Package ‘TSPred’

April 5, 2017

Type Package

Title Functions for Benchmarking Time Series Prediction

Version 3.0.2

Date 2017-04-05

Author Rebecca Pontes Salles [aut, cre, cph] (CEFET/RJ), Eduardo Ogasawara [ths] (CEFET/RJ)

Maintainer Rebecca Pontes Salles <rebeccapsalles@acm.org>

Description Functions for time series prediction and accuracy assessment using automatic linear modelling. The generated linear models and its yielded prediction errors can be used for benchmarking other time series prediction methods and for creating a demand for the refinement of such methods. For this purpose, benchmark data from prediction competitions may be used.

Imports forecast, KFAS, stats, MuMIn

License GPL (>= 2)

BugReports https://github.com/RebeccaSalles/TSPred/issues

URL https://sourceforge.net/p/gpca/wiki/TSPred/

NeedsCompilation no

Depends R (>= 2.10)

Repository CRAN

Date/Publication 2017-04-05 21:57:39 UTC

R topics documented:

`TSPred-package` *Functions for Benchmarking Time Series Prediction*

Description

Functions for time series prediction and accuracy assessment using automatic linear modelling. The generated linear models and its yielded prediction errors can be used for benchmarking other time series prediction methods and for creating a demand for the refinement of such methods. For this purpose, benchmark data from prediction competitions may be used.

Details

Package:	TSPred
Type:	Package
Version:	3.0.2
Date:	2017-04-05
Imports:	forecast, KFAS, stats, MuMIn
LinkingTo:	dlmodeler

License:	GPL (>= 2)
BugReports:	https://github.com/RebeccaSalles/TSPred/issues
URL:	https://sourceforge.net/p/gpca/wiki/TSPred/

Most important functions:

Automatically finding fittest linear model for prediction.

`fittestLM` `fittestArima` Automatic ARIMA fitting, prediction and accuracy evaluation.

`fittestArimaKF` Automatic ARIMA fitting and prediction with Kalman filter.

`fittestPolyR` Automatic fitting and prediction of polynomial regression.

`fittestPolyRKF` Automatic fitting and prediction of polynomial regression with Kalman filter.

`marimapred` Multiple time series automatic ARIMA fitting and prediction.

Note

The authors thank CNPq for partially sponsoring this work.

Author(s)

Rebecca Pontes Salles <rebeccapsalles@acm.org> and Eduardo Ogasawara from CEFET/RJ

See Also

`arimapred`, `marimapred`, `fittestArima`, `fittestArimaKF`, `fittestPolyR`, `fittestPolyRKF`, `fittestLM`

Examples

```
#======== Fittest linear model ========
## Not run:
data(CATS,CATS.cont)
fittest <- fittestLM(CATS[,1],CATS.cont[,1], maxorder=5, se.fit=TRUE, filtered=TRUE)

#fittest model information
fittest$rank[1,]

#predictions of the fittest model
fittest$ranked.results[[1]]$pred

## End(Not run)
#======== ==================== ========

#======== ARIMA fitting and prediction ========
#Example 1 - a single univariate time series
data(SantaFe.A,SantaFe.A.cont)
arimapred(SantaFe.A[,1],n.ahead=100)


#Example 2 - allowing the prediction of multiple univariate time series
marimapred(SantaFe.A,SantaFe.A.cont)
```

```
## Not run:
#Example 3 - automatic fitting, prediction and accuracy evaluation
data(CATS,CATS.cont)
fArima <- fittestArima(CATS[,1],CATS.cont[,1])
#predicted values
pred <- fArima$pred$pred
#model information
cbind(AICc=fArima$AICc, AIC=fArima$AIC, BIC=fArima$BIC,
 logLik=fArima$logLik, MSE=fArima$MSE, NMSE=fArima$NMSE,
 MAPE=fArima$MSE, sMAPE=fArima$MSE, MaxError=fArima$MaxError)

#plotting the time series data
plot(c(CATS[,1],CATS.cont[,1]),type='o',lwd=2,xlim=c(960,1000),ylim=c(0,200),
 xlab="Time",ylab="ARIMA")
#plotting the predicted values
lines(ts(pred,start=981),lwd=2,col='blue')
#plotting the confidence interval
lines(ts(fArima$pred$upper[,2],start=981),lwd=2,col='light blue')
lines(ts(fArima$pred$lower[,2],start=981),lwd=2,col='light blue')


#Example 4 - automatic fitting with Kalman filter, prediction and accuracy evaluation
data(CATS,CATS.cont)
fArimaKF <- fittestArimaKF(CATS[,2],CATS.cont[,2], se.fit=TRUE, filtered=TRUE)
#predicted values and estimated standard errors
pred <- fArimaKF$pred
#model information
fArimaKF$rank[1,]

#extracting Kalman filtered and smoothed time series from the best fitted model
fs <- KFAS::KFS(fArimaKF$model,filtering=c("state","mean"),smoothing=c("state","mean"))
f <- fitted(fs, filtered = TRUE) #Kalman filtered time  series
s <- fitted(fs) #Kalman smoothed time  series
#plotting the time series data
plot(c(CATS[,2],CATS.cont[,2]),type='o',lwd=2,xlim=c(960,1000),ylim=c(200,600),
 xlab="Time",ylab="ARIMAKF")
#plotting the Kalman filtered time series
lines(f,col='red',lty=2,lwd=2)
#plotting the Kalman smoothed time series
lines(s,col='green',lty=2,lwd=2)
#plotting predicted values
lines(ts(pred[,1],start=981),lwd=2,col='blue')
#plotting estimated standard errors (confidence interval)
lines(ts(pred[,1]+pred[,4],start=981),lwd=2,col='light blue')
lines(ts(pred[,1]-pred[,4],start=981),lwd=2,col='light blue')
#======== =============== ========

#======== Polynomial regression fitting and prediction ========
#Example 1 - automatic fitting, prediction and accuracy evaluation
data(CATS,CATS.cont)
fPolyR <- fittestPolyR(CATS[,1],CATS.cont[,1], maxorder=5, se.fit=TRUE)
#predicted values and estimated standard errors
```

```
pred <- fPolyR$pred
#model information
fPolyR$rank[1,]

#plotting the time series data
plot(c(CATS[,1],CATS.cont[,1]),type='o',lwd=2,xlim=c(960,1000),ylim=c(0,200),
 xlab="Time",ylab="PR")
#plotting predicted values
lines(ts(pred$fit,start=981),lwd=2,col='blue')
#plotting estimated standard errors (confidence interval)
lines(ts(pred$fit+pred$se.fit,start=981),lwd=2,col='light blue')
lines(ts(pred$fit-pred$se.fit,start=981),lwd=2,col='light blue')


#Example 2 - automatic fitting with Kalman filter, prediction and accuracy evaluation
data(CATS,CATS.cont)
fPolyRKF <- fittestPolyRKF(CATS[,1],CATS.cont[,1], maxorder=5, se.fit=TRUE, filtered=TRUE)
#predicted values and estimated standard errors
pred <- fPolyRKF$pred
#model information
fPolyRKF$rank[1,]

#extracting Kalman filtered and smoothed time series from the best fitted model
fs <- KFAS::KFS(fPolyRKF$model,filtering=c("state","mean"),smoothing=c("state","mean"))
f <- fitted(fs, filtered = TRUE) #Kalman filtered time  series
s <- fitted(fs) #Kalman smoothed time  series
#plotting the time series data
plot(c(CATS[,1],CATS.cont[,1]),type='o',lwd=2,xlim=c(960,1000),ylim=c(0,200),
 xlab="Time",ylab="PRKF")
#plotting the Kalman filtered time series
lines(f,col='red',lty=2,lwd=2)
#plotting the Kalman smoothed time series
lines(s,col='green',lty=2,lwd=2)
#plotting predicted values
lines(ts(pred[,1],start=981),lwd=2,col='blue')
#plotting estimated standard errors (confidence interval)
lines(ts(pred[,1]+pred[,4],start=981),lwd=2,col='light blue')
lines(ts(pred[,1]-pred[,4],start=981),lwd=2,col='light blue')

## End(Not run)
#======== ============================================ ========
```

---

arimainterp	*Interpolation of unknown values using automatic ARIMA fitting and prediction*

Description

The function predicts nonconsecutive blocks of N unknown values of a single time series using the `arimapred` function and an interpolation approach.

Usage

```
arimainterp(TimeSeries, n.ahead, extrap = TRUE, xreg = NULL,
newxreg = NULL, se.fit= FALSE)
```

Arguments

`TimeSeries`	A matrix, or data frame which contains a set of time series used for fitting ARIMA models. Each column corresponds to one time series. Each time series in `TimeSeries` is assumed to be a sequence of known values of the single time series that intercalates blocks of unknown values. The time series values in column 1 are lagged values of the ones in column 2, and the values in these two columns are assumed to be intercalated by the first block of N unknown values to be predicted. This is also valid for columns 2 and 3, and so forth.
`n.ahead`	A numeric value (N) with the number of consecutive unknown values of each block which is to be predicted of `TimeSeries`, that is, the length of the blocks of N unknown values.
`extrap`	A Boolean parameter which defines whether one of the blocks of N unknown values to be predicted follows the last sequence of known values in `TimeSeries`. If `extrap` is `TRUE`, the last block of N unknown values will be extrapolated from the last time series in `TimeSeries`.
`xreg`	A list of vectors, matrices, data frames or times series of external regressors used for fitting the ARIMA models. The first component of the list contains external regressors for the first time series in `TimeSeries` and therefore must have the same number of rows as this respective time series. This is also valid for the second component, and so forth. Ignored if `NULL`.
`newxreg`	A list of vectors, matrices, data frames or times series with further values of `xreg` to be used for prediction of the blocks of N unknown values. Each component of the list must have at least `n.ahead` rows. Ignored if `NULL`.
`se.fit`	If `se.fit` is `TRUE`, the standard errors of the predictions are returned.

Details

In order to avoid error accumulation, when possible, the function provides the separate prediction of each half of the blocks of unknown values using their past and future known values, respectively. If `extrap` is `TRUE`, this strategy is not possible for the last of the blocks of unknown values, for whose prediction the function uses only its past values. By default the function omits any missing values found in `TimeSeries`.

Value

A vector of time series of predictions, or if `se.fit` is `TRUE`, a vector of lists, each one with the components `pred`, the predictions, and `se`, the estimated standard errors. Both components are time series. See the `predict.Arima` function in the stats package and the function `arimapred`.

Author(s)

Rebecca Pontes Salles

References

H. Cheng, P.-N. Tan, J. Gao, and J. Scripps, 2006, "Multistep-Ahead Time Series Prediction", In: W.-K. Ng, M. Kitsuregawa, J. Li, and K. Chang, eds., Advances in Knowledge Discovery and Data Mining, Springer Berlin Heidelberg, p. 765-774.

See Also

arimapred, marimapred

Examples

```
## Not run:
data(CATS)
arimainterp(CATS[,c(2:3)],n.ahead=20,extrap=TRUE)

## End(Not run)
```

---

arimapar — *Get ARIMA model parameters.*

Description

The function returns the parameters of an automatically fitted ARIMA model, including non-seasonal and seasonal orders and drift.

Usage

```
arimapar(timeseries, na.action = na.omit, xreg = NULL)
```

Arguments

`timeseries`	A vector or univariate time series which contains the values used for fitting an ARIMA model.
`na.action`	A function for treating missing values in `timeseries`. The default function is `na.omit`, which omits any missing values found in `timeseries`.
`xreg`	A vector, matrix, data frame or times series of external regressors used for fitting the ARIMA model. It must have the same number of rows as `timeseries`. Ignored if `NULL`.

Details

The ARIMA model whose adjusted parameters are presented is automatically fitted by the `auto.arima` function in the forecast package. In order to avoid drift errors, the function introduces an auxiliary regressor whose values are a sequence of consecutive integer numbers starting from 1. For more details, see the `auto.arima` function in the `forecast` package.

Value

A numeric vector giving the number of AR, MA, seasonal AR and seasonal MA coefficients, plus the period and the number of non-seasonal and seasonal differences of the automatically fitted ARIMA model. It is also presented the value of the fitted drift constant.

Author(s)

Rebecca Pontes Salles

References

R.J. Hyndman and G. Athanasopoulos, 2013, Forecasting: principles and practice. OTexts.

R.H. Shumway and D.S. Stoffer, 2010, Time Series Analysis and Its Applications: With R Examples. 3rd ed. 2011 edition ed. New York, Springer.

See Also

marimapar, arimapred, marimapred

Examples

```
data(SantaFe.A)
arimapar(SantaFe.A[,1])
```

arimapred *Automatic ARIMA fitting and prediction*

Description

The function predicts and returns the next n consecutive values of a time series using an automatically fitted ARIMA model. It may also plot the predicted values against the actual ones using the function `plotarimapred`.

Usage

```
arimapred(timeseries, timeseries.cont = NULL, n.ahead = NULL, na.action = na.omit,
xreg = NULL, newxreg = NULL, se.fit = FALSE, plot = FALSE,
range.p = 0.2, ylab = NULL, xlab = NULL, main = NULL)
```

Arguments

`timeseries`	A vector or univariate time series which contains the values used for fitting an ARIMA model.
`timeseries.cont`	A vector or univariate time series containing a continuation for `timeseries` with actual values. Ignored if `NULL`.
`n.ahead`	Number of consecutive values of the time series, which are to be predicted. If `n.ahead` is `NULL`, the number of consecutive values to be predicted is assumed to be equal to the number of rows in `timeseries.cont`. Required when `timeseries.cont` is `NULL`.
`na.action`	A function for treating missing values in `timeseries` and `timeseries.cont`. The default function is `na.omit`, which omits any missing values found in `timeseries` or `timeseries.cont`.
`xreg`	A vector, matrix, data frame or times series of external regressors used for fitting the ARIMA model. It must have the same number of rows as `timeseries`. Ignored if `NULL`.
`newxreg`	A vector, matrix, data frame or times series with new values of `xreg` to be used for prediction. Must have at least `n.ahead` rows or the number of rows in `timeseries.cont`. Ignored if `NULL`.
`se.fit`	If `se.fit` is `TRUE`, the standard errors of the predictions are returned.
`plot`	If `plot` is `TRUE`, the function will generate a graphic of the predicted values against the actual ones in `timeseries.cont`.
`range.p`	A percentage which defines how much the range of the graphic's y-axis will be increased from the minimum limits imposed by data.
`ylab`	A title for the graphic's y-axis. Ignored if `NULL`.
`xlab`	A title for the graphic's x-axis. Ignored if `NULL`.
`main`	An overall title for the graphic. Ignored if `NULL`.

Details

The ARIMA model used for time series prediction is automatically fitted by the auto.arima function in the forecast package. In order to avoid drift errors, the function introduces an auxiliary regressor whose values are a sequence of consecutive integer numbers starting from 1. The fitted ARIMA model is used for prediction by the predict.Arima function in the stats package. For more details, see the auto.arima function in the forecast package and the predict.Arima function in the stats package.

Value

A time series of predictions, or if se.fit is TRUE, a list with the components pred, the predictions, and se, the estimated standard errors. Both components are time series. See the predict.Arima function in the stats package.

Author(s)

Rebecca Pontes Salles

References

R.J. Hyndman and G. Athanasopoulos, 2013, Forecasting: principles and practice. OTexts.

R.H. Shumway and D.S. Stoffer, 2010, Time Series Analysis and Its Applications: With R Examples. 3rd ed. 2011 edition ed. New York, Springer.

See Also

auto.arima, predict.Arima, plotarimapred, marimapred

Examples

```
data(SantaFe.A,SantaFe.A.cont)
arimapred(SantaFe.A[,1],SantaFe.A.cont[,1])
arimapred(SantaFe.A[,1],n.ahead=100)
```

CATS *Time series of the CATS Competition*

Description

A univariate artificial time series presenting 5 non-consecutive blocks of 20 unknown points.

Usage

```
data("CATS")
```

Format

A data frame with 980 observations on the following 5 variables.

- `V1` a numeric vector containing the known points 1-980 of the CATS time series.
- `V2` a numeric vector containing the known points 1001-1980 of the CATS time series.
- `V3` a numeric vector containing the known points 2001-2980 of the CATS time series.
- `V4` a numeric vector containing the known points 3001-3980 of the CATS time series.
- `V5` a numeric vector containing the known points 4001-4980 of the CATS time series.

Details

The CATS Competition presented an artificial time series with 5,000 points, among which 100 are unknown. The competition proposed that the competitors predicted the 100 unknown values from the given time series, which are grouped into five non-consecutive blocks of 20 successive values (`CATS.cont`). The unknown points of the series are the 981-1000, 1981-2000, 2981-3000, 3981-4000 and 4981-5000. The performance evaluation done by the CATS Competition was based on the MSEs computed on the 100 unknown values (E1) and on the 80 first unknown values (E2). The E2 error was considered relevant because some of the proposed methods used interpolation techniques, which cannot be applied in the case of the fifth set of unknown points.

References

A. Lendasse, E. Oja, O. Simula, M. Verleysen, and others, 2004, Time Series Prediction Competition: The CATS Benchmark, In: IJCNN'2004-International Joint Conference on Neural Networks

A. Lendasse, E. Oja, O. Simula, and M. Verleysen, 2007, Time series prediction competition: The CATS benchmark, Neurocomputing, v. 70, n. 13-15 (Aug.), p. 2325-2329.

See Also

CATS.cont

Examples

```
data(CATS)
str(CATS)
plot(ts(CATS["V5"]))
```

CATS.cont *Continuation dataset of the time series of the CATS Competition*

Description

A dataset of providing the 5 blocks of 20 unknown points of the univariate time series in CATS

Usage

```
data("CATS.cont")
```

Format

A data frame with 20 observations on the following 5 variables.

- V1 a numeric vector containing the unknown points 981-1000 of the CATS time series in CATS
- V2 a numeric vector containing the unknown points 1981-2000 of the CATS time series in CATS
- V3 a numeric vector containing the unknown points 2981-3000 of the CATS time series in CATS
- V4 a numeric vector containing the unknown points 3981-4000 of the CATS time series in CATS
- V5 a numeric vector containing the unknown points 4981-5000 of the CATS time series in CATS

Details

Contains the 100 unknown observations which were to be predicted of the CATS time series in (CATS) as demanded by the CATS Competition.

Source

A. Lendasse, E. Oja, O. Simula, M. Verleysen, and others, 2004, Time Series Prediction Competition: The CATS Benchmark, In: IJCNN'2004-International Joint Conference on Neural Networks

References

A. Lendasse, E. Oja, O. Simula, and M. Verleysen, 2007, Time series prediction competition: The CATS benchmark, Neurocomputing, v. 70, n. 13-15 (Aug.), p. 2325-2329.

See Also

CATS

Examples

```
data(CATS.cont)
str(CATS.cont)
plot(ts(CATS.cont["V5"]))
```

EUNITE.Loads *Electrical loads of the EUNITE Competition*

Description

The EUNITE Competition main dataset composed of a set of univariate time series of half-an-hour electrical loads measured between 1997 and 1998.

Usage

```
data("EUNITE.Loads")
```

Format

A data frame with 730 observations on the following 48 variables.

`X00.30` a numeric vector with loads measured in the period 00:00-00:30 of 1997-1998.

`X01.00` a numeric vector with loads measured in the period 00:30-01:00 of 1997-1998.

`X01.30` a numeric vector with loads measured in the period 01:00-01:30 of 1997-1998.

`X02.00` a numeric vector with loads measured in the period 01:30-02:00 of 1997-1998.

`X02.30` a numeric vector with loads measured in the period 02:00-02:30 of 1997-1998.

`X03.00` a numeric vector with loads measured in the period 02:30-03:00 of 1997-1998.

`X03.30` a numeric vector with loads measured in the period 03:00-03:30 of 1997-1998.

`X04.00` a numeric vector with loads measured in the period 03:30-04:00 of 1997-1998.

`X04.30` a numeric vector with loads measured in the period 04:00-04:30 of 1997-1998.

`X05.00` a numeric vector with loads measured in the period 04:30-05:00 of 1997-1998.

`X05.30` a numeric vector with loads measured in the period 05:00-05:30 of 1997-1998.

`X06.00` a numeric vector with loads measured in the period 05:30-06:00 of 1997-1998.

`X06.30` a numeric vector with loads measured in the period 06:00-06:30 of 1997-1998.

`X07.00` a numeric vector with loads measured in the period 06:30-07:00 of 1997-1998.

X07.30 a numeric vector with loads measured in the period 07:00-07:30 of 1997-1998.

X08.00 a numeric vector with loads measured in the period 07:30-08:00 of 1997-1998.

X08.30 a numeric vector with loads measured in the period 08:00-08:30 of 1997-1998.

X09.00 a numeric vector with loads measured in the period 08:30-09:00 of 1997-1998.

X09.30 a numeric vector with loads measured in the period 09:00-09:30 of 1997-1998.

X10.00 a numeric vector with loads measured in the period 09:30-10:00 of 1997-1998.

X10.30 a numeric vector with loads measured in the period 10:00-10:30 of 1997-1998.

X11.00 a numeric vector with loads measured in the period 10:30-11:00 of 1997-1998.

X11.30 a numeric vector with loads measured in the period 11:00-11:30 of 1997-1998.

X12.00 a numeric vector with loads measured in the period 11:30-12:00 of 1997-1998.

X12.30 a numeric vector with loads measured in the period 12:00-12:30 of 1997-1998.

X13.00 a numeric vector with loads measured in the period 12:30-13:00 of 1997-1998.

X13.30 a numeric vector with loads measured in the period 13:00-13:30 of 1997-1998.

X14.00 a numeric vector with loads measured in the period 13:30-14:00 of 1997-1998.

X14.30 a numeric vector with loads measured in the period 14:00-14:30 of 1997-1998.

X15.00 a numeric vector with loads measured in the period 14:30-15:00 of 1997-1998.

X15.30 a numeric vector with loads measured in the period 15:00-15:30 of 1997-1998.

X16.00 a numeric vector with loads measured in the period 15:30-16:00 of 1997-1998.

X16.30 a numeric vector with loads measured in the period 16:00-16:30 of 1997-1998.

X17.00 a numeric vector with loads measured in the period 16:30-17:00 of 1997-1998.

X17.30 a numeric vector with loads measured in the period 17:00-17:30 of 1997-1998.

X18.00 a numeric vector with loads measured in the period 17:30-18:00 of 1997-1998.

X18.30 a numeric vector with loads measured in the period 18:00-18:30 of 1997-1998.

X19.00 a numeric vector with loads measured in the period 18:30-19:00 of 1997-1998.

X19.30 a numeric vector with loads measured in the period 19:00-19:30 of 1997-1998.

X20.00 a numeric vector with loads measured in the period 19:30-20:00 of 1997-1998.

X20.30 a numeric vector with loads measured in the period 20:00-20:30 of 1997-1998.

X21.00 a numeric vector with loads measured in the period 20:30-21:00 of 1997-1998.

X21.30 a numeric vector with loads measured in the period 21:00-21:30 of 1997-1998.

X22.00 a numeric vector with loads measured in the period 21:30-22:00 of 1997-1998.

X22.30 a numeric vector with loads measured in the period 22:00-22:30 of 1997-1998.

X23.00 a numeric vector with loads measured in the period 22:30-23:00 of 1997-1998.

X23.30 a numeric vector with loads measured in the period 23:00-23:30 of 1997-1998.

X24.00 a numeric vector with loads measured in the period 23:30-24:00 of 1997-1998.

Details

The EUNITE Competition proposed the prediction of maximum daily electrical loads based on half-an-hour loads and average daily temperatures of 1997-1998 (EUNITE.Temp). The holidays with respect to this period were also provided (EUNITE.Reg) and the use of data on average daily temperatures of 1995-1996 was allowed. The dataset present considerable seasonality due to properties of electrical load demand, climate influence and holiday effects, among other reasons. Competitors were asked to predict the 31 values corresponding to the daily maximum electrical loads of January 1999 (EUNITE.Loads.cont). The performance evaluation done by the EUNITE Competition was based on the MAPE error and on the MAXIMAL error of prediction found by the competitors.

Source

EUNITE 1999, Electricity Load Forecast using Intelligent Adaptive Technology: The EUNITE Network Competition. URL: http://neuron.tuke.sk/competition/index.php.

References

B.-J. Chen, M.-W. Chang, and C.-J. Lin, 2004, Load forecasting using support vector Machines: a study on EUNITE competition 2001, IEEE Transactions on Power Systems, v. 19, n. 4 (Nov.), p. 1821-1830.

See Also

EUNITE.Loads.cont, EUNITE.Reg, EUNITE.Temp

Examples

```
data(EUNITE.Loads)
str(EUNITE.Loads)
plot(ts(EUNITE.Loads["X24.00"]))
```

EUNITE.Loads.cont   *Continuation dataset of the electrical loads of the EUNITE Competition*

Description

A dataset of univariate time series providing 31 points beyond the end of the time series in EUNITE.Loads containing half-an-hour electrical loads measured in January 1999.

Usage

```
data("EUNITE.Loads.cont")
```

Format

A data frame with 31 observations on the following 48 variables.

X00.30 a numeric vector containing further observations of X00.30 in EUNITE.Loads relative to January 1999.

X01.00 a numeric vector containing further observations of X01.00 in EUNITE.Loads relative to January 1999.

X01.30 a numeric vector containing further observations of X01.30 in EUNITE.Loads relative to January 1999.

X02.00 a numeric vector containing further observations of X02.00 in EUNITE.Loads relative to January 1999.

X02.30 a numeric vector containing further observations of X02.30 in EUNITE.Loads relative to January 1999.

X03.00 a numeric vector containing further observations of X03.00 in EUNITE.Loads relative to January 1999.

X03.30 a numeric vector containing further observations of X03.30 in EUNITE.Loads relative to January 1999.

X04.00 a numeric vector containing further observations of X04.00 in EUNITE.Loads relative to January 1999.

X04.30 a numeric vector containing further observations of X04.30 in EUNITE.Loads relative to January 1999.

X05.00 a numeric vector containing further observations of X05.00 in EUNITE.Loads relative to January 1999.

X05.30 a numeric vector containing further observations of X05.30 in EUNITE.Loads relative to January 1999.

X06.00 a numeric vector containing further observations of X06.00 in EUNITE.Loads relative to January 1999.

X06.30 a numeric vector containing further observations of X06.30 in EUNITE.Loads relative to January 1999.

X07.00 a numeric vector containing further observations of X07.00 in EUNITE.Loads relative to January 1999.

X07.30 a numeric vector containing further observations of X07.30 in EUNITE.Loads relative to January 1999.

X08.00 a numeric vector containing further observations of X08.00 in EUNITE.Loads relative to January 1999.

X08.30 a numeric vector containing further observations of X08.30 in EUNITE.Loads relative to January 1999.

X09.00 a numeric vector containing further observations of X09.00 in EUNITE.Loads relative to January 1999.

X09.30 a numeric vector containing further observations of X09.30 in EUNITE.Loads relative to January 1999.

X10.00 a numeric vector containing further observations of X10.00 in EUNITE.Loads relative to January 1999.

X10.30 a numeric vector containing further observations of X10.30 in EUNITE.Loads relative to January 1999.

X11.00 a numeric vector containing further observations of X11.00 in EUNITE.Loads relative to January 1999.

X11.30 a numeric vector containing further observations of X11.30 in EUNITE.Loads relative to January 1999.

X12.00 a numeric vector containing further observations of X12.00 in EUNITE.Loads relative to January 1999.

X12.30 a numeric vector containing further observations of X12.30 in EUNITE.Loads relative to January 1999.

X13.00 a numeric vector containing further observations of X13.00 in EUNITE.Loads relative to January 1999.

X13.30 a numeric vector containing further observations of X13.30 in EUNITE.Loads relative to January 1999.

X14.00 a numeric vector containing further observations of X14.00 in EUNITE.Loads relative to January 1999.

X14.30 a numeric vector containing further observations of X14.30 in EUNITE.Loads relative to January 1999.

X15.00 a numeric vector containing further observations of X15.00 in EUNITE.Loads relative to January 1999.

X15.30 a numeric vector containing further observations of X15.30 in EUNITE.Loads relative to January 1999.

X16.00 a numeric vector containing further observations of X16.00 in EUNITE.Loads relative to January 1999.

X16.30 a numeric vector containing further observations of X16.30 in EUNITE.Loads relative to January 1999.

X17.00 a numeric vector containing further observations of X17.00 in EUNITE.Loads relative to January 1999.

X17.30 a numeric vector containing further observations of X17.30 in EUNITE.Loads relative to January 1999.

X18.00 a numeric vector containing further observations of X18.00 in EUNITE.Loads relative to January 1999.

X18.30 a numeric vector containing further observations of X18.30 in EUNITE.Loads relative to January 1999.

X19.00 a numeric vector containing further observations of X19.00 in EUNITE.Loads relative to January 1999.

X19.30 a numeric vector containing further observations of X19.30 in EUNITE.Loads relative to January 1999.

X20.00 a numeric vector containing further observations of X20.00 in EUNITE.Loads relative to January 1999.

X20.30 a numeric vector containing further observations of X20.30 in EUNITE.Loads relative to January 1999.

X21.00 a numeric vector containing further observations of X21.00 in EUNITE.Loads relative to January 1999.

X21.30 a numeric vector containing further observations of X21.30 in EUNITE.Loads relative to January 1999.

X22.00 a numeric vector containing further observations of X22.00 in EUNITE.Loads relative to January 1999.

X22.30 a numeric vector containing further observations of X22.30 in EUNITE.Loads relative to January 1999.

X23.00 a numeric vector containing further observations of X23.00 in EUNITE.Loads relative to January 1999.

X23.30 a numeric vector containing further observations of X23.30 in EUNITE.Loads relative to January 1999.

X24.00 a numeric vector containing further observations of X24.00 in EUNITE.Loads relative to January 1999.

Details

Contains the 31 values corresponding to the daily maximum electrical loads of January 1999 which were to be predicted of EUNITE.Loads as demanded by the EUNITE Competition.

Source

EUNITE 1999, Electricity Load Forecast using Intelligent Adaptive Technology: The EUNITE Network Competition. URL: http://neuron.tuke.sk/competition/index.php.

References

B.-J. Chen, M.-W. Chang, and C.-J. Lin, 2004, Load forecasting using support vector Machines: a study on EUNITE competition 2001, IEEE Transactions on Power Systems, v. 19, n. 4 (Nov.), p. 1821-1830.

See Also

EUNITE.Loads, EUNITE.Reg, EUNITE.Temp

Examples

```
data(EUNITE.Loads.cont)
str(EUNITE.Loads.cont)
plot(ts(EUNITE.Loads.cont["X24.00"]))
```

EUNITE.Reg *Electrical loads regressors of the EUNITE Competition*

Description

The EUNITE Competition dataset containing a set of variables serving as regressors for the electrical loads measured between 1997 and 1998 in EUNITE.Loads.

Usage

```
data("EUNITE.Reg")
```

Format

A data frame with 730 observations on the following 2 variables.

- `Holiday` a numeric vector containing daily data on the holidays for the time period 1997-1998. Composed of binary values where 1 represents a holiday and 0 a common day.
- `Weekday` a numeric vector containing daily data on the weekdays for the time period 1997-1998. Composed of integer values where 1 represents a Sunday, 2 a Monday, 3 a Tuesday, 4 a Wednesday, 5 a Thursday, 6 a Friday and 7 a Saturday.

Details

The EUNITE Competition proposed the prediction of maximum daily electrical loads based on half-an-hour loads (EUNITE.Loads) and average daily temperatures of 1997-1998 (EUNITE.Temp). Competitors were asked to predict the 31 values corresponding to the daily maximum electrical loads of January 1999 (EUNITE.Loads.cont). For the posed prediction problem, it is useful to consider as regressors the holidays and the weekdays with respect to this period in EUNITE.Reg, which are expected to have a considerable impact on the electrical consumption.

Source

EUNITE 1999, Electricity Load Forecast using Intelligent Adaptive Technology: The EUNITE Network Competition. URL: http://neuron.tuke.sk/competition/index.php.

References

B.-J. Chen, M.-W. Chang, and C.-J. Lin, 2004, Load forecasting using support vector Machines: a study on EUNITE competition 2001, IEEE Transactions on Power Systems, v. 19, n. 4 (Nov.), p. 1821-1830.

See Also

EUNITE.Reg.cont, EUNITE.Loads, EUNITE.Temp

Examples

```
data(EUNITE.Reg)
str(EUNITE.Reg)
```

EUNITE.Reg.cont — *Continuation dataset of the electrical loads regressors of the EUNITE Competition*

Description

A dataset of regressor variables for electrical loads measured in January 1999, providing 31 points beyond the end of the data in EUNITE.Reg.

Usage

```
data("EUNITE.Reg.cont")
```

Format

A data frame with 31 observations on the following 2 variables.

- `Holiday` a numeric vector containing further data of the variable `Holiday` in EUNITE.Reg relative to January 1999.
- `Weekday` a numeric vector containing further data of the variable `Weekday` in EUNITE.Reg relative to January 1999.

Details

Contains the 31 values of the regressors used for the prediction of the daily maximum electrical loads of January 1999 from EUNITE.Loads as demanded by the EUNITE Competition.

Source

EUNITE 1999, Electricity Load Forecast using Intelligent Adaptive Technology: The EUNITE Network Competition. URL: http://neuron.tuke.sk/competition/index.php.

References

B.-J. Chen, M.-W. Chang, and C.-J. Lin, 2004, Load forecasting using support vector Machines: a study on EUNITE competition 2001, IEEE Transactions on Power Systems, v. 19, n. 4 (Nov.), p. 1821-1830.

See Also

EUNITE.Reg, EUNITE.Loads, EUNITE.Temp

Examples

```
data(EUNITE.Reg.cont)
str(EUNITE.Reg.cont)
```

EUNITE.Temp *Temperatures of the EUNITE Competition*

Description

The EUNITE Competition dataset composed of a univariate time series of average daily temperatures measured between 1995 and 1998.

Usage

```
data("EUNITE.Temp")
```

Format

A data frame with 1461 observations on the following variable.

`Temperature` a numeric vector with average daily temperatures measured in the period 1995-1998.

Details

The EUNITE Competition proposed the prediction of maximum daily electrical loads based on half-an-hour loads (`EUNITE.Loads`) and average daily temperatures of 1997-1998, where the latter is used as a regressor. Competitors were asked to predict the 31 values corresponding to the daily maximum electrical loads of January 1999 (`EUNITE.Loads.cont`). For the posed prediction problem, the average daily temperatures of January 1999 must also be predicted and for that, the use of data on average daily temperatures of 1995-1996 was allowed.

Source

EUNITE 1999, Electricity Load Forecast using Intelligent Adaptive Technology: The EUNITE Network Competition. URL: `http://neuron.tuke.sk/competition/index.php`.

References

B.-J. Chen, M.-W. Chang, and C.-J. Lin, 2004, Load forecasting using support vector Machines: a study on EUNITE competition 2001, IEEE Transactions on Power Systems, v. 19, n. 4 (Nov.), p. 1821-1830.

See Also

`EUNITE.Temp.cont`, `EUNITE.Loads`, `EUNITE.Reg`

Examples

```
data(EUNITE.Temp)
str(EUNITE.Temp)
plot(ts(EUNITE.Temp))
```

`EUNITE.Temp.cont` *Continuation dataset of the temperatures of the EUNITE Competition*

Description

A dataset with a univariate time series providing 31 points beyond the end of the time series in `EUNITE.Temp` containing average daily temperatures measured in January 1999.

Usage

```
data("EUNITE.Temp.cont")
```

Format

A data frame with 31 observations on the following variable.

`Temperature` a numeric vector containing further observations of `Temperature` in `EUNITE.Temp` relative to January 1999.

Details

Contains the 31 values corresponding to the average daily temperatures of January 1999 which were to be predicted of `EUNITE.Temp` as demanded by the EUNITE Competition.

Source

EUNITE 1999, Electricity Load Forecast using Intelligent Adaptive Technology: The EUNITE Network Competition. URL: http://neuron.tuke.sk/competition/index.php.

References

B.-J. Chen, M.-W. Chang, and C.-J. Lin, 2004, Load forecasting using support vector Machines: a study on EUNITE competition 2001, IEEE Transactions on Power Systems, v. 19, n. 4 (Nov.), p. 1821-1830.

See Also

`EUNITE.Temp`, `EUNITE.Loads`, `EUNITE.Reg`

Examples

```
data(EUNITE.Temp.cont)
str(EUNITE.Temp.cont)
plot(ts(EUNITE.Temp.cont))
```

`fittestArima` *Automatic ARIMA fitting, prediction and accuracy evaluation*

Description

The function predicts and returns the next n consecutive values of a univariate time series using an automatically best fitted ARIMA model. It also evaluates the fitness of the produced model, using AICc, AIC, BIC and logLik criteria, and its prediction accuracy, using the MSE, NMSE, MAPE, sMAPE and maximal error accuracy measures.

Usage

```
fittestArima(timeseries, timeseries.test, na.action = na.omit)
```

Arguments

- `timeseries` — A vector or univariate time series which contains the values used for fitting an ARIMA model.
- `timeseries.test` — A vector or univariate time series containing a continuation for `timeseries` with actual values. The number of consecutive values to be predicted is assumed to be equal to the length of `timeseries.test`. It is used as a testing set and base for calculation of prediction error measures.
- `na.action` — A function for treating missing values in `timeseries` and `timeseries.test`. The default function is `na.omit`, which omits any missing values found in `timeseries` or `timeseries.test`.

Details

The ARIMA model is automatically fitted by the `auto.arima` function and it is used for prediction by the `forecast` function both in the `forecast` package.

The fitness criteria AICc, AIC (`AIC`), BIC (`BIC`) and log-likelihood (`logLik`) are extracted from the fitted ARIMA model. Also, the prediction accuracy of the model is computed by means of MSE (`MSE`), NMSE (`NMSE`), MAPE (`MAPE`), sMAPE (`sMAPE`) and maximal error (`MAXError`) measures.

Value

A list with components:

- `model` — A list of class "ARIMA" containing the best fitted ARIMA model. See the `auto.arima` function in the `forecast` package.
- `AICc` — Numeric value of the computed AICc criterion of the fitted model.
- `AIC` — Numeric value of the computed AIC criterion of the fitted model.
- `BIC` — Numeric value of the computed BIC criterion of the fitted model.
- `logLik` — Numeric value of the computed log-likelihood of the fitted model.

pred	A list with the components pred, lower and upper, containing the predictions and the lower and upper limits for prediction intervals, respectively. All components are time series. See the forecast function in the forecast package.
MSE	Numeric value of the resulting MSE error of prediction.
NMSE	Numeric value of the resulting NMSE error of prediction.
MAPE	Numeric value of the resulting MAPE error of prediction.
sMAPE	Numeric value of the resulting sMAPE error of prediction.
MaxError	Numeric value of the maximal error of prediction.

Author(s)

Rebecca Pontes Salles

References

R.J. Hyndman and G. Athanasopoulos, 2013, Forecasting: principles and practice. OTexts.

R.H. Shumway and D.S. Stoffer, 2010, Time Series Analysis and Its Applications: With R Examples. 3rd ed. 2011 edition ed. New York, Springer.

See Also

fittestArimaKF, fittestLM, marimapred

Examples

```
## Not run:
data(CATS,CATS.cont)
fArima <- fittestArima(CATS[,1],CATS.cont[,1])
#predicted values
pred <- fArima$pred$pred
#model information
cbind(AICc=fArima$AICc, AIC=fArima$AIC, BIC=fArima$BIC,
 logLik=fArima$logLik, MSE=fArima$MSE, NMSE=fArima$NMSE,
 MAPE=fArima$MSE, sMAPE=fArima$MSE, MaxError=fArima$MaxError)

#plotting the time series data
plot(c(CATS[,1],CATS.cont[,1]),type='o',lwd=2,xlim=c(960,1000),ylim=c(0,200),
 xlab="Time",ylab="ARIMA")
#plotting the predicted values
lines(ts(pred,start=981),lwd=2,col='blue')
#plotting the confidence interval
lines(ts(fArima$pred$upper[,2],start=981),lwd=2,col='light blue')
lines(ts(fArima$pred$lower[,2],start=981),lwd=2,col='light blue')

## End(Not run)
```

fittestArimaKF — *Automatic ARIMA fitting and prediction with Kalman filter*

Description

The function predicts and returns the next n consecutive values of a univariate time series using the best evaluated ARIMA model automatically fitted with Kalman filter. It also evaluates the fitness of the produced model, using AICc, AIC, BIC and logLik criteria, and its prediction accuracy, using the MSE, NMSE, MAPE, sMAPE and maximal error accuracy measures.

Usage

```
fittestArimaKF(timeseries, timeseries.test, na.action = na.omit,
se.fit = FALSE, filtered = TRUE)
```

Arguments

`timeseries`	A vector or univariate time series which contains the values used for fitting an ARIMA model with Kalman filter.
`timeseries.test`	A vector or univariate time series containing a continuation for `timeseries` with actual values. The number of consecutive values to be predicted is assumed to be equal to the length of `timeseries.test`. It is used as a testing set and base for calculation of prediction error measures.
`na.action`	A function for treating missing values in `timeseries` and `timeseries.test`. The default function is `na.omit`, which omits any missing values found in `timeseries` or `timeseries.test`.
`se.fit`	If `se.fit` is `TRUE`, the standard errors of the predictions are returned.
`filtered`	If `filtered` is `TRUE`, Kalman filtered time series observations are used for prediction, otherwise, Kalman smoothed observations are used for prediction.

Details

A best ARIMA model is automatically fitted by the `auto.arima` function in the `forecast` package. The coefficients of this model are then used as initial parameters for optimization of a state space model (`SSModel`) using the Kalman filter and functions of the `KFAS` package (see `SSMarima` and `artransform`). A set of different ARIMA state space models is generated by different initial parameterization during the model optimization process.

The generated models are used for time series prediction. Fitness measures (AICc, AIC, BIC, log-likelihood) and prediction accuracy measures (MSE, NMSE, MAPE, sMAPE, maximal error) are calculated for each model.

The models are ranked and the best evaluated fitted ARIMA model is selected based on the lowest value of the TSPredC criterion.

The TSPredC criterion is calculated as the sum of the rank possitions of a model (1 = 1st position = better ranked model, 2 = 2nd position, etc.) on each calculated fitness or prediction accuracy measure.

Value

A list with components:

model	A list of class "SSModel" containing the best evaluated ARIMA model fitted with Kalman Filter.
AICc	Numeric value of the computed AICc criterion of the best evaluated model.
AIC	Numeric value of the computed AIC criterion of the best evaluated model.
BIC	Numeric value of the computed BIC criterion of the best evaluated model.
logLik	Numeric value of the computed log-likelihood of the best evaluated model.
pred	A time series of predictions of the best evaluated model. If `se.fit` is `TRUE`, the estimated standard errors are included. See `predict.SSModel`.
MSE	Numeric value of the resulting MSE error of prediction.
NMSE	Numeric value of the resulting NMSE error of prediction.
MAPE	Numeric value of the resulting MAPE error of prediction.
sMAPE	Numeric value of the resulting sMAPE error of prediction.
MaxError	Numeric value of the maximal error of prediction.
rank	Data.frame with the fitness and prediction accuracy criteria computed for all ARIMA with Kalman filter models considered ranked by the TSPredC value. It has the attribute `"ranked.models"`, which is a list of objects of class "SSModel" containing all the evaluated ARIMA models fitted with Kalman Filter, also ranked by the TSPredC value.

Author(s)

Rebecca Pontes Salles

References

R.J. Hyndman and G. Athanasopoulos, 2013, Forecasting: principles and practice. OTexts.

R.H. Shumway and D.S. Stoffer, 2010, Time Series Analysis and Its Applications: With R Examples. 3rd ed. 2011 edition ed. New York, Springer.

See Also

`fittestArima`, `fittestLM`, `marimapred`

Examples

```
## Not run:
data(CATS,CATS.cont)
fArimaKF <- fittestArimaKF(CATS[,2],CATS.cont[,2], se.fit=TRUE, filtered=TRUE)
#predicted values and estimated standard errors
pred <- fArimaKF$pred
#model information
fArimaKF$rank[1,]
```

```
#extracting Kalman filtered and smoothed time series from the best fitted model
fs <- KFAS::KFS(fArimaKF$model,filtering=c("state","mean"),smoothing=c("state","mean"))
f <- fitted(fs, filtered = TRUE) #Kalman filtered time  series
s <- fitted(fs) #Kalman smoothed time  series
#plotting the time series data
plot(c(CATS[,2],CATS.cont[,2]),type='o',lwd=2,xlim=c(960,1000),ylim=c(200,600),
 xlab="Time",ylab="ARIMAKF")
#plotting the Kalman filtered time series
lines(f,col='red',lty=2,lwd=2)
#plotting the Kalman smoothed time series
lines(s,col='green',lty=2,lwd=2)
#plotting predicted values
lines(ts(pred[,1],start=981),lwd=2,col='blue')
#plotting estimated standard errors (confidence interval)
lines(ts(pred[,1]+pred[,4],start=981),lwd=2,col='light blue')
lines(ts(pred[,1]-pred[,4],start=981),lwd=2,col='light blue')

## End(Not run)
```

`fittestLM` *Automatically finding fittest linear model for prediction*

Description

The function automatically evaluates and returns the fittest linear model among ARIMA and polynomial regression, with and without Kalman filtering, for prediction of a given univariate time series. Wrapper for the `fittestArima`, `fittestArimaKF`, `fittestPolyR` and `fittestPolyRKF` functions for automatic time series prediction, whose results are also returned.

Usage

```
fittestLM(timeseries, timeseries.test, maxorder = 5, na.action = na.omit,
se.fit = FALSE, filtered = TRUE)
```

Arguments

`timeseries`	A vector or univariate time series which contains the values used for fitting the models.
`timeseries.test`	A vector or univariate time series containing a continuation for `timeseries` with actual values. It is used as a testing set and base for calculation of prediction error measures. The number of consecutive values to be predicted is assumed to be equal to the length of `timeseries.test`.
`maxorder`	A numeric integer value corresponding to the maximal order of polynomial regression to be fitted and evaluated.
`na.action`	A function for treating missing values in `timeseries` and `timeseries.test`. The default function is `na.omit`, which omits any missing values found in `timeseries` or `timeseries.test`.

se.fit	If se.fit is TRUE, the standard errors of the predictions are returned.
filtered	If filtered is TRUE, Kalman filtered time series observations are used for prediction, otherwise, Kalman smoothed observations are used for prediction.

Details

The results of the best evaluated models returned by fittestArima, fittestArimaKF, fittestPolyR and fittestPolyRKF are ranked and the fittest linear model for prediction of the given univariate time series is selected based on the lowest value of the TSPredC criterion.

The TSPredC criterion is calculated as the sum of the rank possitions of a model (1 = 1st position = better ranked model, 2 = 2nd position, etc.) on each calculated fitness or prediction accuracy measure.

The evaluated measures for model fitness are the AIC, AICc, BIC and log-likelihood, and for prediction accuracy the MSE, NMSE, MAPE, sMAPE and maximal error.

Value

A list with components:

model	An object containing the fittest evaluated linear model. The class of the model object is dependent on the results of the evaluation (ranking). See fittestArima, fittestArimaKF, fittestPolyR and fittestPolyRKF.
rank	Data.frame with the fitness and prediction accuracy criteria computed for all models considered, ranked by the TSPredC value.
ranked.results	A list of lists containing the ranked results of the functions fittestArima, fittestArimaKF, fittestPolyR and fittestPolyRKF.

Author(s)

Rebecca Pontes Salles

See Also

fittestArima, fittestArimaKF, fittestPolyR, fittestPolyRKF

Examples

```
## Not run:
data(CATS,CATS.cont)
fittest <- fittestLM(CATS[,1],CATS.cont[,1], maxorder=5, se.fit=TRUE, filtered=TRUE)

#fittest model information
fittest$rank[1,]

#predictions of the fittest model
fittest$ranked.results[[1]]$pred

## End(Not run)
```

fittestPolyR — *Automatic fitting and prediction of polynomial regression*

Description

The function predicts and returns the next n consecutive values of a univariate time series using the best evaluated automatically fitted polynomial regression model. It also evaluates the fitness of the produced model, using AICc, AIC, BIC and logLik criteria, and its prediction accuracy, using the MSE, NMSE, MAPE, sMAPE and maximal error accuracy measures.

Usage

```
fittestPolyR(timeseries, timeseries.test, minorder = 0, maxorder = 5, na.action = na.omit,
se.fit = FALSE)
```

Arguments

- `timeseries` — A vector or univariate time series which contains the values used for fitting a polynomial regression model.
- `timeseries.test` — A vector or univariate time series containing a continuation for `timeseries` with actual values. The number of consecutive values to be predicted is assumed to be equal to the length of `timeseries.test`. It is used as a testing set and base for calculation of prediction error measures.
- `minorder` — A numeric integer value corresponding to the minimum order of polynomial regression to be fitted and evaluated.
- `maxorder` — A numeric integer value corresponding to the maximal order of polynomial regression to be fitted and evaluated.
- `na.action` — A function for treating missing values in `timeseries` and `timeseries.test`. The default function is `na.omit`, which omits any missing values found in `timeseries` or `timeseries.test`.
- `se.fit` — If `se.fit` is `TRUE`, the standard errors of the predictions are returned.

Details

A set with possible polynomial regression models of orders from `minorder` to `maxorder` is generated with help from the `dredge` function from the `MuMIn` package.

The set of generated models are used for time series prediction. Fitness measures (AICc, AIC, BIC, log-likelihood) and prediction accuracy measures (MSE, NMSE, MAPE, sMAPE, maximal error) are calculated for each model.

The models are ranked and the best evaluated fitted polynomial regression model is selected based on the lowest value of the TSPredC criterion.

The TSPredC criterion is calculated as the sum of the rank possitions of a model (1 = 1st position = better ranked model, 2 = 2nd position, etc.) on each calculated fitness or prediction accuracy measure.

Value

A list with components:

- `model` — An object of class "lm" containing the best evaluated polynomial regression model.
- `AICc` — Numeric value of the computed AICc criterion of the best evaluated model.
- `AIC` — Numeric value of the computed AIC criterion of the best evaluated model.
- `BIC` — Numeric value of the computed BIC criterion of the best evaluated model.
- `logLik` — Numeric value of the computed log-likelihood of the best evaluated model.
- `pred` — A time series of predictions of the best evaluated model. If `se.fit` is `TRUE`, the estimated standard errors are included.
- `MSE` — Numeric value of the resulting MSE error of prediction.
- `NMSE` — Numeric value of the resulting NMSE error of prediction.
- `MAPE` — Numeric value of the resulting MAPE error of prediction.
- `sMAPE` — Numeric value of the resulting sMAPE error of prediction.
- `MaxError` — Numeric value of the maximal error of prediction.
- `rank` — Data.frame with the coefficients and the fitness and prediction accuracy criteria computed for all polynomial regression models considered, ranked by the TSPredC value. It has the attribute `"model.calls"`, which is a list of objects of class "expression" containing the calls of all the evaluated polynomial regression models, also ranked by the TSPredC value.

Author(s)

Rebecca Pontes Salles

References

R.J. Hyndman and G. Athanasopoulos, 2013, Forecasting: principles and practice. OTexts.

R.H. Shumway and D.S. Stoffer, 2010, Time Series Analysis and Its Applications: With R Examples. 3rd ed. 2011 edition ed. New York, Springer.

See Also

`fittestPolyRKF`, `fittestLM`

Examples

```
data(CATS,CATS.cont)
fPolyR <- fittestPolyR(CATS[,1],CATS.cont[,1], maxorder=5, se.fit=TRUE)
#predicted values and estimated standard errors
pred <- fPolyR$pred
#model information
fPolyR$rank[1,]

#plotting the time series data
```

```
plot(c(CATS[,1],CATS.cont[,1]),type='o',lwd=2,xlim=c(960,1000),ylim=c(0,200),xlab="Time",ylab="PR")
#plotting predicted values
lines(ts(pred$fit,start=981),lwd=2,col='blue')
#plotting estimated standard errors (confidence interval)
lines(ts(pred$fit+pred$se.fit,start=981),lwd=2,col='light blue')
lines(ts(pred$fit-pred$se.fit,start=981),lwd=2,col='light blue')
```

`fittestPolyRKF` *Automatic fitting and prediction of polynomial regression with Kalman filter*

Description

The function predicts and returns the next n consecutive values of a univariate time series using the best evaluated polynomial regression model automatically fitted with Kalman filter. It also evaluates the fitness of the produced model, using AICc, AIC, BIC and logLik criteria, and its prediction accuracy, using the MSE, NMSE, MAPE, sMAPE and maximal error accuracy measures.

Usage

```
fittestPolyRKF(timeseries, timeseries.test, maxorder = 5, na.action = na.omit,
se.fit = FALSE, filtered = TRUE)
```

Arguments

`timeseries` A vector or univariate time series which contains the values used for fitting a polynomial regression model with Kalman filter.

`timeseries.test` A vector or univariate time series containing a continuation for `timeseries` with actual values. The number of consecutive values to be predicted is assumed to be equal to the length of `timeseries.test`. It is used as a testing set and base for calculation of prediction error measures.

`maxorder` A numeric integer value corresponding to the maximal order of polynomial regression to be fitted and evaluated.

`na.action` A function for treating missing values in `timeseries` and `timeseries.test`. The default function is `na.omit`, which omits any missing values found in `timeseries` or `timeseries.test`.

`se.fit` If `se.fit` is `TRUE`, the standard errors of the predictions are returned.

`filtered` If `filtered` is `TRUE`, Kalman filtered time series observations are used for prediction, otherwise, Kalman smoothed observations are used for prediction.

Details

A set of maxorder+1 polynomial regression models of orders from 0 to `maxorder`, respectively, is generated and represented as state space models (`SSModel`) based on code from the `dlmodeler` package. See `dlmodeler.polynomial`.

All models in the set are optimized using the Kalman filter and functions of the `KFAS` package (see `fitSSM`). Different initial parameterization during the optimization process generate different models, which are included in the set of polynomial regression models to be evaluated.

The set of generated models are used for time series prediction. Fitness measures (AICc, AIC, BIC, log-likelihood) and prediction accuracy measures (MSE, NMSE, MAPE, sMAPE, maximal error) are calculated for each model.

The models are ranked and the best evaluated fitted polynomial regression model is selected based on the lowest value of the TSPredC criterion.

The TSPredC criterion is calculated as the sum of the rank possitions of a model (1 = 1st position = better ranked model, 2 = 2nd position, etc.) on each calculated fitness or prediction accuracy measure.

Value

A list with components:

`model`	An object of class "SSModel" containing the best evaluated polynomial regression model fitted with Kalman Filter.
`AICc`	Numeric value of the computed AICc criterion of the best evaluated model.
`AIC`	Numeric value of the computed AIC criterion of the best evaluated model.
`BIC`	Numeric value of the computed BIC criterion of the best evaluated model.
`logLik`	Numeric value of the computed log-likelihood of the best evaluated model.
`pred`	A time series of predictions of the best evaluated model. If `se.fit` is `TRUE`, the estimated standard errors are included. See `predict.SSModel`.
`MSE`	Numeric value of the resulting MSE error of prediction.
`NMSE`	Numeric value of the resulting NMSE error of prediction.
`MAPE`	Numeric value of the resulting MAPE error of prediction.
`sMAPE`	Numeric value of the resulting sMAPE error of prediction.
`MaxError`	Numeric value of the maximal error of prediction.
`rank`	Data.frame with the fitness and prediction accuracy criteria computed for all polynomial regression with Kalman filter models considered ranked by the TSPredC value. It has the attribute `"ranked.models"`, which is a list of objects of class "SSModel" containing all the evaluated polynomial regression models fitted with Kalman Filter, also ranked by the TSPredC value.

Author(s)

Rebecca Pontes Salles

References

R.J. Hyndman and G. Athanasopoulos, 2013, Forecasting: principles and practice. OTexts.

R.H. Shumway and D.S. Stoffer, 2010, Time Series Analysis and Its Applications: With R Examples. 3rd ed. 2011 edition ed. New York, Springer.

See Also

fittestPolyR, fittestLM

Examples

```
## Not run:
data(CATS,CATS.cont)
fPolyRKF <- fittestPolyRKF(CATS[,1],CATS.cont[,1], maxorder=5, se.fit=TRUE, filtered=TRUE)
#predicted values and estimated standard errors
pred <- fPolyRKF$pred
#model information
fPolyRKF$rank[1,]

#extracting Kalman filtered and smoothed time series from the best fitted model
fs <- KFAS::KFS(fPolyRKF$model,filtering=c("state","mean"),smoothing=c("state","mean"))
f <- fitted(fs, filtered = TRUE) #Kalman filtered time  series
s <- fitted(fs) #Kalman smoothed time  series
#plotting the time series data
plot(c(CATS[,1],CATS.cont[,1]),type='o',lwd=2,xlim=c(960,1000),ylim=c(0,200),
 xlab="Time",ylab="PRKF")
#plotting the Kalman filtered time series
lines(f,col='red',lty=2,lwd=2)
#plotting the Kalman smoothed time series
lines(s,col='green',lty=2,lwd=2)
#plotting predicted values
lines(ts(pred[,1],start=981),lwd=2,col='blue')
#plotting estimated standard errors (confidence interval)
lines(ts(pred[,1]+pred[,4],start=981),lwd=2,col='light blue')
lines(ts(pred[,1]-pred[,4],start=981),lwd=2,col='light blue')

## End(Not run)
```

MAPE *MAPE error of prediction*

Description

The function calculates the MAPE error between actual and predicted values.

Usage

```
MAPE(actual, prediction)
```

Arguments

`actual`	A vector or univariate time series containing actual values for a time series that are to be compared against its respective predictions.
`prediction`	A vector or univariate time series containing time series predictions that are to be compared against the values in `actual`.

Value

A numeric value of the MAPE error of prediction.

Author(s)

Rebecca Pontes Salles

References

Z. Chen and Y. Yang, 2004, Assessing forecast accuracy measures, Preprint Series, n. 2004-2010, p. 2004-10.

See Also

`sMAPE`, `MSE`, `NMSE`, `MAXError`

Examples

```
data(SantaFe.A,SantaFe.A.cont)
pred <- marimapred(SantaFe.A,n.ahead=100)
MAPE(SantaFe.A.cont[,1], pred)
```

`marimapar` *Get parameters of multiple ARIMA models.*

Description

The function returns the parameters of a set of automatically fitted ARIMA models, including non-seasonal and seasonal orders and drift. Based on multiple application of the `arimapar` function.

Usage

```
marimapar(TimeSeries, na.action = na.omit, xreg = NULL)
```

Arguments

- `TimeSeries` A vector, matrix, or data frame which contains a set of time series used for fitting ARIMA models. Each column corresponds to one time series.
- `na.action` A function for treating missing values in `TimeSeries`. The default function is `na.omit`, which omits any missing values found in `TimeSeries`.
- `xreg` A vector, matrix, data frame or times series of external regressors used for fitting all the ARIMA models. It must have the same number of rows as `TimeSeries`. Ignored if `NULL`.

Details

See the `arimapar` function.

Value

A list of numeric vectors, each one giving the number of AR, MA, seasonal AR and seasonal MA coefficients, plus the period and the number of non-seasonal and seasonal differences of the automatically fitted ARIMA models. It is also presented the value of the fitted drift constants.

Author(s)

Rebecca Pontes Salles

References

See the `arimapar` function.

See Also

`arimapar`, `arimapred`, `marimapred`

Examples

```
data(SantaFe.A)
marimapar(SantaFe.A)
```

`marimapred` *Multiple time series automatic ARIMA fitting and prediction*

Description

The function predicts and returns the next n consecutive values of a set of time series using automatically fitted ARIMA models. Based on multiple application of the `arimapred` function.

Usage

```
marimapred(TimeSeries, TimeSeriesCont = NULL, n.ahead = NULL, na.action = na.omit,
xreg = NULL, newxreg = NULL, se.fit = FALSE, plot = FALSE,
range.p = 0.2, ylab = NULL, xlab = NULL, main = NULL)
```

Arguments

`TimeSeries`	A vector, matrix, or data frame which contains a set of time series used for fitting ARIMA models. Each column corresponds to one time series.
`TimeSeriesCont`	A vector, matrix, or data frame containing continuation points for `TimeSeries` with actual values. Each column corresponds to one time series. Ignored if `NULL`.
`n.ahead`	A numeric vector (or a single numeric value) with the number of consecutive values which are to be predicted of each respective time series in `TimeSeries`. If `n.ahead` is `NULL`, the number of values to be predicted of each time series in `TimeSeries` is assumed to be equal to the number of rows in each respective time series in `TimeSeriesCont`. Required when `TimeSeriesCont` is `NULL`.
`na.action`	A function for treating missing values in `TimeSeries` and `TimeSeriesCont`. The default function is `na.omit`, which omits any missing values found in `TimeSeries` or `TimeSeriesCont`.
`xreg`	A list of vectors, matrices, data frames or times series of external regressors used for fitting the ARIMA models. The first component of the list contains external regressors for the first time series in `TimeSeries` and therefore must have the same number of rows as this respective time series. This is also valid for the second component, and so forth. Ignored if `NULL`.
`newxreg`	A list of vectors, matrices, data frames or times series with new values of `xreg` to be used for prediction. The first component of the list must have at least the same number of rows as the respective first value in `n.ahead` or, if `n.ahead` is `NULL`, the number of continuation points in the respective first time series in `TimeSeriesCont`. This is also valid for the second component, and so forth. Ignored if `NULL`.
`se.fit`	If `se.fit` is `TRUE`, the standard errors of the predictions are returned.
`plot`	A Boolean parameter which defines whether the function `arimapred` will generate a graphic. If `plot` is `TRUE`, graphics will be generated for each time series in `TimeSeries`.

range.p	A percentage which defines how much the range of the graphics' y-axis will be increased from the minimum limits imposed by data.
ylab	A title for the graphics' y-axis. Ignored if NULL.
xlab	A title for the graphics' x-axis. Ignored if NULL.
main	An overall title for the graphics. Ignored if NULL.

Details

See the arimapred function.

Value

A vector of time series of predictions, if the number of consecutive values predicted of each time series in TimeSeries is the same, otherwise a list of time series of predictions.

If se.fit is TRUE, a vector of lists, each one with the components pred, the predictions, and se, the estimated standard errors. Both components are time series. See the predict.Arima function in the stats package and the function arimapred.

Author(s)

Rebecca Pontes Salles

References

See the arimapred function.

See Also

arimapred

Examples

```
data(SantaFe.A,SantaFe.A.cont)
marimapred(SantaFe.A,SantaFe.A.cont)
```

MAXError *Maximal error of prediction*

Description

The function calculates the maximal error between actual and predicted values.

Usage

```
MAXError(actual, prediction)
```

Arguments

`actual`	A vector or univariate time series containing actual values for a time series that are to be compared against its respective predictions.
`prediction`	A vector or univariate time series containing time series predictions that are to be compared against the values in `actual`.

Value

A numeric value of the maximal error of prediction.

Author(s)

Rebecca Pontes Salles

See Also

sMAPE, MAPE

Examples

```
data(SantaFe.A,SantaFe.A.cont)
pred <- marimapred(SantaFe.A,n.ahead=100)
MAXError(SantaFe.A.cont[,1], pred)
```

MSE *MSE error of prediction*

Description

The function calculates the MSE error between actual and predicted values.

Usage

```
MSE(actual, prediction)
```

Arguments

actual	A vector or univariate time series containing actual values for a time series that are to be compared against its respective predictions.
prediction	A vector or univariate time series containing time series predictions that are to be compared against the values in actual.

Value

A numeric value of the MSE error of prediction.

Author(s)

Rebecca Pontes Salles

References

Z. Chen and Y. Yang, 2004, Assessing forecast accuracy measures, Preprint Series, n. 2004-2010, p. 2004-10.

See Also

NMSE,MAPE,sMAPE, MAXError

Examples

```
data(SantaFe.A,SantaFe.A.cont)
pred <- marimapred(SantaFe.A,n.ahead=100)
MSE(SantaFe.A.cont[,1], pred)
```

NMSE *NMSE error of prediction*

Description

The function calculates the NMSE error between actual and predicted values.

Usage

```
NMSE(actual, prediction, train.actual)
```

Arguments

actual — A vector or univariate time series containing actual values for a time series that are to be compared against its respective predictions.

prediction — A vector or univariate time series containing time series predictions that are to be compared against the values in actual.

train.actual — A vector or univariate time series that was used to train the model that produced the preditions in prediction.

Value

A numeric value of the NMSE error of prediction.

Author(s)

Rebecca Pontes Salles

References

Z. Chen and Y. Yang, 2004, Assessing forecast accuracy measures, Preprint Series, n. 2004-2010, p. 2004-10.

See Also

MSE,MAPE,sMAPE, MAXError

Examples

```
data(SantaFe.A,SantaFe.A.cont)
pred <- marimapred(SantaFe.A,n.ahead=100)
NMSE(SantaFe.A.cont[,1], pred, SantaFe.A[,1])
```

NN3.A Dataset A of the NN3 Competition

Description

The NN3 Competition dataset composed of monthly time series drawn from homogeneous population of real empirical business time series.

Usage

```
data("NN3.A")
```

Format

A data frame with 126 observations on the following 111 variables.

- `NN3.001` a numeric vector containing the 51 observations of a univariate time series.
- `NN3.002` a numeric vector containing the 51 observations of a univariate time series.
- `NN3.003` a numeric vector containing the 51 observations of a univariate time series.
- `NN3.004` a numeric vector containing the 51 observations of a univariate time series.
- `NN3.005` a numeric vector containing the 51 observations of a univariate time series.
- `NN3.006` a numeric vector containing the 51 observations of a univariate time series.
- `NN3.007` a numeric vector containing the 51 observations of a univariate time series.
- `NN3.008` a numeric vector containing the 51 observations of a univariate time series.
- `NN3.009` a numeric vector containing the 51 observations of a univariate time series.
- `NN3.010` a numeric vector containing the 51 observations of a univariate time series.
- `NN3.011` a numeric vector containing the 51 observations of a univariate time series.
- `NN3.012` a numeric vector containing the 51 observations of a univariate time series.
- `NN3.013` a numeric vector containing the 51 observations of a univariate time series.
- `NN3.014` a numeric vector containing the 51 observations of a univariate time series.
- `NN3.015` a numeric vector containing the 51 observations of a univariate time series.
- `NN3.016` a numeric vector containing the 51 observations of a univariate time series.
- `NN3.017` a numeric vector containing the 51 observations of a univariate time series.
- `NN3.018` a numeric vector containing the 51 observations of a univariate time series.
- `NN3.019` a numeric vector containing the 51 observations of a univariate time series.
- `NN3.020` a numeric vector containing the 51 observations of a univariate time series.
- `NN3.021` a numeric vector containing the 51 observations of a univariate time series.
- `NN3.022` a numeric vector containing the 50 observations of a univariate time series.
- `NN3.023` a numeric vector containing the 51 observations of a univariate time series.
- `NN3.024` a numeric vector containing the 51 observations of a univariate time series.

`NN3.025` a numeric vector containing the 51 observations of a univariate time series.
`NN3.026` a numeric vector containing the 51 observations of a univariate time series.
`NN3.027` a numeric vector containing the 51 observations of a univariate time series.
`NN3.028` a numeric vector containing the 51 observations of a univariate time series.
`NN3.029` a numeric vector containing the 51 observations of a univariate time series.
`NN3.030` a numeric vector containing the 51 observations of a univariate time series.
`NN3.031` a numeric vector containing the 51 observations of a univariate time series.
`NN3.032` a numeric vector containing the 51 observations of a univariate time series.
`NN3.033` a numeric vector containing the 51 observations of a univariate time series.
`NN3.034` a numeric vector containing the 51 observations of a univariate time series.
`NN3.035` a numeric vector containing the 51 observations of a univariate time series.
`NN3.036` a numeric vector containing the 51 observations of a univariate time series.
`NN3.037` a numeric vector containing the 51 observations of a univariate time series.
`NN3.038` a numeric vector containing the 51 observations of a univariate time series.
`NN3.039` a numeric vector containing the 51 observations of a univariate time series.
`NN3.040` a numeric vector containing the 51 observations of a univariate time series.
`NN3.041` a numeric vector containing the 51 observations of a univariate time series.
`NN3.042` a numeric vector containing the 51 observations of a univariate time series.
`NN3.043` a numeric vector containing the 51 observations of a univariate time series.
`NN3.044` a numeric vector containing the 51 observations of a univariate time series.
`NN3.045` a numeric vector containing the 51 observations of a univariate time series.
`NN3.046` a numeric vector containing the 51 observations of a univariate time series.
`NN3.047` a numeric vector containing the 51 observations of a univariate time series.
`NN3.048` a numeric vector containing the 51 observations of a univariate time series.
`NN3.049` a numeric vector containing the 51 observations of a univariate time series.
`NN3.050` a numeric vector containing the 51 observations of a univariate time series.
`NN3.051` a numeric vector containing the 123 observations of a univariate time series.
`NN3.052` a numeric vector containing the 126 observations of a univariate time series.
`NN3.053` a numeric vector containing the 126 observations of a univariate time series.
`NN3.054` a numeric vector containing the 126 observations of a univariate time series.
`NN3.055` a numeric vector containing the 126 observations of a univariate time series.
`NN3.056` a numeric vector containing the 126 observations of a univariate time series.
`NN3.057` a numeric vector containing the 123 observations of a univariate time series.
`NN3.058` a numeric vector containing the 122 observations of a univariate time series.
`NN3.059` a numeric vector containing the 116 observations of a univariate time series.
`NN3.060` a numeric vector containing the 126 observations of a univariate time series.
`NN3.061` a numeric vector containing the 126 observations of a univariate time series.

NN3.062 a numeric vector containing the 122 observations of a univariate time series.
NN3.063 a numeric vector containing the 126 observations of a univariate time series.
NN3.064 a numeric vector containing the 116 observations of a univariate time series.
NN3.065 a numeric vector containing the 126 observations of a univariate time series.
NN3.066 a numeric vector containing the 126 observations of a univariate time series.
NN3.067 a numeric vector containing the 126 observations of a univariate time series.
NN3.068 a numeric vector containing the 121 observations of a univariate time series.
NN3.069 a numeric vector containing the 121 observations of a univariate time series.
NN3.070 a numeric vector containing the 121 observations of a univariate time series.
NN3.071 a numeric vector containing the 126 observations of a univariate time series.
NN3.072 a numeric vector containing the 126 observations of a univariate time series.
NN3.073 a numeric vector containing the 126 observations of a univariate time series.
NN3.074 a numeric vector containing the 126 observations of a univariate time series.
NN3.075 a numeric vector containing the 126 observations of a univariate time series.
NN3.076 a numeric vector containing the 126 observations of a univariate time series.
NN3.077 a numeric vector containing the 126 observations of a univariate time series.
NN3.078 a numeric vector containing the 126 observations of a univariate time series.
NN3.079 a numeric vector containing the 126 observations of a univariate time series.
NN3.080 a numeric vector containing the 126 observations of a univariate time series.
NN3.081 a numeric vector containing the 126 observations of a univariate time series.
NN3.082 a numeric vector containing the 116 observations of a univariate time series.
NN3.083 a numeric vector containing the 126 observations of a univariate time series.
NN3.084 a numeric vector containing the 122 observations of a univariate time series.
NN3.085 a numeric vector containing the 126 observations of a univariate time series.
NN3.086 a numeric vector containing the 126 observations of a univariate time series.
NN3.087 a numeric vector containing the 121 observations of a univariate time series.
NN3.088 a numeric vector containing the 78 observations of a univariate time series.
NN3.089 a numeric vector containing the 126 observations of a univariate time series.
NN3.090 a numeric vector containing the 126 observations of a univariate time series.
NN3.091 a numeric vector containing the 116 observations of a univariate time series.
NN3.092 a numeric vector containing the 115 observations of a univariate time series.
NN3.093 a numeric vector containing the 126 observations of a univariate time series.
NN3.094 a numeric vector containing the 116 observations of a univariate time series.
NN3.095 a numeric vector containing the 126 observations of a univariate time series.
NN3.096 a numeric vector containing the 126 observations of a univariate time series.
NN3.097 a numeric vector containing the 126 observations of a univariate time series.
NN3.098 a numeric vector containing the 126 observations of a univariate time series.

NN3.099 a numeric vector containing the 115 observations of a univariate time series.

NN3.100 a numeric vector containing the 116 observations of a univariate time series.

NN3_101 a numeric vector containing the 126 observations of a univariate time series.

NN3_102 a numeric vector containing the 126 observations of a univariate time series.

NN3_103 a numeric vector containing the 126 observations of a univariate time series.

NN3_104 a numeric vector containing the 115 observations of a univariate time series.

NN3_105 a numeric vector containing the 126 observations of a univariate time series.

NN3_106 a numeric vector containing the 126 observations of a univariate time series.

NN3_107 a numeric vector containing the 126 observations of a univariate time series.

NN3_108 a numeric vector containing the 115 observations of a univariate time series.

NN3_109 a numeric vector containing the 123 observations of a univariate time series.

NN3_110 a numeric vector containing the 126 observations of a univariate time series.

NN3_111 a numeric vector containing the 126 observations of a univariate time series.

Details

The NN3 Competition's Dataset A contains 111 different monthly time series. Each of this time series possess from 50 to 126 observations. Each competitor in NN3 was asked to predict the next 18 corresponding observations of each times series (NN3.A.cont). The performance evaluation done by NN3 Competition was based on the mean SMAPE error of prediction found by the competitors across all time series.

Source

NN3 2007, The NN3 Competition: Forecasting competition for artificial neural networks and computational intelligence. URL: http://www.neural-forecasting-competition.com/NN3/index.htm.

References

S.F. Crone, M. Hibon, and K. Nikolopoulos, 2011, Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction, International Journal of Forecasting, v. 27, n. 3 (Jul.), p. 635-660.

See Also

NN3.A.cont

Examples

```
data(NN3.A)
str(NN3.A)
plot(ts(NN3.A["NN3_111"]))
```

`NN3.A.cont` *Continuation dataset of the Dataset A of the NN3 Competition*

Description

A dataset of univariate time series providing 18 points beyond the end of the time series in NN3.A.

Usage

```
data("NN3.A.cont")
```

Format

A data frame with 18 observations on the following 111 variables.

NN3.001 a numeric vector containing further observations of NN3.001 in NN3.A.
NN3.002 a numeric vector containing further observations of NN3.002 in NN3.A.
NN3.003 a numeric vector containing further observations of NN3.003 in NN3.A.
NN3.004 a numeric vector containing further observations of NN3.004 in NN3.A.
NN3.005 a numeric vector containing further observations of NN3.005 in NN3.A.
NN3.006 a numeric vector containing further observations of NN3.006 in NN3.A.
NN3.007 a numeric vector containing further observations of NN3.007 in NN3.A.
NN3.008 a numeric vector containing further observations of NN3.008 in NN3.A.
NN3.009 a numeric vector containing further observations of NN3.009 in NN3.A.
NN3.010 a numeric vector containing further observations of NN3.010 in NN3.A.
NN3.011 a numeric vector containing further observations of NN3.011 in NN3.A.
NN3.012 a numeric vector containing further observations of NN3.012 in NN3.A.
NN3.013 a numeric vector containing further observations of NN3.013 in NN3.A.
NN3.014 a numeric vector containing further observations of NN3.014 in NN3.A.
NN3.015 a numeric vector containing further observations of NN3.015 in NN3.A.
NN3.016 a numeric vector containing further observations of NN3.016 in NN3.A.
NN3.017 a numeric vector containing further observations of NN3.017 in NN3.A.
NN3.018 a numeric vector containing further observations of NN3.018 in NN3.A.
NN3.019 a numeric vector containing further observations of NN3.019 in NN3.A.
NN3.020 a numeric vector containing further observations of NN3.020 in NN3.A.
NN3.021 a numeric vector containing further observations of NN3.021 in NN3.A.
NN3.022 a numeric vector containing further observations of NN3.022 in NN3.A.
NN3.023 a numeric vector containing further observations of NN3.023 in NN3.A.
NN3.024 a numeric vector containing further observations of NN3.024 in NN3.A.
NN3.025 a numeric vector containing further observations of NN3.025 in NN3.A.

NN3.026 a numeric vector containing further observations of NN3.026 in NN3.A.
NN3.027 a numeric vector containing further observations of NN3.027 in NN3.A.
NN3.028 a numeric vector containing further observations of NN3.028 in NN3.A.
NN3.029 a numeric vector containing further observations of NN3.029 in NN3.A.
NN3.030 a numeric vector containing further observations of NN3.030 in NN3.A.
NN3.031 a numeric vector containing further observations of NN3.031 in NN3.A.
NN3.032 a numeric vector containing further observations of NN3.032 in NN3.A.
NN3.033 a numeric vector containing further observations of NN3.033 in NN3.A.
NN3.034 a numeric vector containing further observations of NN3.034 in NN3.A.
NN3.035 a numeric vector containing further observations of NN3.035 in NN3.A.
NN3.036 a numeric vector containing further observations of NN3.036 in NN3.A.
NN3.037 a numeric vector containing further observations of NN3.037 in NN3.A.
NN3.038 a numeric vector containing further observations of NN3.038 in NN3.A.
NN3.039 a numeric vector containing further observations of NN3.039 in NN3.A.
NN3.040 a numeric vector containing further observations of NN3.040 in NN3.A.
NN3.041 a numeric vector containing further observations of NN3.041 in NN3.A.
NN3.042 a numeric vector containing further observations of NN3.042 in NN3.A.
NN3.043 a numeric vector containing further observations of NN3.043 in NN3.A.
NN3.044 a numeric vector containing further observations of NN3.044 in NN3.A.
NN3.045 a numeric vector containing further observations of NN3.045 in NN3.A.
NN3.046 a numeric vector containing further observations of NN3.046 in NN3.A.
NN3.047 a numeric vector containing further observations of NN3.047 in NN3.A.
NN3.048 a numeric vector containing further observations of NN3.048 in NN3.A.
NN3.049 a numeric vector containing further observations of NN3.049 in NN3.A.
NN3.050 a numeric vector containing further observations of NN3.050 in NN3.A.
NN3.051 a numeric vector containing further observations of NN3.051 in NN3.A.
NN3.052 a numeric vector containing further observations of NN3.052 in NN3.A.
NN3.053 a numeric vector containing further observations of NN3.053 in NN3.A.
NN3.054 a numeric vector containing further observations of NN3.054 in NN3.A.
NN3.055 a numeric vector containing further observations of NN3.055 in NN3.A.
NN3.056 a numeric vector containing further observations of NN3.056 in NN3.A.
NN3.057 a numeric vector containing further observations of NN3.057 in NN3.A.
NN3.058 a numeric vector containing further observations of NN3.058 in NN3.A.
NN3.059 a numeric vector containing further observations of NN3.059 in NN3.A.
NN3.060 a numeric vector containing further observations of NN3.060 in NN3.A.
NN3.061 a numeric vector containing further observations of NN3.061 in NN3.A.
NN3.062 a numeric vector containing further observations of NN3.062 in NN3.A.

NN3.063 a numeric vector containing further observations of NN3.063 in NN3.A.

NN3.064 a numeric vector containing further observations of NN3.064 in NN3.A.

NN3.065 a numeric vector containing further observations of NN3.065 in NN3.A.

NN3.066 a numeric vector containing further observations of NN3.066 in NN3.A.

NN3.067 a numeric vector containing further observations of NN3.067 in NN3.A.

NN3.068 a numeric vector containing further observations of NN3.068 in NN3.A.

NN3.069 a numeric vector containing further observations of NN3.069 in NN3.A.

NN3.070 a numeric vector containing further observations of NN3.070 in NN3.A.

NN3.071 a numeric vector containing further observations of NN3.071 in NN3.A.

NN3.072 a numeric vector containing further observations of NN3.072 in NN3.A.

NN3.073 a numeric vector containing further observations of NN3.073 in NN3.A.

NN3.074 a numeric vector containing further observations of NN3.074 in NN3.A.

NN3.075 a numeric vector containing further observations of NN3.075 in NN3.A.

NN3.076 a numeric vector containing further observations of NN3.076 in NN3.A.

NN3.077 a numeric vector containing further observations of NN3.077 in NN3.A.

NN3.078 a numeric vector containing further observations of NN3.078 in NN3.A.

NN3.079 a numeric vector containing further observations of NN3.079 in NN3.A.

NN3.080 a numeric vector containing further observations of NN3.080 in NN3.A.

NN3.081 a numeric vector containing further observations of NN3.081 in NN3.A.

NN3.082 a numeric vector containing further observations of NN3.082 in NN3.A.

NN3.083 a numeric vector containing further observations of NN3.083 in NN3.A.

NN3.084 a numeric vector containing further observations of NN3.084 in NN3.A.

NN3.085 a numeric vector containing further observations of NN3.085 in NN3.A.

NN3.086 a numeric vector containing further observations of NN3.086 in NN3.A.

NN3.087 a numeric vector containing further observations of NN3.087 in NN3.A.

NN3.088 a numeric vector containing further observations of NN3.088 in NN3.A.

NN3.089 a numeric vector containing further observations of NN3.089 in NN3.A.

NN3.090 a numeric vector containing further observations of NN3.090 in NN3.A.

NN3.091 a numeric vector containing further observations of NN3.091 in NN3.A.

NN3.092 a numeric vector containing further observations of NN3.092 in NN3.A.

NN3.093 a numeric vector containing further observations of NN3.093 in NN3.A.

NN3.094 a numeric vector containing further observations of NN3.094 in NN3.A.

NN3.095 a numeric vector containing further observations of NN3.095 in NN3.A.

NN3.096 a numeric vector containing further observations of NN3.096 in NN3.A.

NN3.097 a numeric vector containing further observations of NN3.097 in NN3.A.

NN3.098 a numeric vector containing further observations of NN3.098 in NN3.A.

NN3.099 a numeric vector containing further observations of NN3.099 in NN3.A.

NN3.100 a numeric vector containing further observations of NN3.100 in NN3.A.

NN3_101 a numeric vector containing further observations of NN3_101 in NN3.A.

NN3_102 a numeric vector containing further observations of NN3_102 in NN3.A.

NN3_103 a numeric vector containing further observations of NN3_103 in NN3.A.

NN3_104 a numeric vector containing further observations of NN3_104 in NN3.A.

NN3_105 a numeric vector containing further observations of NN3_105 in NN3.A.

NN3_106 a numeric vector containing further observations of NN3_106 in NN3.A.

NN3_107 a numeric vector containing further observations of NN3_107 in NN3.A.

NN3_108 a numeric vector containing further observations of NN3_108 in NN3.A.

NN3_109 a numeric vector containing further observations of NN3_109 in NN3.A.

NN3_110 a numeric vector containing further observations of NN3_110 in NN3.A.

NN3_111 a numeric vector containing further observations of NN3_111 in NN3.A.

Details

Contains the 18 observations which were to be predicted of each time series in Dataset A (NN3.A) as demanded by the NN3 Competition.

Source

NN3 2007, The NN3 Competition: Forecasting competition for artificial neural networks and computational intelligence. URL: http://www.neural-forecasting-competition.com/NN3/index.htm.

References

S.F. Crone, M. Hibon, and K. Nikolopoulos, 2011, Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction, International Journal of Forecasting, v. 27, n. 3 (Jul.), p. 635-660.

See Also

NN3.A

Examples

```
data(NN3.A.cont)
str(NN3.A.cont)
plot(ts(NN3.A.cont["NN3_111"]))
```

`NN5.A` *Dataset A of the NN5 Competition*

Description

The NN5 Competition dataset composed of daily time series originated from the observation of daily withdrawals at 111 randomly selected different cash machines at different locations within England.

Usage

```
data("NN5.A")
```

Format

A data frame with 735 observations on the following 111 variables.

`NN5.001` a numeric vector containing observations of a univariate time series.

`NN5.002` a numeric vector containing observations of a univariate time series.

`NN5.003` a numeric vector containing observations of a univariate time series.

`NN5.004` a numeric vector containing observations of a univariate time series.

`NN5.005` a numeric vector containing observations of a univariate time series.

`NN5.006` a numeric vector containing observations of a univariate time series.

`NN5.007` a numeric vector containing observations of a univariate time series.

`NN5.008` a numeric vector containing observations of a univariate time series.

`NN5.009` a numeric vector containing observations of a univariate time series.

`NN5.010` a numeric vector containing observations of a univariate time series.

`NN5.011` a numeric vector containing observations of a univariate time series.

`NN5.012` a numeric vector containing observations of a univariate time series.

`NN5.013` a numeric vector containing observations of a univariate time series.

`NN5.014` a numeric vector containing observations of a univariate time series.

`NN5.015` a numeric vector containing observations of a univariate time series.

`NN5.016` a numeric vector containing observations of a univariate time series.

`NN5.017` a numeric vector containing observations of a univariate time series.

`NN5.018` a numeric vector containing observations of a univariate time series.

`NN5.019` a numeric vector containing observations of a univariate time series.

`NN5.020` a numeric vector containing observations of a univariate time series.

`NN5.021` a numeric vector containing observations of a univariate time series.

`NN5.022` a numeric vector containing observations of a univariate time series.

`NN5.023` a numeric vector containing observations of a univariate time series.

NN5.024 a numeric vector containing observations of a univariate time series.
NN5.025 a numeric vector containing observations of a univariate time series.
NN5.026 a numeric vector containing observations of a univariate time series.
NN5.027 a numeric vector containing observations of a univariate time series.
NN5.028 a numeric vector containing observations of a univariate time series.
NN5.029 a numeric vector containing observations of a univariate time series.
NN5.030 a numeric vector containing observations of a univariate time series.
NN5.031 a numeric vector containing observations of a univariate time series.
NN5.032 a numeric vector containing observations of a univariate time series.
NN5.033 a numeric vector containing observations of a univariate time series.
NN5.034 a numeric vector containing observations of a univariate time series.
NN5.035 a numeric vector containing observations of a univariate time series.
NN5.036 a numeric vector containing observations of a univariate time series.
NN5.037 a numeric vector containing observations of a univariate time series.
NN5.038 a numeric vector containing observations of a univariate time series.
NN5.039 a numeric vector containing observations of a univariate time series.
NN5.040 a numeric vector containing observations of a univariate time series.
NN5.041 a numeric vector containing observations of a univariate time series.
NN5.042 a numeric vector containing observations of a univariate time series.
NN5.043 a numeric vector containing observations of a univariate time series.
NN5.044 a numeric vector containing observations of a univariate time series.
NN5.045 a numeric vector containing observations of a univariate time series.
NN5.046 a numeric vector containing observations of a univariate time series.
NN5.047 a numeric vector containing observations of a univariate time series.
NN5.048 a numeric vector containing observations of a univariate time series.
NN5.049 a numeric vector containing observations of a univariate time series.
NN5.050 a numeric vector containing observations of a univariate time series.
NN5.051 a numeric vector containing observations of a univariate time series.
NN5.052 a numeric vector containing observations of a univariate time series.
NN5.053 a numeric vector containing observations of a univariate time series.
NN5.054 a numeric vector containing observations of a univariate time series.
NN5.055 a numeric vector containing observations of a univariate time series.
NN5.056 a numeric vector containing observations of a univariate time series.
NN5.057 a numeric vector containing observations of a univariate time series.
NN5.058 a numeric vector containing observations of a univariate time series.
NN5.059 a numeric vector containing observations of a univariate time series.
NN5.060 a numeric vector containing observations of a univariate time series.

NN5.061 a numeric vector containing observations of a univariate time series.
NN5.062 a numeric vector containing observations of a univariate time series.
NN5.063 a numeric vector containing observations of a univariate time series.
NN5.064 a numeric vector containing observations of a univariate time series.
NN5.065 a numeric vector containing observations of a univariate time series.
NN5.066 a numeric vector containing observations of a univariate time series.
NN5.067 a numeric vector containing observations of a univariate time series.
NN5.068 a numeric vector containing observations of a univariate time series.
NN5.069 a numeric vector containing observations of a univariate time series.
NN5.070 a numeric vector containing observations of a univariate time series.
NN5.071 a numeric vector containing observations of a univariate time series.
NN5.072 a numeric vector containing observations of a univariate time series.
NN5.073 a numeric vector containing observations of a univariate time series.
NN5.074 a numeric vector containing observations of a univariate time series.
NN5.075 a numeric vector containing observations of a univariate time series.
NN5.076 a numeric vector containing observations of a univariate time series.
NN5.077 a numeric vector containing observations of a univariate time series.
NN5.078 a numeric vector containing observations of a univariate time series.
NN5.079 a numeric vector containing observations of a univariate time series.
NN5.080 a numeric vector containing observations of a univariate time series.
NN5.081 a numeric vector containing observations of a univariate time series.
NN5.082 a numeric vector containing observations of a univariate time series.
NN5.083 a numeric vector containing observations of a univariate time series.
NN5.084 a numeric vector containing observations of a univariate time series.
NN5.085 a numeric vector containing observations of a univariate time series.
NN5.086 a numeric vector containing observations of a univariate time series.
NN5.087 a numeric vector containing observations of a univariate time series.
NN5.088 a numeric vector containing observations of a univariate time series.
NN5.089 a numeric vector containing observations of a univariate time series.
NN5.090 a numeric vector containing observations of a univariate time series.
NN5.091 a numeric vector containing observations of a univariate time series.
NN5.092 a numeric vector containing observations of a univariate time series.
NN5.093 a numeric vector containing observations of a univariate time series.
NN5.094 a numeric vector containing observations of a univariate time series.
NN5.095 a numeric vector containing observations of a univariate time series.
NN5.096 a numeric vector containing observations of a univariate time series.
NN5.097 a numeric vector containing observations of a univariate time series.

`NN5.098` a numeric vector containing observations of a univariate time series.

`NN5.099` a numeric vector containing observations of a univariate time series.

`NN5.100` a numeric vector containing observations of a univariate time series.

`NN5.101` a numeric vector containing observations of a univariate time series.

`NN5.102` a numeric vector containing observations of a univariate time series.

`NN5.103` a numeric vector containing observations of a univariate time series.

`NN5.104` a numeric vector containing observations of a univariate time series.

`NN5.105` a numeric vector containing observations of a univariate time series.

`NN5.106` a numeric vector containing observations of a univariate time series.

`NN5.107` a numeric vector containing observations of a univariate time series.

`NN5.108` a numeric vector containing observations of a univariate time series.

`NN5.109` a numeric vector containing observations of a univariate time series.

`NN5.110` a numeric vector containing observations of a univariate time series.

`NN5.111` a numeric vector containing observations of a univariate time series.

Details

The NN5 Competition's Dataset A contains 111 different daily time series. Each of these time series possesses 735 observations, and may present missing data. The time series also show different patterns of single or multiple overlying seasonal properties. Each competitor in NN5 was asked to predict the next 56 corresponding observations of each times series (`NN5.A.cont`). The performance evaluation done by NN5 Competition was based on the mean SMAPE error of prediction found by the competitors across all time series.

Source

NN5 2008, The NN5 Competition: Forecasting competition for artificial neural networks and computational intelligence. URL: `http://www.neural-forecasting-competition.com/NN5/index.htm`.

References

S.F. Crone, 2008, Results of the NN5 time series forecasting competition. Hong Kong, Presentation at the IEEE world congress on computational intelligence. WCCI'2008.

See Also

`NN5.A.cont`

Examples

```
data(NN5.A)
str(NN5.A)
plot(ts(NN5.A["NN5.111"]))
```

NN5.A.cont *Continuation dataset of the Dataset A of the NN5 Competition*

Description

A dataset of univariate time series providing 56 points beyond the end of the time series in NN5.A.

Usage

```
data("NN5.A.cont")
```

Format

A data frame with 56 observations on the following 111 variables.

NN5.001 a numeric vector containing further observations of NN5.001 in NN5.A.
NN5.002 a numeric vector containing further observations of NN5.002 in NN5.A.
NN5.003 a numeric vector containing further observations of NN5.003 in NN5.A.
NN5.004 a numeric vector containing further observations of NN5.004 in NN5.A.
NN5.005 a numeric vector containing further observations of NN5.005 in NN5.A.
NN5.006 a numeric vector containing further observations of NN5.006 in NN5.A.
NN5.007 a numeric vector containing further observations of NN5.007 in NN5.A.
NN5.008 a numeric vector containing further observations of NN5.008 in NN5.A.
NN5.009 a numeric vector containing further observations of NN5.009 in NN5.A.
NN5.010 a numeric vector containing further observations of NN5.010 in NN5.A.
NN5.011 a numeric vector containing further observations of NN5.011 in NN5.A.
NN5.012 a numeric vector containing further observations of NN5.012 in NN5.A.
NN5.013 a numeric vector containing further observations of NN5.013 in NN5.A.
NN5.014 a numeric vector containing further observations of NN5.014 in NN5.A.
NN5.015 a numeric vector containing further observations of NN5.015 in NN5.A.
NN5.016 a numeric vector containing further observations of NN5.016 in NN5.A.
NN5.017 a numeric vector containing further observations of NN5.017 in NN5.A.
NN5.018 a numeric vector containing further observations of NN5.018 in NN5.A.
NN5.019 a numeric vector containing further observations of NN5.019 in NN5.A.
NN5.020 a numeric vector containing further observations of NN5.020 in NN5.A.
NN5.021 a numeric vector containing further observations of NN5.021 in NN5.A.
NN5.022 a numeric vector containing further observations of NN5.022 in NN5.A.
NN5.023 a numeric vector containing further observations of NN5.023 in NN5.A.
NN5.024 a numeric vector containing further observations of NN5.024 in NN5.A.
NN5.025 a numeric vector containing further observations of NN5.025 in NN5.A.

NN5.026 a numeric vector containing further observations of NN5.026 in NN5.A.

NN5.027 a numeric vector containing further observations of NN5.027 in NN5.A.

NN5.028 a numeric vector containing further observations of NN5.028 in NN5.A.

NN5.029 a numeric vector containing further observations of NN5.029 in NN5.A.

NN5.030 a numeric vector containing further observations of NN5.030 in NN5.A.

NN5.031 a numeric vector containing further observations of NN5.031 in NN5.A.

NN5.032 a numeric vector containing further observations of NN5.032 in NN5.A.

NN5.033 a numeric vector containing further observations of NN5.033 in NN5.A.

NN5.034 a numeric vector containing further observations of NN5.034 in NN5.A.

NN5.035 a numeric vector containing further observations of NN5.035 in NN5.A.

NN5.036 a numeric vector containing further observations of NN5.036 in NN5.A.

NN5.037 a numeric vector containing further observations of NN5.037 in NN5.A.

NN5.038 a numeric vector containing further observations of NN5.038 in NN5.A.

NN5.039 a numeric vector containing further observations of NN5.039 in NN5.A.

NN5.040 a numeric vector containing further observations of NN5.040 in NN5.A.

NN5.041 a numeric vector containing further observations of NN5.041 in NN5.A.

NN5.042 a numeric vector containing further observations of NN5.042 in NN5.A.

NN5.043 a numeric vector containing further observations of NN5.043 in NN5.A.

NN5.044 a numeric vector containing further observations of NN5.044 in NN5.A.

NN5.045 a numeric vector containing further observations of NN5.045 in NN5.A.

NN5.046 a numeric vector containing further observations of NN5.046 in NN5.A.

NN5.047 a numeric vector containing further observations of NN5.047 in NN5.A.

NN5.048 a numeric vector containing further observations of NN5.048 in NN5.A.

NN5.049 a numeric vector containing further observations of NN5.049 in NN5.A.

NN5.050 a numeric vector containing further observations of NN5.050 in NN5.A.

NN5.051 a numeric vector containing further observations of NN5.051 in NN5.A.

NN5.052 a numeric vector containing further observations of NN5.052 in NN5.A.

NN5.053 a numeric vector containing further observations of NN5.053 in NN5.A.

NN5.054 a numeric vector containing further observations of NN5.054 in NN5.A.

NN5.055 a numeric vector containing further observations of NN5.055 in NN5.A.

NN5.056 a numeric vector containing further observations of NN5.056 in NN5.A.

NN5.057 a numeric vector containing further observations of NN5.057 in NN5.A.

NN5.058 a numeric vector containing further observations of NN5.058 in NN5.A.

NN5.059 a numeric vector containing further observations of NN5.059 in NN5.A.

NN5.060 a numeric vector containing further observations of NN5.060 in NN5.A.

NN5.061 a numeric vector containing further observations of NN5.061 in NN5.A.

NN5.062 a numeric vector containing further observations of NN5.062 in NN5.A.

NN5.063 a numeric vector containing further observations of NN5.063 in NN5.A.
NN5.064 a numeric vector containing further observations of NN5.064 in NN5.A.
NN5.065 a numeric vector containing further observations of NN5.065 in NN5.A.
NN5.066 a numeric vector containing further observations of NN5.066 in NN5.A.
NN5.067 a numeric vector containing further observations of NN5.067 in NN5.A.
NN5.068 a numeric vector containing further observations of NN5.068 in NN5.A.
NN5.069 a numeric vector containing further observations of NN5.069 in NN5.A.
NN5.070 a numeric vector containing further observations of NN5.070 in NN5.A.
NN5.071 a numeric vector containing further observations of NN5.071 in NN5.A.
NN5.072 a numeric vector containing further observations of NN5.072 in NN5.A.
NN5.073 a numeric vector containing further observations of NN5.073 in NN5.A.
NN5.074 a numeric vector containing further observations of NN5.074 in NN5.A.
NN5.075 a numeric vector containing further observations of NN5.075 in NN5.A.
NN5.076 a numeric vector containing further observations of NN5.076 in NN5.A.
NN5.077 a numeric vector containing further observations of NN5.077 in NN5.A.
NN5.078 a numeric vector containing further observations of NN5.078 in NN5.A.
NN5.079 a numeric vector containing further observations of NN5.079 in NN5.A.
NN5.080 a numeric vector containing further observations of NN5.080 in NN5.A.
NN5.081 a numeric vector containing further observations of NN5.081 in NN5.A.
NN5.082 a numeric vector containing further observations of NN5.082 in NN5.A.
NN5.083 a numeric vector containing further observations of NN5.083 in NN5.A.
NN5.084 a numeric vector containing further observations of NN5.084 in NN5.A.
NN5.085 a numeric vector containing further observations of NN5.085 in NN5.A.
NN5.086 a numeric vector containing further observations of NN5.086 in NN5.A.
NN5.087 a numeric vector containing further observations of NN5.087 in NN5.A.
NN5.088 a numeric vector containing further observations of NN5.088 in NN5.A.
NN5.089 a numeric vector containing further observations of NN5.089 in NN5.A.
NN5.090 a numeric vector containing further observations of NN5.090 in NN5.A.
NN5.091 a numeric vector containing further observations of NN5.091 in NN5.A.
NN5.092 a numeric vector containing further observations of NN5.092 in NN5.A.
NN5.093 a numeric vector containing further observations of NN5.093 in NN5.A.
NN5.094 a numeric vector containing further observations of NN5.094 in NN5.A.
NN5.095 a numeric vector containing further observations of NN5.095 in NN5.A.
NN5.096 a numeric vector containing further observations of NN5.096 in NN5.A.
NN5.097 a numeric vector containing further observations of NN5.097 in NN5.A.
NN5.098 a numeric vector containing further observations of NN5.098 in NN5.A.
NN5.099 a numeric vector containing further observations of NN5.099 in NN5.A.

NN5.100 a numeric vector containing further observations of NN5.100 in NN5.A.

NN5.101 a numeric vector containing further observations of NN5.101 in NN5.A.

NN5.102 a numeric vector containing further observations of NN5.102 in NN5.A.

NN5.103 a numeric vector containing further observations of NN5.103 in NN5.A.

NN5.104 a numeric vector containing further observations of NN5.104 in NN5.A.

NN5.105 a numeric vector containing further observations of NN5.105 in NN5.A.

NN5.106 a numeric vector containing further observations of NN5.106 in NN5.A.

NN5.107 a numeric vector containing further observations of NN5.107 in NN5.A.

NN5.108 a numeric vector containing further observations of NN5.108 in NN5.A.

NN5.109 a numeric vector containing further observations of NN5.109 in NN5.A.

NN5.110 a numeric vector containing further observations of NN5.110 in NN5.A.

NN5.111 a numeric vector containing further observations of NN5.111 in NN5.A.

Details

Contains the 56 observations which were to be predicted of each time series in Dataset A (NN5.A) as demanded by the NN5 Competition.

Source

NN5 2008, The NN5 Competition: Forecasting competition for artificial neural networks and computational intelligence. URL: http://www.neural-forecasting-competition.com/NN5/index.htm.

References

S.F. Crone, 2008, Results of the NN5 time series forecasting competition. Hong Kong, Presentation at the IEEE world congress on computational intelligence. WCCI'2008.

See Also

NN5.A

Examples

```
data(NN5.A.cont)
str(NN5.A.cont)
plot(ts(NN5.A.cont["NN5.111"]))
```

plotarimapred *Plot ARIMA predictions against actual values*

Description

The function plots ARIMA predictions against its actual values with prediction intervals.

Usage

```
plotarimapred(ts.cont, fit.arima, xlim, range.percent = 0.2, xreg = NULL,
ylab = NULL, xlab = NULL, main = NULL)
```

Arguments

ts.cont	A vector or univariate time series containing actual values for a time series that are to be plotted against its respective predictions. The number of consecutive values to be predicted is assumed to be equal to the number of rows in `ts.cont`. If `xreg` is used, the number of values to be predicted is set to the number of rows of `xreg`.
fit.arima	A fitted ARIMA model for the time series that is to be predicted. An object of class `"Arima"`, `"ar"` or `"fracdiff"`. See the `object` argument of the `forecast.Arima` function in the forecast package.
xlim	Numeric vector containing the initial and final limits of the x-axis to be plotted, respectively.
range.percent	A percentage which defines how much the range of the graphic's y-axis will be increased from the minimum limits imposed by data.
xreg	A vector, matrix, data frame or times series with new values of external regressors to be used for prediction (for class Arima objects only). See the `xreg` argument of the `forecast.Arima` function in the forecast package.
ylab	A title for the graphic's y-axis. Ignored if `NULL`.
xlab	A title for the graphic's x-axis. Ignored if `NULL`.
main	An overall title for the graphic. Ignored if `NULL`.

Details

The model in `fit.arima` is used for prediction by the `forecast.Arima` function in the `forecast` package. The resulting `forecast` object is then used for plotting the predictions and their intervals by the `plot.forecast` function also in the `forecast` package. For more details, see the `forecast.Arima` and the `plot.forecast` functions in the `forecast` package.

Value

None.

Author(s)

Rebecca Pontes Salles

References

See the forecast.Arima and the plot.forecast functions in the forecast package.

See Also

forecast.Arima, plot.forecast, arimapred

Examples

```
data(SantaFe.A,SantaFe.A.cont)
fit <- forecast::auto.arima(SantaFe.A)
ts.cont <- ts(SantaFe.A.cont,start=1001)
plotarimapred(ts.cont, fit, xlim=c(1001,1100))
```

SantaFe.A *Time series A of the Santa Fe Time Series Competition*

Description

A univariate time series derived from laser-generated data recorded from a Far-Infrared-Laser in a chaotic state.

Usage

```
data("SantaFe.A")
```

Format

A data frame with 1000 observations on the following variable.

V1 a numeric vector containing the observations of the univariate time series A of the Santa Fe Time Series Competition.

Details

The main benchmark of the Santa Fe Time Series Competition, time series A, is composed of a clean low-dimensional nonlinear and stationary time series with 1,000 observations. Competitors were asked to correctly predict the next 100 observations (SantaFe.A.cont). The performance evaluation done by the Santa Fe Competition was based on the NMSE errors of prediction found by the competitors.

Source

The Santa Fe Time Series Competition Data, URL: http://www.comp-engine.org/timeseries/time-series_data_source/source-151/.

References

A.S. Weigend, 1993, Time Series Prediction: Forecasting The Future And Understanding The Past. Reading, MA, Westview Press.

See Also

SantaFe.A.cont, SantaFe.D, SantaFe.D.cont

Examples

```
data(SantaFe.A)
str(SantaFe.A)
plot(ts(SantaFe.A))
```

SantaFe.A.cont *Continuation dataset of the time series A of the Santa Fe Time Series Competition*

Description

A univariate time series providing 100 points beyond the end of the time series A in SantaFe.A.

Usage

```
data("SantaFe.A.cont")
```

Format

A data frame with 100 observations on the following variable.

`V1` a numeric vector containing further observations of the univariate time series A of the Santa Fe Time Series Competition in `SantaFe.A`.

Details

Contains the 100 observations which were to be predicted of the time series A (`SantaFe.A`) as demanded by the Santa Fe Time Series Competition.

Source

The Santa Fe Time Series Competition Data, URL: http://www.comp-engine.org/timeseries/time-series_data_source/source-151/.

References

A.S. Weigend, 1993, Time Series Prediction: Forecasting The Future And Understanding The Past. Reading, MA, Westview Press.

See Also

`SantaFe.A`, `SantaFe.D`, `SantaFe.D.cont`

Examples

```
data(SantaFe.A.cont)
str(SantaFe.A.cont)
plot(ts(SantaFe.A.cont))
```

SantaFe.D *Time series D of the Santa Fe Time Series Competition*

Description

A univariate computer-generated time series.

Usage

```
data("SantaFe.D")
```

Format

A data frame with 100000 observations on the following variable.

`V1` a numeric vector containing the observations of the univariate time series D of the Santa Fe Time Series Competition.

Details

One of the benchmarks of the Santa Fe Time Series Competition, time series D, is composed of a four-dimensional nonlinear time series with non-stationary properties and 100,000 observations. Competitors were asked to correctly predict the next 500 observations of this time series (SantaFe.D.cont). The performance evaluation done by the Santa Fe Competition was based on the NMSE errors of prediction found by the competitors.

Source

The Santa Fe Time Series Competition Data, URL: http://www.comp-engine.org/timeseries/time-series_data_source/source-151/.

References

A.S. Weigend, 1993, Time Series Prediction: Forecasting The Future And Understanding The Past. Reading, MA, Westview Press.

See Also

SantaFe.D.cont, SantaFe.A, SantaFe.A.cont

Examples

```
data(SantaFe.D)
str(SantaFe.D)
plot(ts(SantaFe.D),xlim=c(1,2000))
```

SantaFe.D.cont *Continuation dataset of the time series D of the Santa Fe Time Series Competition*

Description

A univariate time series providing 500 points beyond the end of the time series D in SantaFe.D.

Usage

```
data("SantaFe.D.cont")
```

Format

A data frame with 500 observations on the following variable.

V1 a numeric vector containing further observations of the univariate time series D of the Santa Fe Time Series Competition in SantaFe.D.

Details

Contains the 500 observations which were to be predicted of the time series D (SantaFe.D) as demanded by the Santa Fe Time Series Competition.

Source

The Santa Fe Time Series Competition Data, URL: http://www.comp-engine.org/timeseries/time-series_data_source/source-151/.

References

A.S. Weigend, 1993, Time Series Prediction: Forecasting The Future And Understanding The Past. Reading, MA, Westview Press.

See Also

SantaFe.D, SantaFe.A, SantaFe.A.cont

Examples

```
data(SantaFe.D.cont)
str(SantaFe.D.cont)
plot(ts(SantaFe.D.cont))
```

---

slidingWindows *Generating sliding windows of data*

Description

The function extracts all possible subsequences (of the same length) of a time series (or numeric vector), generating a set of sliding windows of data, often used to train machine learning methods.

Usage

```
slidingWindows(timeseries,swSize)
```

Arguments

timeseries	A vector or univariate time series from which the sliding windows are to be extracted.
swSize	Numeric value of the required size (length) of each sliding window.

Details

The function returns all (overlapping) subsequences of size swSize of timeseries.

Value

A numeric matrix of size (length(`timeseries`)-`swSize`+1) by `swSize`, where each line is a sliding window.

Author(s)

Rebecca Pontes Salles

References

Lampert, C. H., Blaschko, M. B., and Hofmann, T. (2008). Beyond sliding windows: Object localization by efficient subwindow search. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1-8. IEEE.

Keogh, E. and Lin, J. (2005). Clustering of time series subsequences is meaningless: Implications for previous and future research. Knowledge and Information Systems, 8(2):154-177.

Examples

```
data("CATS")
SW <- slidingWindows(CATS[,1],4)
```

---

sMAPE *sMAPE error of prediction*

Description

The function calculates the sMAPE error between actual and predicted values.

Usage

```
sMAPE(actual, prediction)
```

Arguments

`actual`	A vector or univariate time series containing actual values for a time series that are to be compared against its respective predictions.
`prediction`	A vector or univariate time series containing time series predictions that are to be compared against the values in `actual`.

Value

A numeric value of the sMAPE error of prediction.

Author(s)

Rebecca Pontes Salles

References

Z. Chen and Y. Yang, 2004, Assessing forecast accuracy measures, Preprint Series, n. 2004-2010, p. 2004-10.

See Also

MAPE, MSE, NMSE, MAXError

Examples

```
data(SantaFe.A,SantaFe.A.cont)
pred <- marimapred(SantaFe.A,n.ahead=100)
sMAPE(SantaFe.A.cont[,1], pred)
```

Index