value

A list containing:

- pvalue - P-values of Chi-square tests of individual-lag CCM being zero-matrix.
- ndiag - Percentage of significant diagonal elements for each lag.
- noff - Percentage of significant off-diagonal elements for each lag.

Examples

```
data(TaiwanAirBox032017)
output <- Summaryccm(as.matrix(TaiwanAirBox032017[,1:4]))
```

`SummaryModel` *Collects All Models Specified by "arimaSpec"*

Description

Collects all models Specified by "arimaSpec".

Usage

```
SummaryModel(x, maxorder = c(5, 1, 3), criterion = "bic", method = "CSS")
```

Arguments

`x`	T by k data matrix: T data points in rows with each row being data at a given time point, and k time series in columns.
`maxorder`	Maximum order of (p, d, q) where p is the AR order, d the degree of differencing, and q the MA order. Default value is (5,1,3).
`criterion`	Information criterion used for model selection. Either AIC or BIC. Default is "bic".
`method`	Estimation method. See the arima command in R. Possible values are "CSS-ML", "ML", and "CSS". Default is "CSS".

Value

A list containing:

- Order - Orders (p, d, q) of each series. A matrix of (ncol(x),3). The three columns are "p", "d", "q".
- Mean - A logical vector indicating whether each series needs a constant (or mean).
- M1 - A matrix with three columns (p, 0, q). The number of rows is the number of stationary time series. M1 is NULL if there is no stationary series.
- M2 - A matrix with three columns (p, 1, q). The number of rows is the number of first-differenced series. M2 is NULL if there is no first-differenced series.
- M3 - A matrix with three columns (p, 2, q). The number of rows is the number of 2nd-differenced series. M3 is NULL if there is no 2nd-differenced series.
- data - Time series.

Examples

```
x <- matrix(rnorm(300, mean = 10, sd = 4), ncol = 3, nrow = 100)
summary <- SummaryModel(x)
```

SummaryOutliers *Summary Outliers*

Description

Use the command "tso" of the R package "tsoutliers" to identify outliers for each individual time series.

Usage

```
SummaryOutliers(
  x,
  type = c("LS", "AO", "TC"),
  tsmethod = "arima",
  args.tsmethod = list(order = c(5, 0, 0))
)
```

Arguments

x	T by k data matrix: T data points in rows with each row being data at a given time point, and k time series in columns.
type	A character vector indicating the type of outlier to be considered by the detection procedure. See 'types' in tso function.
tsmethod	The framework for time series modeling. Default is "arima". See 'tsmethod' in tso function.
args.tsmethod	An optional list containing arguments to be passed to the function invoking the method selected in tsmethod. See 'args.tsmethod' in tso function. Default value is c(5,0,0).

Value

A list containing:

- Otable - Summary of various types of outliers detected.
- x.cleaned - Outlier-adjusted data.
- xadja - T-dimensional vector containing the number of time series that have outlier at a given time point.

Examples

```
data(TaiwanAirBox032017)
output <- SummaryOutliers(TaiwanAirBox032017[1:50,1:3])
```

TaiwanAirBox032017 *Hourly PM25 Measurements from Air-Box Devices in Taiwan*

Description

Hourly PM25 measurements were constructed from random minute observations collected by Air-Box devices for March 2017. There are 744 observations and 516 series.

Usage

```
data(TaiwanAirBox032017)
```

Format

An object of class `"data.frame"`.

Source

https://sites.google.com/site/cclljj/research/dataset-airbox

References

Chen, L.J. et al. (2017). Open framework for participatory PM2.5 monitoring in smart cities. *IEEE Access Journal*, Vol. 5, pp. 14441-14454.

TaiwanPM25 *Hourly PM25 Measurements in Taiwan*

Description

Hourly measurements of PM_{25} at 15 monitoring stations from southern part of Taiwan from January 1, 2006 to December 31, 2015. The first two columns are the date and the hour. Missing values are filled using fixed window around the missing values. Data of February 29 are removed so that there are 87600 observations in total.

Usage

```
data(TaiwanPM25)
```

Format

An object of class `"data.frame"`.

Source

https://airtw.epa.gov.tw/

temperatures *World Temperatures*

Description

Three series with 106 observations (from year 1910 to 2016) with the deviation with respect to average value of temperatures in November in three regions: Europe, North America and South America. First columns contains the years. Units are degrees Celsius.

Usage

```
data(temperatures)
```

Format

An object of class `"data.frame"`.

Source

https://www.ncdc.noaa.gov/cag/

ts.box *Boxplots of the Medians of Subperiods*

Description

Find the median of each time series in the time span and obtain the boxplots of the medians.

Usage

```
ts.box(x, maxbox = 200)
```

Arguments

`x`	T by k data matrix: T data points in rows with each row being data at a given time point, and k time series in columns.
`maxbox`	Maximum number of boxes. Default value is 200.

Value

Boxplots of the medians of subperiods.

Examples

```
data(TaiwanAirBox032017)
ts.box(as.matrix(TaiwanAirBox032017[,1:10]), maxbox = 10)
```

`tsBoost` *Boosting with Simple Linear Regression*

Description

It uses simple linear regression as the weak learner to perform L2 Boosting for time series data.

Usage

```
tsBoost(y, X, v = 0.01, m = 1000, rm.mean = TRUE)
```

Arguments

`y`	T by 1 scalar dependent variable.
`X`	T by k data matrix of predictors: T data points in rows with each row being data at a given time point, and k time series in columns.
`v`	Learning rate of boosting. Default value is 0.01.
`m`	Maximum number of boosting iterations. Default is 1000.
`rm.mean`	a logical command. Default is TRUE. If rm.mean=TRUE, both the dependent and predictors are mean-adjusted. If rm.mean=FALSE, no mean adjustment is made.

Value

A list containing:

- beta - the estimates of coefficient vector.
- residuals - residuals after the boosting fit.
- m - the maximum number of boosting iterations (from input).
- v - learning rate (from input).
- selection - the indexes for selected predictors. That is, the indexes for large beta estimates.
- count: the number of selected predictors.
- yhat - the fitted value of y.

Examples

```
data(TaiwanAirBox032017)
output <- tsBoost(TaiwanAirBox032017[,1], TaiwanAirBox032017[,2])
```

UMEdata20002018 *Quarterly Economic Series of the European Monetary Union*

Description

Series of Gross Domestic Product at market prices, Household and NPISH final consumption expenditure and Gross Fixed Capital Formation for the 19 Euro Area country members, a total of 57 series. The source of the data is Eurostat and the data was extracted 08-07-2019. Seasonally and calendar adjusted data. The data includes 76 quaterly observations from Q1-2000 to Q4-2018.

Usage

```
data("UMEdata20002018")
```

Format

An object of class `"data.frame"`.

Source

https://ec.europa.eu/eurostat

Index