Package 'FGN'

July 2, 2014

Version 2.0-12

Date 2014-05-15

Title Fractional Gaussian Noise and power law decay time series model fitting

Author A.I. McLeod and Justin Veenstra

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

Depends R (>= 2.1.0)

Imports akima, ltsa

Description Exact MLE and Whittle MLE estimation for power law decay models.

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

Classification/MSC 62M10, 91B84

LazyLoad yes

LazyData yes

License CC BY-NC-SA 3.0

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

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-05-16 07:33:31

R topics documented:

`FGN-package` *Fractional Gaussian Noise and hyperbolic decay time series model fitting*

Description

Exact and Whittle MLE for time series models with hyperbolic decay. Simulation and regression supported for FGN.

Details

Package:	FGN
Type:	Package
Version:	2.0-12
Date:	2014-05-15
License:	CC BY-NC-SA 3.0
LazyLoad:	yes
LazyData:	yes

Author(s)

A. I. McLeod and Justin Veenstra

Maintainer: aimcleod@uwo.ca

References

Hipel, K.W. and McLeod, A.I., (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/.

McLeod, A.I., Yu, Hao, Krougly, Zinovi L. (2007). Algorithms for Linear Time Series Analysis, Journal of Statistical Software.

McLeod, A.I. and Veenstra, Justin (2012). Hyperbolic Decay Time Series Models (in press).

See Also

HurstK, FitFGN, FitRegressionFGN, SimulateFGN, print.FitFGN, summary.FitFGN, predict.FitFGN, plot.FitFGN, residuals.FitFGN, GetFitFGN, GetFitFD, GetFitPLS, GetFitPLA

Examples

```
#Example 1
#Compare HurstK and MLE for H
#Hurst K for Nile Minima
```

```
data(NileMin)
HurstK(NileMin)
out<-FitFGN(NileMin)
summary(out)
plot(out)
coef(out)
#
#Example 2.
#Compare models
## Not run:
   T1 <- proc.time()[3]
   ansFD <- GetFitFD(NileMin)
   T2 <- proc.time()[3]
   ansFGN <- GetFitFGN(NileMin)
   T3 <- proc.time()[3]
   ansPLS <- GetFitPLS(NileMin)
   T4 <- proc.time()[3]
   ansPLA <- GetFitPLA(NileMin)
   T5 <- proc.time()[3]
   tbLLE <- c(ansFD[[2]],ansFGN[[2]],ansPLS[[2]],ansPLA[[2]])
   est <-  c(ansFD[[3]],ansFGN[[3]],ansPLS[[3]],ansPLA[[3]])
   tbLL <- round(tbLLE, 2)
   est <- round(est, 3)
   T<-c(T2-T1,T3-T2,T4-T3,T5-T4)
   m<-matrix(c(est,tbLL, T),nrow=4, ncol=3)
   dimnames(m)<-list(list("FD","FGN","PLS","PLA"), list("alpha","logL", "time"))
   mE <- m
   mE
   #
   T1 <- proc.time()[3]
   ansFD <- GetFitFD(NileMin, algorithm="wmle")
   T2 <- proc.time()[3]
   ansFGN <- GetFitFGN(NileMin, algorithm="wmle")
   T3 <- proc.time()[3]
   ansPLA <- GetFitPLS(NileMin, algorithm="wmle")
   T4 <- proc.time()[3]
   ansPLS <- GetFitPLA(NileMin, algorithm="wmle")
   T5 <- proc.time()[3]
   #tbLL <- c(ansFD[[2]],ansFGN[[2]],ansPLS[[2]],ansPLA[[2]])
   z <- NileMin-mean(NileMin)
  tbLLW <- c(LLFD(ansFD[[1]],z), LLFGN(ansFGN[[1]],z), LLPLS(ansPLS[[1]],z), LLPLA(ansPLA[[1]],z))
   est <-  c(ansFD[[3]],ansFGN[[3]],ansPLS[[3]],ansPLA[[3]])
   tbLL <- round(tbLLW, 2)
   est <- round(est, 3)
   T<-c(T2-T1,T3-T2,T4-T3,T5-T4)
   m<-matrix(c(est,tbLL, T),nrow=4, ncol=3)
   dimnames(m)<-list(list("FD","FGN","PLS","PLA"), list("alpha","logL", "time"))
   mW<-m
   mW
   m<-cbind(mE,mW)
   m

## End(Not run)
```

`acvfFD` *autocovariance function of fractionally-differenced white noise*

Description

The autocovariance function of fractionally differenced white noise is computed for lags 0, 1, ..., maxlag.

Usage

```
acvfFD(d, maxlag)
```

Arguments

`d`	fractional difference parameter
`maxlag`	maximum lag

Value

vector of length maxlag+1 containing the autocovariances

Note

White noise corresponds to d=0.

Author(s)

A. I. McLeod

See Also

`LLFD`

Examples

```
acvfFD(0.2, 10)
```

`acvfFGN` *Autocovariance of FGN*

Description

The FGN time series is an example of a time series exhibiting long-range dependence and characterized by the fact that its autocorrelation function exhibits hyperbolic decay rather than exponential decay found in stationary ARMA time series. The FGN and other alternatives are discussed in Hipel and McLeod (2005).

Usage

```
acvfFGN(H, maxlag)
```

Arguments

`H`	Hurst parameter
`maxlag`	acvf computed at lags 0,1,...,maxlag

Value

value of the autocorrelation at lag(s) k

Note

The parameter H should be in (0,1). An error message is given if it is not.

Author(s)

A.I. McLeod

References

Hipel, K.W. and McLeod, A.I., (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/.

See Also

`LLFGN`, `acf`

Examples

```
#compute the acf at lags 0,1,...,10 when H=0.7
acvfFGN(0.7, 10)
```

`acvfPLA` *autocovariance function for the PLA model*

Description

The autocovariance function is computed for the power-law autocorrelation time series model.

Usage

```
acvfPLA(alpha, maxlag)
```

Arguments

`alpha`	decay parameter
`maxlag`	maximum lag

Value

vector of maxlag+1 containing the autocovariances at lags 0, 1, ..., maxlag.

Note

alpha=1 corresponds to white noise

Author(s)

A. I. McLeod and Justin Veenstra

See Also

`LLPLA`

Examples

```
acvfPLA(0.4, 10)
```

acvfPLS *autocovariance function for PLS model*

Description

The autocovariance function is computed for the PLS time series model.

Usage

```
acvfPLS(alpha, maxlag)
```

Arguments

alpha	decay parameter
maxlag	maximum lag

Value

vector of maxlag+1 containing the autocovariances at lags 0, 1, ..., maxlag.

Note

alpha=1 corresponds to white noise

Author(s)

A. I. McLeod and Justin Veenstra

See Also

LLPLA

Examples

```
acvfPLS(0.4, 10)
```

Boot *Generic Bootstrap Function*

Description

Generic function to bootstrap a fitted model.

Usage

```
Boot(obj, R=1, ...)
```

Arguments

obj	fitted object
R	number of bootstrap replicates
...	optional arguments

Value

Parametric bootstrap simulation

Author(s)

A.I. McLeod

References

McLeod, A.I., Yu, Hao, Krougly, Zinovi L. (2007). Algorithms for Linear Time Series Analysis, Journal of Statistical Software.

See Also

Boot.FitFGN

Examples

```
## Not run:
data(NileMin)
out<-FitFGN(NileMin)
Boot(out, R=3)

## End(Not run)
```

`Boot.FitFGN` *Simulate Fitted FGN Model*

Description

Simulate a realization from a fitted AR model. This is useful in the parametric bootstrap. Generic function for "Boot" method.

Usage

```
## S3 method for class 'FitFGN'
Boot(obj, R = 1, ...)
```

Arguments

`obj`	the output from FitAR
`R`	number of bootstrap replications
`...`	optional arguments

Details

The method of Davies and Harte (1987) is used if it is applicable, otherwise the Durbin-Levinsion recursion is used.

Value

If R=1, a simulated time series with the same length as the original fitted time series is produced. Otherwise if R>1, a matrix with R columns and number of rows equal to the length of the series containing R replications of the bootstrap.

Author(s)

A.I. McLeod

References

McLeod, A.I., Yu, Hao, Krougly, Zinovi L. (2007). Algorithms for Linear Time Series Analysis, Journal of Statistical Software.

See Also

`SimulateFGN`, `DHSimulate` `DLSimulate`

Examples

```
#Example 1
#Fit a FGN model and determine the bootstrap sd of H
#Measure cpu time. With R=250, it takes about 23 sec
#on 3.6 GHz Pentium IV.
## Not run:
data(NileMin)
outNileMin<-FitFGN(NileMin)
start<-proc.time()[1]
R<-25
Hs<-numeric(R)
Z<-Boot(outNileMin, R=R)
for (i in 1:R)
    Hs[i]<-GetFitFGN(Z[,i])$H
BootSD<-sd(Hs) #this is the bootstrap sd
end<-proc.time()[1]
totTim<-end-start

## End(Not run)
```

---

coef.FitFGN *Display estimated parameters from FitFGN*

Description

Method function to display fitted parameters, their standard errors and Z-ratio for FGN models fit with FitFGN.

Usage

```
## S3 method for class 'FitFGN'
coef(object, ...)
```

Arguments

`object`	`obj` the output from `FitFGN`
`...`	optional parameters

Value

A matrix is returned. The columns of the matrix are labeled MLE, sd and Z-ratio. The rows labels indicate the AR coefficients which were estimated followed by mu, the estimate of mean.

Author(s)

A.I. McLeod

References

McLeod, A.I., Yu, Hao, Krougly, Zinovi L. (2007). Algorithms for Linear Time Series Analysis, Journal of Statistical Software.

Examples

```
data(NileMin)
out<-FitFGN(NileMin)
coef(out)
```

`earfima` *Exact MLE for ARFIMA*

Description

The time series is corrected for the sample mean and then exact MLE is used for the other parameters.

Usage

```
earfima(z, order = c(0, 0, 0), lmodel = c("FD", "FGN", "PLA", "NONE"))
```

Arguments

`z`	time series
`order`	(p,d,q) where p=order AR, d=regular difference, q=order MA
`lmodel`	type of long-memory component: FD, FGN, PLA or NONE

Details

The sample mean is asymptotically efficient.

Value

list with components:

`bHat`	transformed optimal parameters
`alphaHat`	estimate of alpha
`HHat`	estimate of H
`dHat`	estimate of d
`phiHat`	estimate of phi
`thetaHat`	estimate of theta
`wLL`	optimized value of Whittle approximate log-likelihood
`LL`	corresponding exact log-likelihood
`convergence`	convergence indicator
`algorithm`	optimization algorithm used, 1 for L-BFGS-B, 2 for Nelder-Mead, 3 for SANN

Author(s)

Justin Veenstra and A. I. McLeod

Examples

```
z <- rnorm(100)
earfima(z, lmodel="FGN")
```

FitFGN *MLE estimation for FGN*

Description

Exact MLE estimation for FGN

Usage

```
FitFGN(z, demean = TRUE, MeanMLEQ = FALSE, lag.max = "default")
```

Arguments

`z`	time series, vector or ts object.
`demean`	if True, subtract mean. Otherwise assume it is zero.
`MeanMLEQ`	if True, an iterative algorithm is used for exact simultaneous MLE estimation of the mean and other parameters.
`lag.max`	the residual autocorrelations are tabulated for lags 1, . . . , lag.max. Also lag.max is used for the Ljung-Box portmanteau test.

Details

The exact loglikelihood function is maximized numerically using `optimize`. The standard error for the H parameter is estimated (McLeod, Yu and Krougly, 2007).

Value

A list with class name "FitAR" and components:

`loglikelihood`	value of the loglikelihood
`H`	estimate of H parameter
`SEH`	SE of H estimate
`sigsqHat`	innovation variance estimate
`muHat`	estimate of the mean
`SEmu`	SE of mean
`Rsq`	R-squared, coefficient of forecastability
`LjungBox`	table of Ljung-Box portmanteau test statistics

`res`	normalized residuals, same length as z
`demean`	TRUE if mean estimated otherwise assumed zero
`IterationCount`	number of iterations in mean mle estimation
`MLEMeanQ`	TRUE if mle for mean algorithm used
`tsp`	tsp(z)
`call`	result from match.call() showing how the function was called
`DataTitle`	returns attr(z,"title")

Author(s)

A.I. McLeod

References

McLeod, A.I., Yu, Hao, Krougly, Zinovi L. (2007). Algorithms for Linear Time Series Analysis, Journal of Statistical Software.

See Also

`GetFitFGN`, `FitRegressionFGN`, `Boot.FitFGN`, `coef.FitFGN`, `plot.FitFGN`, `print.FitFGN`, `summary.FitFGN`, `HurstK`

Examples

```
data(NileMin)
out<-FitFGN(NileMin)
summary(out)
plot(out)
coef(out)
```

`FitRegressionFGN` *Regression with FGN Errors*

Description

Fits a multiple linear regression with FGN errors

Usage

```
FitRegressionFGN(X, y)
```

Arguments

`X`	design matrix, must include column of 1's if constant term is present
`y`	the response variable, a time series

Details

An iterative algorithm is used to compute the exact MLE.

Value

a list with 3 elements:

`Loglikelihood`	value of the maximized loglikelihood
`H`	MLE for H
`alpha`	MLE for regression coefficients corresponding to colums of X

Note

It is assumed that X is not collinear.

Author(s)

A.I. McLeod

References

McLeod, A.I., Yu, Hao, Krougly, Zinovi L. (2007). Algorithms for Linear Time Series Analysis, Journal of Statistical Software.

See Also

`FitFGN`, `lsfit`

Examples

```
#simulate FGN with mean zero and H=0.2 and fit exact mle for H and mean
H<-0.2
z<-SimulateFGN(512, H)
mean(z)
X<-matrix(rep(1,length(z)), ncol=1)
ans<-FitRegressionFGN(X,z)
ans


#fit a step intervention model to the Nile annual riverflow data
data(NileFlowCMS)
n<-length(NileFlowCMS)
X<-matrix(c(rep(1,n),rep(0,32),rep(1,n-32)),ncol=2)
ans<-FitRegressionFGN(X,NileFlowCMS)
ans
```

GetFitFD — *Fit FD Time Series Model*

Description

Exact maximum likelihood estimation of the parameter d in fractionally-differenced white noise (FD). In this model, alpha=1-2*d.

Usage

```
GetFitFD(z, MeanZeroQ = FALSE, algorithm=c("emle","wmle"), ciQ=FALSE)
```

Arguments

`z`	time series data vector
`MeanZeroQ`	optional argument, default is MeanZeroQ=FALSE. Set to TRUE if the mean is known to be zero
`algorithm`	"emle" or "wmle" for exact or Whittle mle
`ciQ`	TRUE or FALSE according as 95 percent confidence interval computed and plotted

Value

a list with four/five elements:

`d`	MLE for d
`Loglikelihood`	value of the maximized loglikelihood
`alpha`	MLE for alpha
`algorithm`	either "emle" or "wmle"
`ci`	95 percent confidence interval for d

Author(s)

A. I. McLeod

See Also

`GetFitFGN`

Examples

```
#Example 1
#fit Gaussian White Noise, d=0
z<-rnorm(500, 100, 10)
GetFitFD(z)

#Example 2
#estimate d for NileMin series
data(NileMin)
GetFitFD(NileMin)
GetFitFD(NileMin, algorithm="wmle")
```

GetFitFGN *Fit FGN Time Series Model*

Description

Exact maximum likelihood estimation of the parameter H in fractional Gaussian noise (FGN). This is a utility function used by FitFGN but it is also useful in simulation experiments since it is faster than using FitFGN. See example below. In this model, alpha=2-2*H.

Usage

```
GetFitFGN(z, MeanZeroQ = FALSE, algorithm=c("emle", "wmle"), ciQ=FALSE)
```

Arguments

z	time series data vector
MeanZeroQ	optional argument, default is MeanZeroQ=FALSE. Set to TRUE if the mean is known to be zero
algorithm	"emle" or "wmle" for exact or Whittle mle
ciQ	TRUE or FALSE according as 95 percent confidence interval computed and plotted

Details

The function `optimize` is used. It is very rare but it has been observed that `optimize` can incorrectly choose an endpoint. If this happens a warning is given and `optim` is used.

Value

a list with four/five elements:

H	MLE for H
Loglikelihood	value of the maximized loglikelihood
alpha	MLE for alpha
algorithm	either "emle" or "wmle"
ci	95 percent confidence interval for H

Author(s)

A.I. McLeod

References

McLeod, A.I., Yu, Hao, Krougly, Zinovi L. (2007). Algorithms for Linear Time Series Analysis, Journal of Statistical Software.

See Also

optimize, optim, Boot.FitFGN, FitFGN, FitRegressionFGN

Examples

```
#Example 1
#fit Gaussian White Noise, H=0.5
z<-rnorm(500, 100, 10)
GetFitFGN(z)

#Example 2
#estimate H for NileMin series
data(NileMin)
GetFitFGN(NileMin)

#Example 3
#Timing comparison for GetFitFGN and FitFGN
ns<-c(500,1000) #may extend this to other n's
H<-0.8
nR<-10
tim1<-tim2<-numeric(length(ns))
for (i in 1:length(ns)){
    n <- ns[i]
    t1<-t2<-0
    s1<-proc.time()[1]
    for (iR in 1:nR){
        z<-SimulateFGN(n, H)
        H1<-GetFitFGN(z)
        }
    e1<-proc.time()[1]
    t1<-t1+(e1-s1)
    s2<-proc.time()[1]
    for (iR in 1:nR){
        z<-SimulateFGN(n, H)
        H2<-FitFGN(z)
        }
    e2<-proc.time()[1]
    t2<-t2+(e2-s2)
    tim1[i]<-t1
    tim2[i]<-t2
}
tb<-matrix(c(tim1,tim2),ncol=2)
dimnames(tb)<-list(ns,c("GetFitFGN","FitFGN"))
```

`GetFitPLA` *Fit PLA Time Series Model*

Description

Exact maximum likelihood estimation of the decay parameter a in PLA model. In this model a=alpha.

Usage

```
GetFitPLA(z, MeanZeroQ = FALSE, algorithm=c("emle", "wmle"), ciQ=FALSE)
```

Arguments

`z`	time series data vector
`MeanZeroQ`	optional argument, default is MeanZeroQ=FALSE. Set to TRUE if the mean is known to be zero
`algorithm`	"emle" for exact MLE or "wmle" for Whittle method
`ciQ`	TRUE or FALSE according as 95 percent confidence interval computed and plotted

Value

a list with four/five elements:

`p`	MLE for a
`Loglikelihood`	value of the maximized loglikelihood
`alpha`	MLE for a
`ci`	95 percent confidence interval for a
`algorithm`	either "emle" or "wmle"

Author(s)

A. I. McLeod and Justin Veenstra

See Also

`GetFitFD`

Examples

```
#Example 1
#fit Gaussian White Noise, alpha=1
z<-rnorm(500, 100, 10)
GetFitPLA(z)

#Example 2
#estimate alpha for NileMin series
data(NileMin)
GetFitPLA(NileMin)
```

`GetFitPLS` *Fit PLS Time Series Model*

Description

Exact maximum likelihood estimation of the decay parameter p in PLS model. In this model p = alpha.

Usage

```
GetFitPLS(z, MeanZeroQ = FALSE, algorithm=c("emle", "wmle"), ciQ=FALSE)
```

Arguments

`z`	time series data vector
`MeanZeroQ`	optional argument, default is MeanZeroQ=FALSE. Set to TRUE if the mean is known to be zero
`algorithm`	"emle" for exact MLE or "wmle" for Whittle method
`ciQ`	TRUE or FALSE according as 95 percent confidence interval computed and plotted

Value

a list with four/five elements:

`p`	MLE for p
`Loglikelihood`	value of the maximized loglikelihood
`alpha`	MLE for alpha
`algorithm`	either "emle" or "wmle"
`ci`	95 percent confidence interval for p

Author(s)

A. I. McLeod and Justin Veenstra

See Also

GetFitFD

Examples

```
#Example 1
#fit Gaussian White Noise, alpha=1
z<-rnorm(500, 100, 10)
GetFitPLS(z)

#Example 2
#estimate alpha for NileMin series
data(NileMin)
GetFitPLS(NileMin)
```

---

globtp *Annual global temperature, 1880-1985*

Description

Surface air temperature departure from from a set value.

Usage

```
data(globtp)
```

Format

The format is: Time-Series [1:106] from 1880 to 1985: -0.4 -0.37 -0.43 -0.47 -0.72 -0.54 -0.47 -0.54 -0.39 -0.19 ...

Source

James Hansen and Sergej Lebedeff, Global Trends of Measured Surface Air Temperature, Journal of Geophysical Research, Vol. 92, No. D11, pages 13,345-13,372, November 20, 1987.

Examples

```
data(globtp)
plot(globtp)
```

HurstK *Hurst K Coefficient*

Description

The Hurst K provides a non-parametric estimate for the Hurst H coefficient

Usage

```
HurstK(z)
```

Arguments

z	time series vector

Details

There are many alternative non-parametric estimators for H. Some of the popular ones are discussed in Hipel and McLeod (2005).

Value

an estimate of H

Author(s)

A.I. McLeod

References

Hipel, K.W. and McLeod, A.I., (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/.

McLeod, A.I., Yu, Hao, Krougly, Zinovi L. (2007). Algorithms for Linear Time Series Analysis, Journal of Statistical Software.

See Also

FitFGN

Examples

```
# the Hurst coefficient for NID series is 0.5
z<-rnorm(1000)
HurstK(z)
#Hurst K for Nile Minima
data(NileMin)
HurstK(NileMin)
```

LLFD *Concentrated Loglikelihood Function for d*

Description

The concentrated loglikelihood, that is, the loglikelihood function maximized over the innovation variance parameter, is computed.

Usage

```
LLFD(d, z)
```

Arguments

d	fractional difference parameter
z	time series

Value

the value of the loglikelihood

Author(s)

A. I. McLeod

See Also

GetFitFD

Examples

```
#compute loglikelihood for NileFlowCMS with
#  H=0.9 and with d=H-0.5=0.4
data(NileFlowCMS)
z<-NileFlowCMS
z<-z-mean(z)
LLFGN(0.9, z)
LLFD(0.4, z)
```

LLFGN — *Concentrated Loglikelihood Function for H*

Description

The concentrated loglikelihood, that is, the loglikelihood function maximized over the innovation variance parameter, is computed.

Usage

```
LLFGN(H, z)
```

Arguments

H	parameter
z	data vector, assumed to be mean corrected

Value

the value of the loglikelihood

Author(s)

A.I. McLeod

References

McLeod, A.I., Yu, Hao, Krougly, Zinovi L. (2007). Algorithms for Linear Time Series Analysis, Journal of Statistical Software.

See Also

FitFGN, DLLoglikelihood

Examples

```
#compute loglikelihood for NileFlowCMS with H=0.9
data(NileFlowCMS)
z<-NileFlowCMS
z<-z-mean(z)
LLFGN(0.9, z)


#simulate Gaussian white noise and tabulate the loglikelihood for H=0.40, 0.45, 0.50, 0.55, 0.60
set.seed(4321)
h<-c(0.40, 0.45, 0.50, 0.55, 0.60)
z<-rnorm(500, 100, 50)
z<-z-mean(z)
LL<-numeric(length(h))
```

```
for (i in 1:length(h))
LL[i]<-LLFGN(h[i],z)
matrix(c(h,LL),ncol=2)
```

LLPLA *Concentrated Loglikelihood Function for parameter alpha in the PLA model*

Description

The concentrated loglikelihood, that is, the loglikelihood function maximized over the innovation variance parameter, is computed.

Usage

```
LLPLA(alpha, z)
```

Arguments

alpha	decay parameter
z	time series

Value

the value of the loglikelihood

Author(s)

A. I. McLeod

See Also

GetFitPLA

Examples

```
#compute loglikelihood for NileFlowCMS with
#  H=0.9 and with alpha=2-2*H=0.2
data(NileFlowCMS)
z<-NileFlowCMS
z<-z-mean(z)
LLFGN(0.9, z)
LLPLA(0.2, z)
```

LLPLS *Concentrated Loglikelihood Function for alpha in PLS model*

Description

The concentrated loglikelihood, that is, the loglikelihood function maximized over the innovation variance parameter, is computed.

Usage

```
LLPLS(alpha, z)
```

Arguments

alpha	decay parameter
z	time series

Value

the value of the loglikelihood

Author(s)

A. I. McLeod

See Also

GetFitPLS

Examples

```
#compute loglikelihood for NileFlowCMS with
#  H=0.9 and with alpha=2-2*H=0.2
data(NileFlowCMS)
z<-NileFlowCMS
z<-z-mean(z)
LLFGN(0.9, z)
LLPLS(0.2, z)
```

NileFlowCMS *Annual flow of Nile River at Aswan, 1871-1945*

Description

This is average annual flow of the Nile River below the Aswan Dam. The units are CMS (cubic meters per second).

Usage

```
data(NileFlowCMS)
```

Format

The format is: Time-Series [1:75] from 1870 to 1944: 3958 3370 3485 3438 3702 ...

Source

Hipel, K.W. and McLeod, A.I., (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

```
#Plot the time series
data(NileFlowCMS)
ts.plot(NileFlowCMS)

#Hurst K estimate
HurstK(NileFlowCMS)
```

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")= "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 full dataset is available StatLib Datasets 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

```
#Example 1
#Compute Hurst's K estimate of H
data(NileMin)
HurstK(NileMin)
GetFitFGN(NileMin)

#Example 2
#Script for comparing FGN/ARMA forecast performance
#
## Not run:
data(NileMin)
outNileMin<-FitFGN(NileMin)
set.seed(12177)
z<-Boot(outNileMin)
n<-length(z)
K<-100 #number of out-of-sample data values
z1<-z[1:(length(z)-K)] #training data
z2<-z[-(1:(length(z)-K))] #testing data
#
#FGN fit to z1 and forecast using z2
maxLead<-3
n1<-length(z1)
outz1<-FitFGN(z1)
H<-outz1$H
mu<-outz1$muHat
rFGN<-var(z1)*acvfFGN(H, n + maxLead -1)
F<-TrenchForecast(c(z1,z2), rFGN, mu, n1, maxLead=maxLead)$Forecasts
nF<-nrow(F)
err1<-z2-F[,1][-nF]
err2<-z2[-1]-F[,2][-c(nF,(nF-1))]
err3<-z2[-c(1,2)]-F[,3][-c(nF,(nF-1),(nF-2))]
rmse1<-sqrt(mean(err1^2))
```

```
rmse2<-sqrt(mean(err2^2))
rmse3<-sqrt(mean(err3^2))
FGNrmse<-c(rmse1,rmse2,rmse3)
#
#ARMA(p,q) fit to z1 and forecast using z2
p<-2
q<-1
ansz1<-arima(z1, c(p,0,q))
phi<-theta<-numeric(0)
if (p>0) phi<-coef(ansz1)[1:p]
if (q>0) theta<-coef(ansz1)[(p+1):(p+q)]
zm<-coef(ansz1)[p+q+1]
sigma2<-ansz1$sigma2
vz<-tacvfARMA(phi=phi, theta=theta, sigma2=sigma2, maxLag=0)
r<-vz*ARMAacf(ar=phi, ma=theta, lag.max=n + maxLead -1)
F<-TrenchForecast(c(z1,z2), r, zm, n1, maxLead=3)$Forecasts
err1<-z2-F[,1][-nF]
err2<-z2[-1]-F[,2][-c(nF,(nF-1))]
err3<-z2[-c(1,2)]-F[,3][-c(nF,(nF-1),(nF-2))]
rmse1<-sqrt(mean(err1^2))
rmse2<-sqrt(mean(err2^2))
rmse3<-sqrt(mean(err3^2))
ARMArmse<-c(rmse1,rmse2,rmse3)
#
#tabulate result
tb<-matrix(c(FGNrmse,ARMArmse),ncol=2)
dimnames(tb)<-list(c("lead1","lead2","lead3"),c("FGN","ARMA"))

## End(Not run)
```

`plot.FitFGN` *Plot Method for "FitFGN" Object*

Description

Diagnostic plots of the residual autocorrelations and Ljung-Box test.

Usage

```
## S3 method for class 'FitFGN'
plot(x, maxLag=30, ...)
```

Arguments

x	object of class "FitFGN"
maxLag	maximum lag in residual acf plot
...	optional arguments

Details

The top plot shows the residual autocorrelations and their 5% signficance limits. The bottom plot shows the p-values of the Ljung-Box test for various lags.

Value

No value is returned. A plots are produced as side-effect. The plot is a two-panel disply showing the residual autocorrelations and the p-values for the Ljung-Box test.

Author(s)

A.I. McLeod

References

Ljung, G.M., Box, G.E.P. (1978). On a Measure of Lack of Fit in Time Series Models. Biometrika 65, 297-303.

McLeod, A.I., Yu, Hao, Krougly, Zinovi L. (2007). Algorithms for Linear Time Series Analysis, Journal of Statistical Software.

See Also

`summary.FitFGN`, `FitFGN`

Examples

```
data(NileMin)
obj<-FitFGN(NileMin, c(1,2,6,7))
plot(obj)
```

`predict.FitFGN` *Forecasts from a fitted FGN model*

Description

The exact finite-sample minimum mean square error forecasts are computed using the Trench algorithm.

Usage

```
## S3 method for class 'FitFGN'
predict(object, n.ahead = 1, ...)
```

Arguments

`object`	"FitFGN" object produced by FitFGN
`n.ahead`	forecasts are done for lead times 1,...,n.ahead
`...`	optional arguments, are ignored

Value

A list with components

`Forecasts`	matrix with m+1 rows and maxLead columns with the forecasts
`SDForecasts`	matrix with m+1 rows and maxLead columns with the sd of the forecasts

Author(s)

A.I. McLeod

References

McLeod, A.I., Yu, Hao, Krougly, Zinovi L. (2007). Algorithms for Linear Time Series Analysis, Journal of Statistical Software.

See Also

`FitFGN`, `TrenchForecast` `PredictionVariance` `predict.Arima`

Examples

```
data(NileMin)
out<-FitFGN(NileMin)
predict(out, n.ahead=15)
```

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

Description

A terse summary is given.

Usage

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

Arguments

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

Value

A terse summary is displayed

Author(s)

A.I. McLeod

References

McLeod, A.I., Yu, Hao, Krougly, Zinovi L. (2007). Algorithms for Linear Time Series Analysis, Journal of Statistical Software.

See Also

`summary.FitFGN`, `FitFGN`

Examples

```
data(NileMin)
FitFGN(NileMin)
```

`Reimann` *Computes Reimann zeta function on [0,3]*

Description

Computes Reimann zeta function on [0,3] by interpolating a table of values. Since the function has a pole at 1, values of the function in the interval [0.999,1.001] are NA. This function is used to define acfPLA.

Usage

```
Reimann(v)
```

Arguments

`v`	argument, between 0 and 3

Value

The function value is returned.

Author(s)

A. I. McLeod

See Also

`acvfPLA`

Examples

```
Reimann(0.5)
```

residuals.FitFGN *Extract Residuals from "FitFGN" Object*

Description

Method function.

Usage

```
## S3 method for class 'FitFGN'
residuals(object, ...)
```

Arguments

object	object of class "FitFGN"
...	optional arguments

Value

Vector of standardized prediction residuals

Author(s)

A.I. McLeod

See Also

FitFGN

Examples

```
data(NileMin)
out<-FitFGN(NileMin)
qqnorm(resid(out))
```

sdfarma *ARMA spectral density function*

Description

The spectral density function is evaluated at the Fourier frequencies, 2*pi*seq(from=1/n, to=1/2, by=1/n) for the ARMA model.

Usage

```
sdfarma(n, phi = numeric(0), theta = numeric(0))
```

Arguments

n	length of series, number of Fourier frequencies is [n/2]
phi	autoregressive coefficients
theta	moving-average coefficients

Value

vector of spectral density values corresponding to the Fourier frequencies

Author(s)

A. I. McLeod

Examples

```
sdfarma(100, 0.9, 0.7)
```

---

sdfFD *Spectral density function for FD*

Description

The spectral density function is evaluated at the Fourier frequencies, k/n, k=1,...,[n/2], for FD (fractionally differenced white noise).

Usage

```
sdfFD(d, n)
```

Arguments

d	FD parameter
n	series length

Value

vector of the sdf values

Author(s)

A. I. McLeod

See Also

GetFitFD

Examples

```
sdfFD(0.2, 50)
```

sdfFGN *Spectral density function for FGN*

Description

Computes the spectral density function for the FGN model with parameter H at the Fourier frequencies, 2*Pi*j/n, j=1,...,[n/2], where n is the length of the time series. The evaluation is very fast since bivariate interpolation is used.

Usage

```
sdfFGN(H, n)
```

Arguments

H	FGN parameter
n	length of time series

Details

The details of the implementation are discussed in the accompanying vignette.

Value

a vector of length [n/2] of the spectral density values.

Author(s)

A. I. McLeod and J. Veenstra

See Also

sdfFD

Examples

```
sdfFGN(0.7, 100)
```

`sdfhd` *Spectral density of hyperbolic decay models*

Description

The spectral density function is evaluated at the Fourier frequencies, 2*pi*seq(from=1/n, to=1/2, by=1/n) for various types of hyperbolic decay time series models.

Usage

```
sdfhd(n, alpha = 1, phi = numeric(0), theta = numeric(0),
  lmodel = c("FD", "FGN", "PLA", "NONE"))
```

Arguments

`n`	series length, number of Fourier frequencies [n/2]
`alpha`	canonical hyperbolic decay parameter
`phi`	autoregressive coefficients
`theta`	moving-average coefficients, Box-Jenkins style
`lmodel`	type of hyperbolic decay model

Value

vector of values of the spectral density function at the Fourier frequencies

Author(s)

A. I. McLeod

Examples

```
sdfhd(100, 0.2)
```

`sdfPLA` *Spectral density function for PLA*

Description

Computes the spectral density function for the PLA model with parameter a at the Fourier frequencies, 2*Pi*j/n, j=1,...,[n/2], where n is the length of the time series. The evaluation is very fast. Bivariate interpolation and asymptotic approximation are used.

Usage

```
sdfPLA(a, n)
```

Arguments

a	PLA parameter
n	length of time series

Details

The details of the implementation are discussed in the accompanying vignette. The parameter a should be in the interval (0,2). series n should be greater than 2.

Value

a vector of length [n/2] of the spectral density values.

Author(s)

A. I. McLeod and J. Veenstra

See Also

sdfFD

Examples

```
sdfPLA(0.2, 100)
```

---

sdfPLS *Spectral density function for PLS*

Description

Computes the spectral density function for the PLS model with parameter a at the Fourier frequencies, 2*Pi*j/n, j=1,...,[n/2], where n is the length of the time series.

Usage

```
sdfPLS(p, n)
```

Arguments

p	PLS parameter
n	length of time series

Details

The details of the implementation are discussed in the accompanying vignette. The parameter p should be in the interval (0, 2) and the length of the series n should be greater than 2.

Value

a vector of length [n/2] of the spectral density values.

Author(s)

A. I. McLeod

See Also

`sdfFD`

Examples

```
sdfPLS(0.2, 100)
```

SeriesB *Series B, close price IBM stock*

Description

Closing price of IBM common stock, daily, May 17 1961 to November 2 1962

Usage

```
data(SeriesB)
```

Format

The format is: num [1:369] 460 457 452 459 462 459 463 479 493 490 ...

Source

G. E. P. Box, G.E.P., Jenkins, G.M. and Reinsel, G.C. (2008). Time Series Analysis: Forecasting and Control, 4th Ed., Wiley.

Examples

```
r <- diff(log(SeriesB))
a <- abs(diff(log(SeriesB)))
layout(c(2,1))
acf(r, main="log returns")
acf(a, main="absolute log returns")
#
```

`SimulateFD` *Simulates FD*

Description

A fractional Gaussian noise time series is simulated.

Usage

```
SimulateFD(n, d)
```

Arguments

`n`	length of time series
`d`	fractional difference parameter

Details

The FFT is used so it is most efficient if you select n to be a power of 2. Note, d=H-1/2.

Value

vector of length containing the simulated time series

Author(s)

A.I. McLeod

References

Davies, R. B. and Harte, D. S. (1987). Tests for Hurst Effect. Biometrika 74, 95–101.

McLeod, A.I., Yu, Hao, Krougly, Zinovi L. (2007). Algorithms for Linear Time Series Analysis, Journal of Statistical Software.

See Also

`SimulateFGN`, `DLSimulate`

Examples

```
#Example 1
#simulate a process with H=0.2 and plot it
z<-SimulateFD(100, 0.2)
ts.plot(z)
```

`SimulateFGN` *Simulates FGN*

Description

A fractional Gaussian noise time series is simulated.

Usage

```
SimulateFGN(n, H)
```

Arguments

`n`	length of time series
`H`	Hurst coefficient

Details

The FFT is used so it is most efficient if you select n to be a power of 2.

Value

vector of length containing the simulated time series

Author(s)

A.I. McLeod

References

Davies, R. B. and Harte, D. S. (1987). Tests for Hurst Effect. Biometrika 74, 95–101.

McLeod, A.I., Yu, Hao, Krougly, Zinovi L. (2007). Algorithms for Linear Time Series Analysis, Journal of Statistical Software.

See Also

`DLSimulate`

Examples

```
#Example 1
#simulate a process with H=0.2 and plot it
z<-SimulateFGN(100, 0.2)
ts.plot(z)
#
#Example 2
#simulate FGN and compare theoretical and sample autocovariances
H<-0.7
```

```
n<-8192
z<-SimulateFGN(n, H)
#autocovariances
sacvf<-acf(z, plot=FALSE,type="covariance")$acf
tacf<-acvfFGN(H, n-1)
tb<-matrix(c(tacf[1:10],sacvf[1:10]),ncol=2)
dimnames(tb)<-list(0:9, c("Tacvf","Sacvf"))
tb
```

---

summary.FitFGN *Summary Method for "FitFGN" Object*

Description

summary for "FitFGN" object.

Usage

```
## S3 method for class 'FitFGN'
summary(object, ...)
```

Arguments

`object`	"FitFGN" object
`...`	optional arguments

Value

A printed summary is given

Author(s)

A.I. McLeod

See Also

`print.FitFGN`, `FitFGN`

Examples

```
data(NileMin)
out<-FitFGN(NileMin)
summary(out)
```

`warfima` *Whittle mle arfima*

Description

Fit time series arfima model using Whittle approximate maximum likelihood estimation.

Usage

```
warfima(z, order = c(0, 0, 0), lmodel = c("FD", "FGN", "PLA", "PLS", "NONE"))
```

Arguments

`z`	vector of time series values
`order`	c(p,d,q), where is AR order, d is differencing, q is MA order
`lmodel`	type of hyperbolic decay model

Value

list with components:

`bHat`	transformed optimal parameters
`alphaHat`	estimate of alpha
`HHat`	estimate of H
`dHat`	estimate of d
`phiHat`	estimate of phi
`thetaHat`	estimate of theta
`LL`	optimized value of Whittle approximate log-likelihood
`convergence`	convergence indicator
`algorithm`	optimization algorithm used, 1 for L-BFGS-B, 2 for Nelder-Mead, 3 for SANN

Author(s)

A. I. McLeod

Examples

```
warfima(NileMin, lmodel="FGN")
```

Index