Package ‘artfima’

July 14, 2016

Type Package

Title ARTFIMA Model Estimation

Version 1.5

Date 2016-06-28

Author A. I. McLeod, Mark M. Meerschaert, Farzad Sabzikar

Maintainer A.I. McLeod `<aimcleod@uwo.ca>`

Description Fit and simulate ARTFIMA. Theoretical autocovariance function and spectral density function for stationary ARTFIMA.

Depends R (>= 2.1.0)

Imports ltsa, gsl

LazyLoad yes

LazyData yes

Classification/ACM G.3, G.4, I.5.1

Classification/MSC 62M10, 91B84

License GPL (>= 2)

URL `http://www.stats.uwo.ca/faculty/aim`

NeedsCompilation no

Repository CRAN

Date/Publication 2016-07-14 00:28:43

R topics documented:

`artfima-package` *ARTFIMA Model Estimation*

Description

Fit and simulate ARTFIMA. Theoretical autocovariance function and spectral density function for stationary ARTFIMA.

Details

Package:	artfima
Type:	Package
Title:	ARTFIMA Model Estimation
Version:	1.5
Date:	2016-06-28
Author:	A. I. McLeod, Mark M. Meerschaert, Farzad Sabzikar
Maintainer:	A.I. McLeod <aimcleod@uwo.ca>
Description:	Fit and simulate ARTFIMA. Theoretical autocovariance function and spectral density function for stati
Depends:	R (>= 2.1.0)
Imports:	ltsa, gsl
LazyLoad:	yes
LazyData:	yes
Classification/ACM:	G.3, G.4, I.5.1
Classification/MSC:	62M10, 91B84
License:	GPL (>= 2)
URL:	http://www.stats.uwo.ca/faculty/aim

Index of help topics:

```
Periodogram             Periodogram
SB32                    Turbulent flow data from Station SB32
```

`artfima`	MLE for ARTFIMA model
`artfima-package`	ARTFIMA Model Estimation
`artfimaSDF`	Computation of theoretical spectral density function (SDF)
`artfimaTACVF`	Autocovariance function of ARTFIMA
`artsim`	Simulation of stationary ARTFIMA
`bestModels`	Best BIC Models
`best_glp_models`	Best AIC/BIC Models for Specified GLP
`bev`	Beveridge Wheat Price Index, 1500 to 1869
`eaglecol`	Tree-ring indicies for Douglas Fir, Colorado, 1107-1964.
`ifisher`	Information matrix for ARTFIMA
`nilemin`	Nile Annual Minima, 622 AD to 1284 AD
`ogden`	Mean Annual St. Lawrence Riverflow
`plot.artfima`	Plot Method for "arfima" Object
`predict.artfima`	Predict method for artfima
`print.artfima`	Print Method for "arfima" Object
`print.bestmodels`	Print Method for "bestmodels" Object
`seriesa`	Series A from Box and Jenkins
`tseg`	Simulate Some Time Series Models of Interest

Author(s)

A. I. McLeod, Mark M. Meerschaert, Farzad Sabzikar Maintainer: A.I. McLeod <aimcleod@uwo.ca>

References

TBA

See Also

`ltsa`

Examples

```
artfima(rnorm(100))
```

`artfima` *MLE for ARTFIMA model*

Description

Maximum likelihood estimation of the ARTFIMA model as well as the edge cases ARIMA and ARFIMA. Exact MLE and Whittle approximate MLE are implemented.

Usage

```
artfima(z, glp = c("ARTFIMA", "ARFIMA", "ARIMA"), arimaOrder = c(0, 0, 0),
  likAlg = c("exact", "Whittle"),  fixd = NULL, b0 = NULL,
        lambdaMax = 3, dMax = 10)
```

Arguments

`z`	time series data
`glp`	general linear process type: ARTFIMA, ARFIMA or ARMA.
`arimaOrder`	c(p,D,q), where p is the AR order, D is the regular difference parameter and q is the MA order.
`likAlg`	"exact" or "Whittle" or "Whittle2"
`fixd`	only used with ARTFIMA, default setting fixd=NULL means the MLE for the parameter d is obtained other if fixed=d0, where d0 is a numeric value in the interval (-2, 2) the d parameter in ARTFIMA is fixed at this value while the remaining parameters are estimated.
`b0`	initial estimates - use only for high order AR models. See Details and Example.
`lambdaMax`	ARTFIMA boundard setting - upper limit for lambda
`dMax`	ARTFIMA boundard setting - absolute magnitude for d. See Note and Example

Details

The ARFIMA and ARIMA are subsets or edge-cases of the ARTFIMA model. The likelihood and probability density function for these models is defined by the multivariate normal distribution. The log-likelihood, AIC and BIC are comparable across models. When the Whittle MLE algorithm is used, the final log-likelihood is obtained by plugging this estimates into the exact log-likelihood.

The argument b0 is provided for fitting for fitting high order AR models with ARTFIMA. That is ARTFIMA(p,0,0) when p is large. This fitting is best done by fitting values with p=1,2,...,pmax. For p>1, set b0 equal to c(ans$b0, 0), where ans is the output from artfima for the p-1 order model. An example is given below. This technqiue is used by bestModels with q=0 and p>3.

Value

A lengthy list is produced. A terse summary is provided by the associated print method.

Note

Note: ARTFIMA parameters d and lambda on the boundary. The output from this function is normally viewed using the print method that has been implemented for class artfima. Check this output to see if any of the estimates are on the boundary. This may happen with the lambda or d parameter estimates in ARTFIMA. Another famous case is with the MA(1) models. Often when this happens the model is not statistically adequate because it is too parsimonious or otherwise misspecified. For example an AR(1) instead of an MA(1). See the R code for artfima if you wish to change the boundary limits set on the parameters - only for researchers not recommended otherwise.

Author(s)

A. I. McLeod, aimcleod@uwo.ca

References

McLeod, A.I., Yu, Hao and Krougly, Z. (2007). Algorithms for Linear Time Series Analysis: With R Package. Journal of Statistical Software 23/5 1-26.

See Also

`bestModels`

Examples

```
artfima(Nile) #Nile is a built in dataset in R
artfima(Nile, likAlg = "exact")
#
#fitting a high-order AR using recursion
## Not run:
#This may take 3 to 6 hours if exact MLE used!
#But Whittle MLE doesn't work properly for this example!!
 data(SB32)
 z <- SB32
 likAlg <- "exact"
 pmax <- 30
 startTime <- proc.time()[3]
 ic <- matrix(numeric(0), ncol=3, nrow=pmax+1)
 out <- artfima(z, arimaOrder=c(0,0,0), likAlg=likAlg)
 ic[1, 1] <- out$aic
 ic[1, 2] <- out$bic
 ic[1, 3] <- out$LL
 b1 <- c(out$b0, 0)
 for (i in 1:pmax) {
  out <- artfima(z, arimaOrder=c(i,0,0), b0=b1, likAlg=likAlg)
  b1 <- c(out$b0, 0)
  ic[i+1, 1] <- out$aic
  ic[i+1, 2] <- out$bic
  ic[i+1, 3] <- out$LL
 }
 endTime <- proc.time()[3]
 (totTime <- endTime-startTime)
 plot(0:pmax, ic[,1], xlab="AR order", ylab="AIC", pch=20, col="blue")
 indBest <- which.min(ic[,1])
 pBest <- indBest-1
 icBest <- ic[indBest,1]
 abline(h=icBest, col="brown")
 abline(v=pBest, col="brown")
 plot(0:pmax, ic[,2], xlab="AR order", ylab="BIC", pch=20, col="blue")
 indBest <- which.min(ic[,2])
 pBest <- indBest-1
 icBest <- ic[indBest,2]
 abline(h=icBest, col="brown")
```

```
 abline(v=pBest, col="brown")
 plot(0:pmax, ic[,3], xlab="AR order", ylab="log-lik", pch=20)

## End(Not run)#end dontrun
#
#setting new boundary limit
## Not run:
data(SB32)
#ARTFIMA(1,0,2) - MLE for d on boundar, dHat = 10
artfima(SB32, arimaOrder=c(1,0,2))
#note:
#log-likelihood = -10901.14, AIC = 21816.29, BIC = 21862.41
#Warning: estimates converged to boundary!
#mean     -0.5558988 8.443794e-02
#d         9.9992097 1.396002e-05
#lambda    2.9304658 8.050071e-02
#phi(1)    0.9271892 6.862294e-03
#theta(1)  0.8440911 1.709824e-02
#theta(2) -0.3650004 2.744227e-02
#
#now reset upper limit dMax and lambdaMax
#NOTE - there is only a very small improvement in the log-likelihood
artfima(SB32, arimaOrder=c(1,0,2), lambdaMax=20, dMax=40)
#ARTFIMA(1,0,2), MLE Algorithm: exact, optim: BFGS
#snr = 4.665, sigmaSq = 3.38228734331338
#log-likelihood = -10900.56, AIC = 21815.12, BIC = 21861.25
#               est.    se(est.)
#mean     -0.5558988  0.08443794
#d        27.0201256 36.94182328
#lambda    3.9412050  1.38296970
#phi(1)    0.9276901  0.00676589
#theta(1)  0.8342879  0.01715041
#theta(2) -0.3644787  0.02691869

## End(Not run)
```

`artfimaSDF` *Computation of theoretical spectral density function (SDF)*

Description

Computes the theoretical SDF at the Fourier frequencies for a time series of length n. Used for Whittle MLE. Assumes model parameters are valid for a stationary process.

Usage

```
artfimaSDF(n = 100, d = 0, lambda = 0, phi = numeric(0), theta = numeric(0),
     obj = NULL, plot=c("loglog", "log", "none"))
```

Arguments

`n`	length of time series
`d`	ARTFIMA difference parameter, any real value. When d=numeric(0), reduces to ARMA and lambda is ignored.
`lambda`	ARTFIMA tempered decay parameter. When lambda=numeric(0), reduces to ARFIMA
`phi`	AR coefficients
`theta`	MA coefficients, Box-Jenkins definition
`obj`	object of class artfima
`plot`	type of plot, "log-log", "log" or "none"

Details

The Fourier frequencies, 2*pi*c(1/n, floor(n/2)/n, 1/n), are used in the definition of the SDF. The SDF is normalized so that the area over (0, 0.5) equals the variance of the time series assuming unit innovation variance. The periodogram is normalized in the same way, so the mean of the periodogram is an estimate of the variance of the time series. See example below.

Value

vector of length floor(n/2) containing the values of the SDF at the Fourier frequencies, 2*pi*c(1/n, floor(n/2)/n, 1/n).

Warning

This function serves as a utility function for Whittle estimation so, for speed, we skip the checking if the parameters d, phi, or lambda are valid parameters for a stationary process.

Author(s)

A. I. McLeod, aimcleod@uwo.ca

References

TBA

See Also

`artfimaTACVF`, `Periodogram`

Examples

```
phi <- 0.8
n <- 256
set.seed(4337751)
z <- artsim(n, phi=phi)
VarZ <- mean((z-mean(z))^2)
Ip <- Periodogram(z)
```

```
length(Ip)
x <- (1/n)*(1:length(Ip))
plot(x, Ip, xlab="frequency", ylab="Spectral density & Periodogram",
     main=paste("AR(1), phi =", phi), type="l", col=rgb(0,0,1,0.5))
n <- 5000
y <- artfimaSDF(n, phi=phi)
x <- (1/n)*(1:length(y))
lines(x, y, type="l", lwd=1.25)
h <- x[2]-x[1] #step length
SimpsonsRule <- function(h, y) {
  n <- length(y)
  h/3*sum(y * c(1, rep(c(4,2), n-1), 1))
}
AreaApprox <- SimpsonsRule(h, y)
text(0.2, 50, labels=paste("Area under SDF using Simpson's Rule =",
                           round(AreaApprox,4)))
TVarZ <- 1/(1-phi^2)
text(0.2, 40, labels=paste("Theoretical AR Variance =", round(TVarZ,4)))
text(0.2, 30, labels=paste("mean(Ip) =", round(mean(Ip),4)))
text(0.2, 20, labels=paste("sample variance =", round(VarZ,4)))
```

`artfimaTACVF` *Autocovariance function of ARTFIMA*

Description

Theoretical autocovariance function of ARTFIMA model

Usage

```
artfimaTACVF(d = numeric(0), lambda = numeric(0), phi = numeric(0),
      theta = numeric(0), maxlag, sigma2 = 1, obj = NULL)
```

Arguments

`d`	ARTFIMA difference parameter, any real value. When d=0, reduces to ARMA and lambda is ignored.
`lambda`	ARTFIMA tempered decay parameter. When lambda=0, reduces to ARFIMA
`phi`	AR coefficients
`theta`	MA coefficients, Box-Jenkins definition
`maxlag`	maxlag+1 lags computed corresponding to 0,1,...,maxlag
`sigma2`	innovation variance
`obj`	output from artfima function

Value

vector of length maxlag+1 of the specified autocovariances

Author(s)

A. I. McLeod, aimcleod@uwo.ca

See Also

`ARMAacf`, `artfimaSDF`, `artsim`, `artfima`

Examples

```
#ARTFIMA - area under SDF equals theoretical Var(z[t])
#and sample variance = mean of periodogram
#
lambda <- 0.045
d <- 5/6
TVarZ <- artfimaTACVF(d=d, lambda=lambda, maxlag=3)[1]
TVarZ
n <- 256
set.seed(4337751)
z <- artsim(n, lambda=lambda, d=d)
VarZ <- mean((z-mean(z))^2)
Ip <- Periodogram(z)
mean(Ip)
length(Ip)
x <- (1/n)*(1:length(Ip))
plot(x, Ip, xlab="frequency", ylab="Spectral density & Periodogram",
     main=paste("lambda, d =", lambda, d), type="l", col=rgb(0,0,1,0.5))
n <- 5000
y <- artfimaSDF(n, lambda=lambda, d=d)
x <- (1/n)*(1:length(y))
lines(x, y, type="l", lwd=1.25)
h <- x[2]-x[1] #step length
SimpsonsRule <- function(h, y) {
  n <- length(y)
  h/3*sum(y * c(1, rep(c(4,2), n-1), 1))
}
AreaApprox <- SimpsonsRule(h, y)
text(0.2, 230, labels=paste("Area under SDF using Simpson's Rule =",
                            round(AreaApprox,4)))
text(0.2, 200, labels=paste("Theoretical ARTFIMA Variance =", round(TVarZ,4)))
text(0.2, 170, labels=paste("mean(Ip) =", round(mean(Ip),4)))
text(0.2, 140, labels=paste("sample variance =", round(VarZ,4)))
```

`artsim` *Simulation of stationary ARTFIMA*

Description

Simulation of stationary ARTFIMA, ARFIMA or ARIMA or bootstrap a fitted model. Useful for the parametric bootstrap.

Usage

```
artsim(n = 100, d = 0, lambda = 0, phi = numeric(0),
    theta = numeric(0), mean = 0, sigma2 = 1, obj = NULL)
```

Arguments

`n`	length of time series
`d`	artfima difference parameter, real value greater than zero. If d=0, ARIMA model is used.
`lambda`	lambda artfima temper decay parameter, if lambda=0, ARFIMA model is simulated
`phi`	AR coefficients
`theta`	MA coefficients
`mean`	mean of series
`sigma2`	innovation variance
`obj`	output from artfima(). If obj is not output from artfima() then the other arguments are used to determine the time series parameters, except for the series length n.

Value

vector of length n, the simulated time series

Author(s)

A. I. McLeod, aimcleod@uwo.ca

References

McLeod, A.I., Yu, Hao and Krougly, Z. (2007). Algorithms for Linear Time Series Analysis: With R Package. Journal of Statistical Software 23/5 1-26.

Examples

```
z <- artsim(5000, d=5/6, lambda=0.045)
var(z)
artfimaTACVF(d=5/6, lambda=0.045, maxlag=1)[1]
```

`bestModels` *Best BIC Models*

Description

ARIMA(p,0,q), ARFIMA(p,0,q) and ARTFIMA(p,0,q) models are fit for various p=0,1,..., and q=0,1,... and the best models according to the BIC criterion are selected.

Usage

```
bestModels(z, parMax = 4, nbest = 4, likAlg = c("exact", "Whittle"),
     d=0, ...)
```

Arguments

`z`	time series data
`parMax`	maximum number of parameters - see Details
`nbest`	number of models in selection
`likAlg`	likelihood method to use
`d`	regular differencing parameter indicating the number of times to difference
`...`	optional arguments for artfima such as lambdaMax

Details

$numPar = K$, where K is the number of structural models defined by $K = p + q + n(glp)$, where $n(glp) = 0, 1, 2$ according as the model is ARIMA, ARFIMA or ARTFIMA respectively.

These models are ranked according to the AIC/BIC criterion and the best ones are shown.

The plausibility is shown. This is defined for AIC by the eqn $p(AIC) = exp(0.5 * (min(AIC) - AIC))$, where AIC is the vector of AIC values. Similarly for the BIC.

Value

An S3 list object, "bestmodels". Output is provided using the print method for the "bestmodels"

Note

There are often small differences in the likelihood among a group of 5 or more of the best models. So the "exact" and "Whittle" likelihood methods may produce a different ranking of the models. For this reason the "exact" likelihood method may be preferred.

Author(s)

A.I. McLeod

See Also

best_glp_models print.bestmodels

Examples

```
## Not run:
data(ogden)
\dontrun{ #about 10 seconds
bestModels(ogden)
}

## End(Not run)
```

`best_glp_models` *Best AIC/BIC Models for Specified GLP*

Description

This function is used by bestModels

Usage

```
best_glp_models(z, glp = c("ARTFIMA", "ARFIMA", "ARIMA"), p = 2, q = 2,
   likAlg = c("exact", "Whittle"), d=0, ...)
```

Arguments

`z`	time series
`glp`	glp is equal to one of the following choices: "ARTFIMA", "ARFIMA" or "ARIMA"
`p`	maximum order of AR component
`q`	maximum order of MA component
`likAlg`	likAlg = c("exact", "Whittle")) either "exact" or "Whittle"
`d`	regular integer differencing parameter
`...`	optional arguments for artfima such as lambdaMax

Value

A list with 4 entries:

`LL`	log-likelihood of models
`artfima_time`	total time
`aic`	list with best aic models
`bic`	list with best bic models

Each of the components aic and bic is a list with three components:

`bestaic`	best aic models
`bestbicModel`	best model
`aic`	plausability

Similarly for the bic component.

Author(s)

A. I. McLeod

See Also

`bestModels`

Examples

```
## Not run:
#takes about 4 minutes. Checking result for bestmodels()
z<-tseg(1000, "BJARMA11")
ansARIMA <- best_glp_models(z, glp = "ARIMA", p=2, q=2)
ansARFIMA <- best_glp_models(z, glp = "ARFIMA", p=2, q=2)
ansARTFIMA <- best_glp_models(z, glp = "ARTFIMA", p=2, q=2)
ansARIMA$bic$bic
ansARFIMA$bic$bic
ansARTFIMA$bic$bic
bestModels(z)

## End(Not run)
```

`bev` *Beveridge Wheat Price Index, 1500 to 1869*

Description

Beveridge Wheat Price Index which gives annual price data from 1500 to 1869.

Usage

```
data("bev")
```

Format

The format is: Time-Series [1:370] from 1500 to 1869: 17 19 20 15 13 14 14 14 14 11 ...

Details

Baille suggests the time series is overdifferenced and is best fit by an ARFIMA model.

Source

CRAN package tseries.

References

R. T. Baillie (1996): Long Memory Processes and Fractional Integration in Econometrics. Journal of Econometrics, 73, 5-59.

Examples

```
data(bev)
#series needs a log transformation as is evident from the plot
plot(bev)
## Not run:
w <- diff(bev)
bestModels(w)

## End(Not run)
```

`eaglecol` *Tree-ring indicies for Douglas Fir, Colorado, 1107-1964.*

Description

Tree-ring indicies for Douglas Fir, Colorado, 1107-1964. There are 858 consecutive values. When the environment is suboptimal, tree ring growth is limited by the climate, usually either ambient temperature or precipitation. For this tree-ring time series, the tree is located on a mountain and the limiting growth factor is temperature.

Usage

```
data("eaglecol")
```

Format

The format is: Time-Series [1:858] from 1107 to 1964: 78 62 26 100 121 97 102 85 214 245 ...

Source

Laboratory of Tree-ring Research (LTRR), The University of Arizona http://ltrr.arizona.edu/

References

Fritts, H.C. et al. (1971) Multivariate techniques for specifying tree-growth and climatic relationships and for reconstructing anomalies in Paleoclimate. Journal of Applied Meteorology, 10, pp.845-864.

Hipel, K.W. and McLeod, A.I. (1994). Time Series Modelling of Water Resources and Environmental Systems. Elsevier. http://www.stats.uwo.ca/faculty/aim/1994Book/default.htm

McLeod, A.I. & Hipel, K.W. (1978), Preservation of the rescaled adjusted range, Water Resources Research 14, 491-516.

Examples

```
data(eaglecol)
plot(eaglecol)
## Not run: #confidence ellipse
library("ellipse") #needs this package!
ansTFD <- artfima(eaglecol)
v <- ansTFD$varbeta
bHat <- c(ansTFD$dHat, ansTFD$lambdaHat)
xy <- ellipse(v, centre=bHat, level=0.9)
plot(xy, type="l", lwd=2, xlab=expression(delta), ylab=expression(lambda))
points(matrix(bHat,ncol=2), pch=16, cex=3, col="blue")
#setwd("D:/DropBox/R/2016/artfima/Explore_ts_data/eaglecol")
#postscript(file="eaglecolCI.eps")
#plot(xy, type="l", lwd=2, xlab=expression(delta), ylab=expression(lambda))
#points(matrix(bHat,ncol=2), pch=16, cex=3, col="blue")
#graphics.off()

## End(Not run)
## Not run: #forecast comparison


## End(Not run)
```

`ifisher` *Information matrix for ARTFIMA*

Description

The information matrix for the lambda and d in ARTFIMA model. At present only the TFD and FD models are supported but it is planned to extend this to the full ARTFIMA model.

Usage

```
ifisher(d = numeric(0), lambda = numeric(0), phi = numeric(0),
  theta = numeric(0), sigma2 = 1, n = 1, obj = NULL,
  alg = c("Fisher", "Whittle", "approx"))
```

Arguments

`d`	d parameter
`lambda`	lambda parameter
`phi`	AR coefficients
`theta`	MA coefficients, Box-Jenkins definition
`sigma2`	innovation variance
`n`	series length
`obj`	object of class artfima
`alg`	"Fisher", "Whittle" or "approx"

Details

This is the expected information matrix. The artfima() function returns the component varbeta that is the inverse of the observed information for a fitted model computed from the Hessian matrix.

Value

`se`	standard errors
`f`	information matrix

Author(s)

A. I. McLeod

References

TBA

See Also

`artfima`

Examples

```
ifisher(d=0.2, lambda=0.0025)
ifisher(d=0.2, lambda=0.0025, alg="Whittle")
ifisher(d=0.2, lambda=0.0025, alg="approx")
```

`nilemin` *Nile Annual Minima, 622 AD to 1284 AD*

Description

Annual Minimum flow of Nile River. See below for details.

Usage

```
data("nilemin")
```

Format

The format is: Time-Series [1:663] from 622 to 1284: 11.57 10.88 11.69 11.69 9.84 ... - attr(*, "title")= chr "#Nile River minima series"

Details

The minimum annual level of the Nile has been recorded over many centuries and was given by Toussoun (1925). The data over the period 622 AD to 1284 AD is considered more homogenous and reliable than the full dataset and has been analyzed by Beran (1994) and Percival and Walden (2000). The complete dataset is available StatLib Datasets - see: hipel-mcleod archive, file: Minimum.

Source

Toussoun, O. (1925). Memoire sur l'Histoire du Nil. In Memoires a l'Institut d'Egypte, 18, 366-404.

References

Beran, J. (1994). Statistics for Long-Memory Processes. Chapman and Hall, New York.

Percival, D.B. and Walden, A.T. (2000) Wavelet Methods for Time Series Analysis. Cambridge University Press.

Examples

```
data(nilemin)
artfima(nilemin, likAlg="Whittle")
## Not run:
#compare exact and Whittle using bestModel()
start <- proc.time()[3]
ans<-bestModel(nilemin)
tot <- proc.time()[3]-start
start <- proc.time()[3]
ansW <- bestModel(nilemin, likAlg="Whittle")
totW <- proc.time()[3]-start
t <- c(tot, totW)
names(t) <- c("exact", "Whittle")
#compare times - about 100 seconds vs 3 seconds
```

```
t
#compare best models
ans
ansW
#AIC/BIC scores similar but rankings to change.
#ARTFIMA(0,0,0) is ranked best by both AIC and BIC
#ARIMA(2,0,1) is ranked second best by both AIC and BIC
#ARFIMA(0,0,0) is ranked 3rd by BIC and is not among top 5 by AIC

## End(Not run)
```

`ogden` *Mean Annual St. Lawrence Riverflow*

Description

Mean Annual unregulated riverflows of the St. Lawrence River at Ogdensburg, N.Y. from 1860 to 1957 is comprised of 97 consecutive observations.

Usage

```
data("ogden")
```

Format

The format is: Time-Series [1:97] from 1860 to 1956: 7788 8040 7733 7528 7528 ...

Details

Hipel and McLeod (1994, 2005) showed this time series could be adequately modelled using an AR(1).

Source

Hipel, K.W. and McLeod, A.I., (1994, 2005). Time Series Modelling of Water Resources and Environmental Systems. Electronic reprint of our book orginally published in 1994. http://www.stats.uwo.ca/faculty/aim/1994Book/.

Examples

```
data(ogden)
#compare fits of AR(1) and TFD
arima(ogden, order=c(1,0,0))
artfima(ogden) #this model has one more parameter

#Find AIC/BIC 3 best models. Takes about 10 sec
## Not run:
system.time(ans <- bestModels(ogden, nbest=3))
summary(ans) #summary provides plausibility as well as scores

## End(Not run)
```

`Periodogram` *Periodogram*

Description

Computes the raw periodogram defined by,

$$I(f_j) = \frac{1}{n}|sumz[t]exp(2\pi f_j)|^2$$

Usage

```
Periodogram(z)
```

Arguments

`z`	vector, time series

Details

The expected value of the periodogram equals the spectral density function.

Value

the periodogram

Author(s)

A. I. McLeod

See Also

`artfimaSDF`

Examples

```
data(sunspot.year)
Ip <- Periodogram(sunspot.year)
fr <- (1:length(Ip))/length(sunspot.year)
plot(fr, Ip, xlab="frequency", ylab="Periodogram")
```

`plot.artfima` *Plot Method for "arfima" Object*

Description

Plots the observed periodogram and the fitted spectral density function.

Usage

```
## S3 method for class 'artfima'
plot(x, which = c("all", "logsdf", "loglogsdf", "res"),
                 mainQ = TRUE, subQ = TRUE, lag.max = 30, ...)
```

Arguments

`x`	object of class "artfima"
`which`	"all", "logsd", "loglogsdf" or "res" plot
`mainQ`	include plot title
`subQ`	include subtitle
`lag.max`	maximum lag in residual autocorrelation plot and test
`...`	optional arguments

Value

None. Plot produced is a side-effect.

Author(s)

A. I. McLeod, aimcleod@uwo.ca

See Also

`artfima`

Examples

```
z <- artsim(n=500, d=5/6, lambda=0.045)
ans <- artfima(z)
plot(ans)
plot(ans, which="loglogsdf", subQ=FALSE, mainQ=FALSE)
title(main="Simulated Series", sub="delta=5/6")
```

`predict.artfima` *Predict method for artfima*

Description

The optimal minimum mean square error forecast and its standard deviation for lags 1, 2, ..., n.ahead is computed at forecast origin starting at the end of the observed series used in fitting. The exact algorithm discussed in McLeod, Yu and Krougly is used.

Usage

```
## S3 method for class 'artfima'
predict(object, n.ahead=10, ...)
```

Arguments

`object`	object of class "artfima"
`n.ahead`	number of steps ahead to forecast
`...`	optional arguments

Value

a list with two components

`Forecasts`	Description of 'comp1'
`SDForecasts`	Description of 'comp2'

Author(s)

A. I. McLeod, aimcleod@uwo.ca

References

McLeod, A.I., Yu, Hao and Krougly, Z. (2007). Algorithms for Linear Time Series Analysis: With R Package. Journal of Statistical Software 23/5 1-26.

See Also

`predict.Arima`

Examples

```
ans <- artfima(seriesa, likAlg="Whittle")
predict(ans)
#compare forecasts from ARTFIMA etc.
  ## Not run:
ML <- 10
ans <- artfima(seriesa)
```

```
Ftfd <- predict(ans, n.ahead=10)$Forecasts
ans <- artfima(seriesa, glp="ARIMA", arimaOrder=c(1,0,1))
Farma11 <- predict(ans, n.ahead=10)$Forecasts
ans <- artfima(seriesa, glp="ARFIMA")
Ffd <- predict(ans, n.ahead=10)$Forecasts
#arima(0,1,1)
ans <- arima(seriesa, order=c(0,1,1))
fEWMA <- predict(ans, n.ahead=10)$pred
yobs<-seriesa[188:197]
xobs<-188:197
y <- matrix(c(yobs,Ffd,Ftfd,Farma11,fEWMA), ncol=5)
colnames(y)<-c("obs", "FD", "TFD", "ARMA11","FEWMA")
x <- 197+1:ML
x <- matrix(c(xobs, rep(x, 4)), ncol=5)
plot(x, y, type="n", col=c("black", "red", "blue", "magenta"),
     xlab="t", ylab=expression(z[t]))
x <- 197+1:ML
points(xobs, yobs, type="o", col="black")
points(x, Ffd, type="o", col="red")
points(x, Ftfd, type="o", col="blue")
points(x, Farma11, type="o", col="brown")
points(x, fEWMA, type="o", col="magenta")
legend(200, 18.1, legend=c("observed", "EWMA", "FD", "TFD", "ARMA"),
       col=c("black", "magenta", "red", "blue", "brown"),
       lty=c(rep(1,5)))

## End(Not run)
```

`print.artfima` *Print Method for "arfima" Object*

Description

Displays the fitted model. The exact log-likelihood, AIC and BIC are shown. The signal-to-noise ratio (snr) is defined the (sample variance minus the estimated innovation variance) divided by the innovation variance. Similar to the coefficient of determination in regression, it indicates how much of the randomness is captured by the model.

Usage

```
## S3 method for class 'artfima'
print(x, ...)
```

Arguments

`x`	object of class "artfima"
`...`	optional arguments

Value

A terse summary is displayed

Author(s)

A. I. McLeod, aimcleod@uwo.ca

References

TBA

See Also

`artfima`

Examples

```
artfima(rnorm(100))
```

`print.bestmodels` *Print Method for "bestmodels" Object*

Description

Methods function for bestModels.

Usage

```
## S3 method for class 'bestmodels'
print(x, ...)
```

Arguments

`x`	produced by bestModels
`...`	additional arguments

Details

The plausibility is shown. This is defined for AIC by the eqn $p(AIC) = exp(0.5 * (min(AIC) - AIC))$, where AIC is the vector of AIC values. Similarly for the BIC.

Value

Data frame with 6 rows and 5 columns. The first column corresonds to best models, second the second best, etc. The rows corresond respectively to the chosen AIC models, AIC values, AIC plausibility, BIC models, BIC values and BIC plausibility

Author(s)

A. I. McLeod

See Also

`bestModels`

Examples

```
## Not run:  #takes about 10 seconds
data(ogden)
ans<-bestModels(ogden)
ans

## End(Not run)
```

SB32 *Turbulent flow data from Station SB32*

Description

Turbulent flow water time series, Lake Huron, during 2009-2010. Sampled every second.

Usage

```
data("SB32")
```

Format

The format is: num [1:5374] -2.2 1.4 -0.6 -0.4 -1.5 -2.6 -0.9 0.5 -0.9 1.5 ...

Details

See paper by Meerschaert, Sabzikar, Phanikumar and Zeleke (2014).

References

M.M. Meerschaert, Farzad Sabzikar, M.S. Phanikumar, and A. Zeleke, Tempered fractional time series model for turbulence in geophysical flows, Journal of Statistical Mechanics: Theory and Experiment, Vol. 2014 p. P09023 (13 pp.) doi:10.1088/1742-5468/2014/09/P09023.

Examples

```
data(SB32)
str(SB32)

#Figure from our paper
## Not run:
ans0 <- artfima(SB32, fixd=5/6)
ans1 <- artfima(SB32, arimaOrder=c(1,0,2)) #best
p <- ans1$arimaOrder[1]
q <- ans1$arimaOrder[3]
sigmaSq1 <- ans1$sigmaSq
sigmaSq0 <- ans0$sigmaSq
w <- SB32
n <- length(w)
Ip <- Periodogram(w)
fr <- (1/n)*(1:length(Ip))
plot(log(fr), log(Ip), xlab="log frequency", ylab="log power",
     type="p", col=rgb(0,0,1,0.4), pch=16)
y <- sigmaSq1*artfimaSDF(n=length(SB32), obj=ans1, plot="none")
lines(log(fr), log(y), type="l", lwd=2.5, col="red")
y0 <- sigmaSq0*artfimaSDF(n=length(SB32), obj=ans0, plot="none")
lines(log(fr), log(y0), type="l", lwd=3.5, col="green", lty=2)
TFD_label <- expression(paste("TFD, ", delta == 5/6, ", ",
                              hat(lambda) == 0.045))
legend(x=-8, y=-5, xjust=0, yjust=0, legend=c("ARTFIMA(1,0,2)", TFD_label),
       lty=c(1,2), lwd=c(2.5,3.5), col=c("red", "green"), bty="n")

## End(Not run)
```

`seriesa` *Series A from Box and Jenkins*

Description

Chemical process concentration readings every two hours is comprised of 197 consecutive observations. Box and Jenkins fit ARMA(1,1) and ARIMA(0,1,1) to this data.

Usage

```
data("seriesa")
```

Format

The format is: Time-Series [1:197] from 1 to 197: 17 16.6 16.3 16.1 17.1 16.9 16.8 17.4 17.1 17 ...

Source

listed in Box and Jenkins book

References

Box and Jenkins (1970). Time Series Analysis: Forecasting and Control.

Examples

```
data(seriesa)
#compare ARMA(1,1) models and timings
system.time(arima(seriesa, order=c(1,0,1)))
system.time(artfima(seriesa, arimaOrder=c(1,0,1)))
#Remark: there is a slight difference due to the fact that arima()
#uses the exact MLE for the mean parameter whereas artfima() uses
#the sample average. In practice, the difference is almost negible.
#
#Find AIC/BIC 3 best models. Takes about 15 sec
## Not run:
system.time(ans <- bestModels(seriesa, nbest=3))
summary(ans) #summary provides plausibility as well as scores

## End(Not run)
```

`tseg` *Simulate Some Time Series Models of Interest*

Description

Time series models are simulated based on some familar characteristics described in Details.

Usage

```
tseg(n, which = c("BJAR2", "BJAR1", "BJAR3", "PWAR4", "BJARMA11", "MHAR9",
"NileMin", "SB32"))
```

Arguments

`n`	length of series
`which`	which model

Details

BJAR1 is the AR(1) model fit to the sunspot series in BJR BJAR2 is the AR(2) model fit to the sunspot series in BJR BJAR3 is the AR(3) model fit to the sunspot series in BJR BJAR2 is the AR(2) model fit to the sunspot series in BJR PWAR4 is the AR(4) model, PW, BJARMA11 is the ARMA(1,1) model fit to Series A in BJR MHAR9 is the AR(9) model fit to the sunspot series in MHL NileMin is ARFIMA(0,0,0), d=0.39 SB32 is ARTFIMA(0,0,0), d=5/8, lambda=0.045

Value

vector of time series values

Author(s)

A. I. McLeod

References

BJR) Box, Jenkins and Reinsel (2005), Table 7.11 PW) Percival and Walden, 1990, p.45 MHL) McLeod, Hipel and Lennox, 1978, p.581

See Also

`artsim`

Examples

```
z <- tseg(5000, "MHAR9")
arima(z, order=c(9,0,0), fixed=c(NA,NA,0,0,0,0,0,NA,NA), transform.pars=FALSE)
```

Index