Package ‘afmtools’

February 15, 2013

Type Package

Version 0.1.8

Date 2011-04-15

Title Estimation, Diagnostic and Forecasting Functions for ARFIMA models

Author Javier E. Contreras-Reyes, Georg M. Goerg, Wilfredo Palma

Maintainer Javier E. Contreras-Reyes <jecontrre@mat.puc.cl>

Depends R (>= 2.6.0), polynom, fracdiff, hypergeo, sandwich, longmemo

Description A collection of estimation, forecasting and diagnostic tools for autoregressive fractionally integrated moving-average process (ARFIMA).

License GPL (>= 2)

LazyLoad yes

Repository CRAN

Date/Publication 2012-12-28 22:23:38

NeedsCompilation no

R topics documented:

afmtools-package ................................................. 2
arfima-methods .................................................. 4
arfima.whittle ................................................. 5
arfima.whittle.loglik .......................................... 7
check.parameters.arfima ....................................... 9
gw.test .......................................................... 10
ir.arfima ....................................................... 12
MammothCreek .................................................. 13
per.arfima ...................................................... 14
pi.j ............................................................ 15
afmtools-package

Estimation, Diagnostic and Forecasting functions for ARFIMA models

Description

A collection of estimation, forecasting and diagnostic tools for autoregressive fractionally integrated moving-average process (ARFIMA).

Details

<table>
<thead>
<tr>
<th>Package</th>
<th>afmtools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Package</td>
</tr>
<tr>
<td>Version</td>
<td>0.1.8</td>
</tr>
<tr>
<td>Date</td>
<td>2011-03-22</td>
</tr>
<tr>
<td>License</td>
<td>GPL &gt;= 2</td>
</tr>
<tr>
<td>LazyLoad</td>
<td>yes</td>
</tr>
</tbody>
</table>

Functions

The package includes several functions. The following ones are those more relevant for practical use:

- summary, plot, print, residuals and tsdiag options associated to arfima class object and residuals diagnostic. arfima.whittle and arfima.whittle.loglik for Whittle estimation (produce an arfima class object) and the log-likelihood function. gw.test is the forecasting tool. spectrum.arfima, rho.sowell and var.afm produce the spectrum density, autocovariance function and parameter variance for ARFIMA models, respectively.

It is suggested that the user starts by reading the documentation of (some of) these functions.

Requirements

- R >= 2.6.0
- Packages fracdiff, polynom, longmemo, sandwich and hypergeo.
Licence

This package and its documentation are usable under the terms of the GNU General Public License, a copy of which is distributed with the package. While the software is freely usable, it would be appreciated if a reference is inserted in publications or other work which makes use of it.

Author(s)

Javier E. Contreras-Reyes, Instituto de Fomento Pesquero, Valparaiso, Chile. <Email: jecontrrr@mat.puc.cl>.
Georg M. Goerg, Department of Statistics, Carnegie Mellon University. <Email: gmg@stat.cmu.edu>
Wilfredo Palma, Departamento de Estadistica, Facultad de Matematicas, Pontificia Universidad Catolica de Chile. <Email: wilfredo@mat.puc.cl>
Maintainer: Javier E. Contreras-Reyes <jecontrrr@mat.puc.cl>

References

arfima-methods

Methods for fitted ARFIMA models

Description

summary, print, residuals, tsdiag and plot methods for class arfima model. A equivalent
function of summary is provided in afmtools package called summary.arfima. tsdiag its a generic
diagnostic function which produces several plots of the residual from a fitted ARFIMA model.

Usage

```r
## S3 method for class 'arfima'
plot(x, ...)
## S3 method for class 'arfima'
residuals(object, ...)
## S3 method for class 'arfima'
summary(object, ...)
## S3 method for class 'arfima'
print(x, ...)
## S3 method for class 'arfima'
tsdiag(object, gof.lag = 1/zero.noslash, ...)
```

Arguments

- `object, x` object of class arfima; usually a result of a call to `arfima.whittle`
- `gof.lag` the maximum number of lags for a Portmanteau goodness-of-fit test
- `...` further arguments passed to or from other methods.

Details

`plot` produces 4 figure: 1) AR and MA roots of the model; 2) Sample ACF and theoretical ACF
implied by the estimates; 3) Periodogram and theoretical spectrum implied by the estimates; 4)
Sample ACF of Residuals

`summary` (and basically the same for `print`) gives a summary output in the `summary.lm` style - i.e.
parameter estimates, standard errors, significance, etc.

`residuals` gives the residuals from the estimated ARFIMA model. This is not implemented di-
rectly via the $AR(\infty)$ represenation of an ARFIMA(p,d,q) process, but using a “trick”: first the
original series is differenced with $\hat{d}$ using `diffseries` in the package `fracdiff`. Consequently an
ARMA(p,q) should remain. Now instead of estimating the parameters again, an ARMA(p,q) model
where ALL parameters are fixed to the Whittle estimates is “estimated” with `arima` and then the
residuals are obtained.

Value

For `residuals` a vector of class `ts`; for `summary` and `print` the fitted ARFIMA model object. For
`tsdiag` produce plots of standardized residuals, autocorrelation function of the residuals, and the
p-values of a Portmanteau test for all lags up to `gof.lag`. 
arfima.whittle

Author(s)
Georg M. Goerg, Javier E. Contreras-Reyes

References

See Also
Box.test

Examples
```r
data(MammothCreek)
y=MammothCreek-mean(MammothCreek)
mod <- arfima.whittle(y,nar=1,nma=1)
res=residuals(mod)
summary(mod)
print(mod)
plot(res)
acf(res)
tsdiag(mod, gof.lag=12)
```

arfima.whittle                      Whittle Estimation for ARFIMA models

Description
Estimates the parameters of an ARFIMA(p,d,q) model via the Whittle method. Parameters can be fixed to 0 by the fixed option.

Usage
arfima.whittle(series, nar = 0, nma = 0, fixed = NA)

Arguments
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>series</td>
<td>a numerical vector with time series</td>
</tr>
<tr>
<td>nar</td>
<td>number of AR parameters p</td>
</tr>
<tr>
<td>nma</td>
<td>number of MA parameters q</td>
</tr>
</tbody>
</table>
The `arfima` function in R provides a method of Whittle estimation for fitting ARFIMA models. The function allows fixing parameters to 0. By default, no parameters are fixed. The function works by counting the number of parameters in the model and setting the specified ones in `fixed` to 0. The order of parameters is 1) long memory parameter \( d \); 2) AR parameters; 3) MA parameters. For example, for an ARFIMA(2, d, 1) model, there are 4 parameters; to fix the second AR parameter \( \phi_2 \), the 3rd parameter has to be set equal to zero, i.e., \( fixed=3 \).

### Value
- `arfima`: an object of class `arfima`
- `call`: function call
- `method`: method of Whittle estimation
- `data`: time serie
- `par`: estimated parameter \( \theta \)
- `loglik`: value of the loglikelihood function
- `d`: estimated \( d \)
- `ar`: estimated AR parameters
- `ma`: estimated MA parameters
- `nar`: number of AR parameters \( p \)
- `nma`: number of MA parameters \( q \)
- `fixed`: indicator of possible fixed values
- `sd.innov`: innovation standard deviation; estimated by the ratio of theoretical spectrum and periodogram
- `model`: character string of the specified ARFIMA\( (p, d, q) \) model
- `spectrum`: theoretical spectrum implied by the parameter estimates

### Author(s)
Wilfredo Palma, Georg M. Goerg

### References

### See Also
- `arima`
Examples

data(MammothCreek)
x=MammothCreek-mean(MammothCreek)
plot(x)
x11()
acf(x)  ## sample ACF displays long range dependence

mod=arfima.whittle(x)  # estimate just the long memory parameter
summary(mod)  ## highly significant; but AR and/or MA parameters
## have to be considered too

mod11=arfima.whittle(x, nar=1, nma=1)  # AR parameter non-significant
summary(mod11)

mod01=arfima.whittle(x, nar=0, nma=1)  # d increased even more
summary(mod01)
plot(mod01)

res=residuals(mod01)
acf(res)
Box.test(res, 10, type="Ljung")

arfima.whittle.loglik  Whittle loglikelihood for ARFIMA models

Description

Computes the Whittle loglikelihood of an ARFIMA model for the data.

Usage

arfima.whittle.loglik(theta, series, nar, nma, fixed)

Arguments

theta  parameter value at which the loglikelihood function should be evaluated
series  a numerical vector of time series
nar  number of AR parameters p
nma  number of MA parameters q
fixed  option to fix parameters to 0. By default no parameters are fixed. Works as
## counting the number of parameters in the model and setting the specified ones
## in fixed to 0. The order of parameters is 1) long memory parameter d; 2) AR
## parameters; 3) MA parameters. For example, for an ARFIMA(2,d,1) model we
## have four parameters and we need to fix phi_2 (the second AR parameter) to
## zero, this is the third parameter of the list: (d, phi_1, phi_2, theta_1). On this
## case, we put fixed=3 to fix phi_2=0
Details

The approximate maximum-likelihood estimates consist in

\[ L(\theta) = -\frac{1}{2n} \left[ \log(|\Gamma_\theta|) - Y'\Gamma_\theta^{-1}Y \right] \]

with

\[ (\Gamma_\theta)_{ij} = \gamma_\theta(i-j) = \int_{-\pi}^{\pi} f_\theta(\lambda)e^{i(i-j)\lambda}d\lambda \]

Value

- \text{L} \quad \text{value of the loglikelihood}
- \text{sigma2} \quad \text{innovation variance; estimated by the ratio of theoretical spectrum and periodogram}

Author(s)

Georg M. Goerg

References


See Also

arfima.whittle, per

Examples

data(MammothCreek)
x=MammothCreek-mean(MammothCreek)
param=c(0.11, 0.39, -0.28)
loglik=arfima.whittle.loglik(theta=param, series=x, nar=1, nma=1)
loglik$L
loglik$sigma

####### For other examples, see 'arfima.whittle' function
Parameter check for ARFIMA models

Description

Given the autoregressive and moving average and fractional coefficients, this function checks the stationarity of the process.

Usage

check.parameters.arfima(d = 0, ar = 0, ma = 0, plot = FALSE)

Arguments

d  long memory parameter
ar  autoregressive parameters
ma  moving average parameters
plot should the output be plotted

Value

d          TRUE/FALSE indicators for the fractional parameter
ar         TRUE/FALSE indicators for the autoregressive polynomial
ma         TRUE/FALSE indicators for the mobile average polynomial
model.OK   TRUE/FALSE indicators for the whole model

Side Effects

A graphical output is produced if plot==TRUE. This plot consists of the unit circle and the inverse AR and MA Roots.

Author(s)

Georg M. Goerg

References


Examples

```r
data(MammothCreek)
y = MammothCreek - mean(MammothCreek)
mod <- arfima.whittle(y, nar=1, nma=1)
check.parameters.arfima(d=mod$d, ar=mod$ar, ma=mod$ma, plot=TRUE)
```

```r
######## Some particular cases ########
# Invertible process
check.parameters.arfima(d=0.2, ar=-0.4, ma=0.7, plot=TRUE)
# No invertible process
check.parameters.arfima(d=0.2, ar=-0.4, ma=c(0.7, -0.9), plot=TRUE)
# Causal process
check.parameters.arfima(d=-0.5, ar=0.4, ma=-0.7, plot=TRUE)
# No causal process
check.parameters.arfima(d=-0.5, ar=-1.4, ma=-0.7, plot=TRUE)
# Fractional parameter in the (-1, 0.5) interval
check.parameters.arfima(d=-0.7, ar=0.4, ma=0, plot=TRUE)
# Fractional parameter outside the (-1, 0.5) interval
check.parameters.arfima(d=0.5, ar=0.4, ma=0, plot=TRUE)
```

---

**gw.test**

*Giacomini & White test of Predictive Ability*

**Description**

Forecasting evaluation method proposed by Giacomini and White (GW) to compare two vectors of predictions provided by two time series models.

**Usage**

```r
gw.test(x, y, p, T, tau,
        method=c("HAC","NeweyWest","Andrews","LumleyHeagerty"),
        alternative=c("two.sided","less","greater"))
```

**Arguments**

- `x`: a numeric vector of data values corresponding to model 1
- `y`: a numeric vector of data values corresponding to model 2
- `p`: a numeric vector of data values corresponding to observations
- `T`: sample total size
- `tau`: forecasting horizon parameter. If `tau=1`, the *Standard Statistic Simple Regression Estimator* method is used. If `tau>1`, the chosen method by the user is used only if `tau>1` is chosen, method able to select between a set of Matrix Covariance Estimation methods, such as HAC, NeweyWest, Andrews and LumleyHeagerty
- `method`: a character string specifying the alternative hypothesis, must be one of `two.sided` (default), greater or less
- `alternative`: a character string specifying the alternative hypothesis, must be one of `two.sided` (default), greater or less
gw.test

Value

- **statistic**: the value of the GW statistic
- **alternative**: a character string describing the alternative hypothesis
- **p.value**: the p-value for the test
- **method**: a character string indicating what type of Matrix Covariance Estimation method was performed
- **data.name**: a character string giving the name(s) of the data

Author(s)

Javier E. Contreras-Reyes

References


See Also

*predict*

Examples

```r
r = 100
s = 30
y = arima.sim(n = r, list(ar = c(0.8, -0.4),
ma = c(-0.2, 0.3)), sd = sqrt(0.18))
y.real = y[(length(y)-s+1):length(y)]
obs = y[1:(r-s)]
mod.arma <- arima(obs, order = c(1,0,1))
mod.arima <- arima(obs, order = c(1,1,1))
P = matrix(NA, s, 2) # matrix of predictions

for(i in 1:s) {
  p.arma <- predict(mod.arma, ahead = 1)
p.arima <- predict(mod.arima, ahead = 1)
P[i,] = c(p.arma$pred, p.arima$pred)
obs = c(obs, y.real[i])
mod.arma <- arima(obs, order = c(1,0,1))
mod.arima <- arima(obs, order = c(1,1,1))
}
gw.test(x = P[,1], y = P[,2], p = y.real, T = length(y),
tau = 1, method = "HAC", alternative = "less")
```
**ir.arfima**

*Impulse Response Function for ARFIMA models*

**Description**

Compute the Impulse Response (IR) associated to ARFIMA(p,d,q) model. Consider the Normal and Asymptotic (Hassler and Kokoszka, 2010) method(s). See below for details.

**Usage**

`ir.arfima(h, model, plot.it = TRUE)`

**Arguments**

- `h`: a numeric value for number of lags
- `model`: a fitted time-series model of ARFIMA class
- `plot.it`: a TRUE/FALSE indicator for plotting the impulse response function

**Details**

For a fractional noise ARFIMA(0,d,0) process, we have the MA coefficients

\[ \psi_j = \frac{\Gamma(j + d)}{\Gamma(j + 1)\Gamma(d)}. \]

For a general ARFIMA(p,d,q) process we have the asymptotic approximation of MA coefficients

\[ \psi_j \sim \left( \frac{1 + \sum_{i=1}^{q} \theta_i}{1 - \sum_{i=1}^{p} \phi_i} \right)^{j^{d-1}} \frac{\Gamma(d)}{\Gamma(1 + d)}. \]

for a large \( j \). The IR function is computed first via the Normal method

\[ R_j = \sum_{i=0}^{j} \psi_i \eta_{j-i}, \]

where \( \eta_t = \frac{\Gamma(t + d)}{\Gamma(t + 1)\Gamma(d)} \). Then, the IR function is computed via the Asymptotic method

\[ R_j \sim \frac{j^{d-1}}{\Gamma(d)} \sum_{i=0}^{\infty} \psi_i, \]

for a large \( j \).
Value

h       a numeric vector with values 1 to h
RE      a numeric vector of Normal impulse response function
RA      a numeric vector of asymptotic impulse response function
psi.j   a numeric value of the asymptotic MA coefficients
int     a numeric vector of coordinates of the intersection between Normal IR and Asymptotic IR functions

Author(s)

Javier E. Contreras-Reyes

References


See Also

psi.j, arfima.whittle

Examples

data(MammothCreek)
y=MammothCreek-mean(MammothCreek)
mod <- arfima.whittle(y,nar=0,nma=0)
mod1 <- arfima.whittle(y,nar=1,nma=1)

ir.arfima(50, mod)
ir.arfima(50, mod1)
ir.arfima(100, mod)
ir.arfima(100, mod1)
ir.arfima(150, mod)
ir.arfima(150, mod1)

MammothCreek

*Mammoth Creek Time Series Data*

Description

Centered annual pinus longaeva tree ring time series width measurements at Mammoth Creek, Utah, from 0 A.D to 1989 A.D.
Usage

data(MammothCreek)

Format

MammothCreek is a data frame with 1990 observations.

Source

Available at the National Climatic Data Center, URL http://www.ncdc.noaa.gov/paleo/metadata/noaa-tree-3376.html

References


Examples

data(MammothCreek)
str(MammothCreek)
plot(MammothCreek)

y=MammothCreek-mean(MammothCreek)
plot(y,type="l",xlab="Time (annual)",ylab="width measurements")
acf(y,lag=100,main="")

-------

per.arfima  Periodogram by Fast Fourier Transform

Description

Computes the periodogram of a signal z by the Fast Fourier Transform (FFT).

Usage

per.arfima(z)

Arguments

z  univariate signal / time series z

Value

freq  Fourier frequencies $\omega_j = 2\pi j/n$, where $n$ is the length of z, and $j = 1, \ldots, [n/2]$

spec  corresponding value of the periodogram $I(\omega_j)$
pi.j

Asymptotic Infinite AR expansion

Description

Computes the asymptotic AR coefficients related to a ARFIMA process.

Usage

pi.j(h, ar = 0, ma = 0, d = 0)
Arguments

- **h**: a numeric value related to the selected coefficient
- **ar**: a numeric vector containing the autoregressive polynomial parameters
- **ma**: a numeric vector containing the mobile average polynomial parameters
- **d**: a numeric value of fractional parameter

Details

Under the assumption that the roots of the polynomials $\Phi(B)$ and $\Theta(B)$ are outside the closed unit disk and $d \in (-1, 0.5)$, the $y_t \sim \text{ARFIMA}(p, d, q)$ process is stationary, causal, and invertible. In this case we can write $\varepsilon_t = \Pi(B) y_t$. Then, the AR($\infty$) coefficients, $\pi_j$, associated to $\Pi(B)$ polynomial are given by the following asymptotic relationship

$$\pi_j \sim \frac{\Phi(1) \ j^{-d-1}}{\Theta(1) \ \Gamma(-d)}$$

Value

- a numeric value of the asymptotic AR coefficient

Author(s)

Javier E. Contreras-Reyes

References


See Also

`psi.j`, `predict`

Examples

data(MammothCreek)
y=MammothCreek-mean(MammothCreek)
mod <- arfima.whittle(y,nar=2,nma=1)
h=1:100
pwd=pi.j(h,ar=mod$ar,ma=mod$ma,d=mod$d)
plot(h,pwd,type="l",ylab=" ")

#### Asymptotic convergence

h1=1:24
psi1=psi.j(h1,ar=0,ma=0.2,d=0.3)
h2=25:70
psi2=psi.j(h2,ar=0,ma=0.2,d=0.3)
psi.j

Description
Computes the asymptotic MA coefficients related to a ARFIMA process.

Usage
psi.j(h, ar = 0, dr = 0, d = 0)

Arguments
- h: a numeric value related to the selected coefficient
- ar: a numeric vector containing the autoregressive polynomial parameters
- ma: a numeric vector containing the mobile average polynomial parameters
- d: a numeric value of fractional parameter

Details
Under the assumption that the roots of the polynomials $\Phi(B)$ and $\Theta(B)$ are outside the closed unit disk and $d \in (-1, 0.5)$, the $y_t \sim ARFIMA(p, d, q)$ process is stationary, causal, and invertible. In this case we can write $y_t = \Psi(B)\xi_t$. Then, the MA($\infty$) coefficients, $\psi_j$, associated to $\Psi(B)$ polynomial are given by the following asymptotic relationship

$$\psi_j \sim \frac{\Theta(1) j^{d-1}}{\Phi(1) \Gamma(d)}$$

Value
A numeric value of the asymptotic MA coefficients

Author(s)
Javier E. Contreras-Reyes

References
rho.arma

**Description**

Computes the autocorrelation function for a given ARMA process. Done via an inverse Fourier transform.

**Usage**

```r
rho.arma(object = NULL, lag.max = NULL, ar = object$ar, ma = object$ma)
```

**Arguments**

- `object`: a fitted time-series model of ARFIMA class
- `lag.max`: the maximum number of lags
- `ar`: a numeric vector containing the autoregressive polynomial parameters. By default, the value of `object` is selected
- `ma`: a numeric vector containing the mobile average polynomial parameters. By default, the value of `object` is selected

**Details**

This function is used for `plot` function to plotting the ARFIMA class objects.

**Examples**

```r
data(MammothCreek)
y=MammothCreek-mean(MammothCreek)
mod <- arfima.whittle(y, nar=2, rma=2)
h=1:100
pwd=psi.j(h, ar=mod$ar, ma=mod$ma, d=mod$d)
plot(h, pwd, type="l", ylab=" ")

#### Asymptotic convergence
h1=1:12
psi1=psi.j(h1, ar=-0.4, ma=0.7, d=-0.5)
h2=13:50
psi2=psi.j(h2, ar=-0.4, ma=0.7, d=-0.5)
h=1:50
plot(h, c(psi1, rep(NA, 50-12)), type="l", ylab=" ")
lines(h, c(rep(NA, 12), psi2), lty=2)
```
Value

a numerical value of size lag.max with the autocorrelations function values. In addition, a graphical output is produced if plot==TRUE with the autocorrelation function

Author(s)

Georg M. Goerg

References


See Also

acf, rho.sowell, spectrum.arma

Examples

#### For a ARFIMA model:

data(TreeRing)
y=TreeRing-mean(TreeRing)
mod <- arfima.whittle(y,nar=1,nma=1)
lag=12
r.arma=rho.arma(object=mod, lag.max=lag)

n=length(y)
par(mfrow=c(1,2))
plot(r.arma,type="h",main="ACF",ylab=" ",ylim=c(-2/sqrt(n),1))
abline(h=c(-2/sqrt(n),2/sqrt(n)),col=c(1,2,2),lty=c(1,2,2))
acf(y,lag=lag)

#### For a ARMA model:

sim <- arima.sim(n=1000, list(ar = c(0.8897), ma = c(0.2488)), sd = 1)
r.arma=rho.arma(lag.max=6, ar=0.8897, ma = 0.2488)

n=length(sim)

par(mfrow=c(1,2))
plot(r.arma,type="h",main="ACF",ylab=" ",ylim=c(-2/sqrt(n),1))
abline(h=c(0,-2/sqrt(n),2/sqrt(n)),col=c(1,2,2),lty=c(1,2,2))
acf(sim,lag=6)
Usage

```r
rho.sowell(object = NULL, lag.max = NULL, ar = object$ar,
        ma = object$ma, d = object$d,
        sd.innov = object$sd.innov, plot = TRUE)
```

Arguments

- **object**: a fitted time-series model of ARFIMA class
- **lag.max**: the maximum number of lags for exact Variance-Covariance matrix (Sowell)
- **ar**: a numeric vector containing the autoregressive polynomial parameters. By default, the value of `object` is selected
- **ma**: a numeric vector containing the moving average polynomial parameters. By default, the value of `object` is selected
- **d**: a numeric value of fractional parameter. By default, the value of `object` is selected
- **sd.innov**: a numeric value of innovation standard deviation. By default, the value of `object` is selected
- **plot**: a TRUE/FALSE indicator for plotting the exact autocovariance function

Details

For the case of `lag.max = h \leq 50` (See Palma (2007) example), the asymptotic autocovariance calculus is obtained by Sowell algorithm (See References). In the case of `lag.max = h > 50`, the asymptotic autocovariance calculus is obtained using $c_\gamma |h|^{2d-1}$, where

$$c_\gamma = \frac{\sigma^2}{\pi} \frac{|\Theta(1)|^2}{|\Phi(1)|^2} \Gamma(1 - 2d) \sin(\pi d)$$

and $(\sigma^2 / \pi)|\Theta(1)|^2 / |\Phi(1)|^2$ is obtained by `spectrum arma` function multiplied by 2.

Value

- **av**: a numeric vector of autocovariance function values
- **ac**: a numeric vector of autocorrelation function values

Author(s)

Javier E. Contreras-Reyes

References


See Also

acf, spectrum arma

Examples

data(TreeRing)
y=TreeRing-mean(TreeRing)
mod <- arfima.whittle(y,nar=1,nma=1)
lag=12

par(mfrow=c(1,2))
rho.sowell(object=mod, lag.max=lag)
acf(y,lag=lag)

smv.afm

Exact and asymptotic sample mean variance

Description

Compute the exact sample mean variance and the asymptotic sample mean variance of a time series.

Usage

smv.afm(object,lag.max=NULL,comp=TRUE)

Arguments

object a fitted time-series model of ARFIMA class.
lag.max the maximum number of lags for a exact Variance-Covariance matrix (Sowell).
comp indicator for exact/asymptotic (respectively) sample mean variance.

Details

The exact mean variance of $\bar{y}$ is given by

$$\text{Var}(\bar{y}) = \frac{1}{n} \left[ 2 \sum_{j=1}^{n-1} \left( 1 - \frac{j}{n} \right) \gamma(j) + \gamma(0) \right]$$

where the Sowell exact autocovariance function $\gamma(.)$ is used. In addition, for large $n$, we have the asymptotic mean variance

$$\text{Var}(\bar{y}) \sim \frac{c_\gamma}{d(2d+1)} n^{2d-1}$$

where $c_\gamma = \frac{\sigma^2}{\varphi(\Theta(1))} \Gamma(1-2d) \sin(\pi d)$ and $(\sigma^2/\pi) |\Theta(1)|^2 / |\Phi(1)|^2$ is obtained by spectrum arma function multiplied by 2.
Value

*ν*  
a numeric value of the exact sample mean variance or the asymptotic sample mean variance such as chosen by the user.

Author(s)

Javier E. Contreras-Reyes.

References


See Also

`spectrum.arma, rho.sowell`

Examples

```r
data(MammothCreek)
y=MammothCreek-mean(MammothCreek)
mod <- arfima.whittle(y,nar=1,nma=1)
smv.afm(object=mod,comp=TRUE)
smv.afm(object=mod,comp=FALSE)
```

---

**spectrum.arfima**  
*Theoretical spectrum of an ARFIMA model*

Description

Computes the spectral density $f(\lambda|\Theta)$ for an ARFIMA$(p,d,q)$ model with parameter vector $\Theta = (d, \phi_1, ..., \phi_p, \theta_1, ..., \theta_q, \sigma_\varepsilon)$.

Usage

`spectrum.arfima(d = 0, ar = 0, ma = 0, sd.innov = 1)`

Arguments

- **d** long memory parameter between -1 and 0.5
- **ar** AR parameters
- **ma** MA parameters
- **sd.innov** standard deviation of input white noise
Details

The spectral density of fractional noise \( \{X_t\} \) it is

\[
f_X(\lambda) = \frac{\sigma^2}{2\pi} (2\sin(\lambda/2))^{2d}
\]

For an \( Y_t \sim \text{ARFIMA}(p, d, q) \) process it is

\[
f_Y(\lambda) = \frac{\sigma^2}{2\pi} (2\sin(\lambda/2))^{2d} \left| \frac{\Theta(e^{i\lambda})}{\Phi(e^{i\lambda})} \right|^2
\]

where \( \Phi(z) \) and \( \Theta(z) \) are the lag polynomials of the ARMA part of the ARFIMA model.

Value

an \( R \) function \( f(\lambda) \).

Author(s)

Georg M. Goerg

References


See Also

\( \text{spectrum} \)

Examples

```r
x = spectrum.arfima(d = 0.2)
u = spectrum.arfima(d = -0.2)

plot(x, xlim = c(0.01, pi), ylim = c(0, 1))
plot(u, xlim = c(0.01, pi), add = TRUE, lty = 2)
```

---

**spectrum.arma**

**Theoretical spectrum of an ARMA model**

Description

Computes the spectrum for an ARMA model. Adapted code by David Stoffer (Shumway R. & Stoffer D., 2006).
Usage

spectrum.arma(ar = 0, ma = 0, sd.innov = 1)

Arguments

ar      AR parameters
ma      MA parameters
sd.innov standard deviation of input white noise

Details

The spectrum (spectral density) of a covariance stationary \( Y_t \sim \text{ARMA}(p, q) \) process equals

\[
    f_Y(\lambda) = \sigma^2 \left| \Theta(e^{i\lambda}) \right|^2 \left| \Phi(e^{i\lambda}) \right|^2
\]

where \( \Phi(z) \) and \( \Theta(z) \) are the lag polynomials of the ARMA model.

Value

an R function \( f(\lambda) \).

Author(s)

Georg M. Goerg

References


See Also

spectrum

Examples

```r
x = spectrum.arma(ar = c(0.2, 0.7), ma = -0.5)
u = spectrum.arma(ar = c(0, 0.7), ma = -0.5)
plot(x, xlim = c(0, pi), ylim = c(0, 1))
plot(u, xlim = c(0, pi), add = TRUE, lty = 2)
```
**TreeRing**

**Tree Rings Time Series Data**

**Description**
Width measurements Time Series Data corresponding to Campito Tree Rings.

**Usage**
data(TreeRing)

**Format**
TreeRing is a data frame with 1164 observations.

**Source**

**References**


**Examples**
data(TreeRing)
str(TreeRing)
plot(TreeRing)

\[ y = \text{TreeRing} - \text{mean(TreeRing)} \]
plot(y,type="l",xlab="Time (annual)",ylab="width measurements")
acf(y,lag=100,main="")

---

**var.afm**

**Parameter Variance-Covariance matrix for ARFIMA models**

**Description**
Method to estimate the Parameter Variance-Covariance matrix associated to ARFIMA \((p, d, q)\) process. Correspond to a generalization of the illustrated method for ARFIMA \((1, d, 1)\) model by Palma (2007). This method, is based in the explicit formula obtained by the derivatives of the parameters log-likelihood gradients.
Usage

\texttt{var.afm(\phi = 0, \theta = 0, n)}

Arguments

\texttt{phi} \\
\hspace{1em} a numeric vector containing the autoregressive polynomial parameters. By default, is 0.

\texttt{theta} \\
\hspace{1em} a numeric vector containing the mobile average polynomial parameters. By default, is 0.

\texttt{n} \\
\hspace{1em} sample total size.

Details

We considering the distribution convergence $\sqrt{n}(\hat{\alpha} - \alpha) \to N[0, \Gamma^{-1}(\alpha)]$ so that $s.d(\hat{\alpha}_i) = \sqrt{\frac{1}{n}\Gamma^{-1}_{ii}}$ with

$$
\Gamma = \frac{1}{4\pi} \int_{-\pi}^{\pi} [\nabla \log f(\lambda)][\nabla \log f(\lambda)]' d\lambda
$$

where the gradients are obtained by the exact derivatives (See Contreras-Reyes J. & Palma W., 2011). Finally, the integrals are obtained by \texttt{integrate} function.

Value

\texttt{G} \\
\hspace{1em} the exact Variance-Covariance matrix.

\texttt{SD} \\
\hspace{1em} a vector with standard deviations associated to d, phi_i and theta_j parameters, with i=1,...,p and j=1,...,q.

Author(s)

Javier E. Contreras-Reyes, Wilfredo Palma

References


See Also

\texttt{arfima.whittle, summary.arfima, integrate}
Examples

data(MammothCreek)
y=MammothCreek-mean(MammothCreek)
S=arfima.whittle(y,nar=1,nma=1)
V=var.afm(phi=S$ar,theta=S$ma,n=length(y))
V$SD
summary(S)

# ARFIMA(1,d,1) particular case: (Palma, 2007; page 106 & 108).

phi1=-0.5
theta1=0.2
n1 = 100

var.arfima1d1 <-
  function(phi,theta,n)
  {
    V <- matrix(NA,3,3)
    V[1,1]=pi^2/6
    V[2,2]=1/(1-phi^2)
    V[3,3]=1/(1-theta^2)
    V[1,2]=V[2,1]=-log(1+phi)/phi
    V[1,3]=V[3,1]=log(1+theta)/theta
    V[2,3]=V[3,2]=-1/(1-phi*theta)
    V1=(solve(V))/n
    return(V1)
  }

var.arfima1d1(phi = -phi1, theta = theta1, n = n1)
# note the conversion by definition: phi = -phi1

var.afm(phi = phi1, theta = theta1, n = n1)$G
Index

*Topic **arfima**
  afmtools-package, 2
  arfima-methods, 4
  arfima.whittle, 5
  arfima.whittle.loglik, 7
  check.parameters.arfima, 9
  ir.arfima, 12
  per.arfima, 14
  pi.j, 15
  psi.j, 17
  rho.sowell, 19
  smv.afm, 21
  spectrum.arfima, 22
  var.afm, 25

*Topic **arma**
  rho arma, 18
  spectrum arma, 23

*Topic **asymptotic**
  ir.arfima, 12
  pi.j, 15
  psi.j, 17

*Topic **autocovariance**
  rho arma, 18
  rho.sowell, 19

*Topic **datasets**
  MammothCreek, 13
  TreeRing, 25

*Topic **diagnostic**
  check.parameters.arfima, 9

*Topic **estimation**
  arfima.whittle, 5
  arfima.whittle.loglik, 7
  var.afm, 25

*Topic **forecasting**
  gw.test, 10
  pi.j, 15
  psi.j, 17

*Topic **loglikelihood**
  arfima.whittle.loglik, 7

  per.arfima, 14

*Topic **package**
  afmtools-package, 2

*Topic **spectrum**
  spectrum.arfima, 22
  spectrum arma, 23

*Topic **test**
  gw.test, 10

*Topic **variance**
  smv afm, 21
  var afm, 25

acf, 19, 21
afmtools (afmtools-package), 2
arfima-methods, 4
arfima.whittle, 2, 4, 5, 13, 26
arfima.whittle.loglik, 2, 7
arima, 4, 6

Box.test, 5

check.parameters.arfima, 9
gw.test, 2, 10
integrate, 26
ir.arfima, 12
MammothCreek, 13

per.arfima, 14
pi.j, 15, 18
plot arfima (arfima-methods), 4
predict, 16, 18
print arfima (arfima-methods), 4
psi.j, 13, 16, 17
residuals arfima (arfima-methods), 4
rho arma, 18
rho.sowell, 2, 19, 19, 22
INDEX

smv.afm, 21
spectrum, \textit{15, 23}
spectrum.arfima, \textit{2, 15, 22}
spectrum arma, \textit{19–22, 23}
summary.arfima, 26
summary.arfima (arfima-methods), 4

TreeRing, 25
tsdiag.arfima (arfima-methods), 4

var.afm, 2, 25