# Package ‘FitAR’

June 21, 2007

**Title** AR and Subset AR Modelling

**Version** 1.0

**Date** 2006-11-06

**Author** A.I. McLeod and Ying Zhang

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

**Depends** R (>= 2.0.0), lattice, leaps

**Description** Complete functions are given for model identification, estimation and diagnostic checking for AR and subset AR models. Two types of subset AR models are supported. One family of subset AR models, denoted by ARp, is formed by taking subet of the original AR coefficients and in the other, denoted by ARz, subsets of the partial autocorrelations are used. The main advantage of the ARp model is its applicability to very large order models. For the nonsubset, AR(p), case, the function given in this package outperforms the buit-in R function "ar" for exact mle when p is large. In addition to speed, another advantage of this package is that it is entirely written in the R language which makes it easier to port to other systems such as Mathematica or MatLab.



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

## R topics documented:

---

`AR1Est` *Exact MLE Mean-Zero AR(1)*

---

### Description

This function is used by GetFitAR in the AR(1) case. It is a fast exact solution using the root of a cubic equation.

### Usage

```
AR1Est(z, MeanValue = 0)
```

### Arguments

| | |
|---|---|
| `z` | time series or vector |
| `MeanValue` | known mean |

### Details

The exact MLE for mean-zero AR(1) satisfies a cubic equation. The solution of this equation for the MLE given by Ying (2002) is used. This approach is more reliable as well as faster than the usual approach to the exact MLE using a numerical optimization technique which can occasionally have convergence problems.

### Value

MLE for the parameter

### Author(s)

A.I. McLeod and Ying Zhang

### References

Zhang, Y. (2002). Topics in Autoregression, Ph.D. Thesis, University of Western Ontario.

### See Also

`GetFitAR`

### Examples

```
AR1Est(lynx-mean(lynx))
```

ARSdf *Autoregressive Spectral Density Function*

## Description

Spectral density function is computed.

## Usage

```
ARSdf(phi, pFFT = 8)
```

## Arguments

| | |
|---|---|
| phi | AR Coefficients |
| pFFT | FFT with $2^pFFF$ frequencies, default 8 |

## Details

The Fast Fourier Transform (FFT) is used to compute the spectral density function.

## Value

A vector of the density function values, $(f(1), ..., f(2^pFFF))$

## Author(s)

A.I. McLeod

## See Also

spectrum, spec.pgram, spec.ar

## Examples

```
ARSdf(0.8)
```

ARToMA *Coefficients in Infinite Moving Average Expansion*

## Description

A stationary-causal AR(p) can be written as a general linear process (GLP). This function obtains the moving-average expansion out to the L-th lag, z[t] = a[t]+psi[1]*a[t-1]+...+psi[L]*a[t-L].

## Usage

```
ARToMA(phi, lag.max)
```

### Arguments

| | |
|---|---|
| `phi` | AR Coefficients |
| `lag.max` | maximum lag |

### Details

The coefficients are computed recursively as indicated in Box and Jenkins (1970).

### Value

vector of length L+1 containing, (1,psi[1],...,psi[L])

### Author(s)

A.I. McLeod

### References

Box and Jenkins (1970), Time Series Analysis, Forecasting & Control

### See Also

`InvertibleQ`

### Examples

```
ARToMA(0.5,20)
```

---

`ARToPacf` *Reparametrize AR coefficients in Terms of PACF*

---

### Description

Transform AR parameter coefficients into partial autocorrelation function (PACF).

### Usage

```
ARToPacf(phi)
```

### Arguments

| | |
|---|---|
| `phi` | vector of AR parameter coefficients |

### Details

For details see McLeod and Zhang (2006).

### Value

Vector of length(phi) containing the parameters in the transformed PACF domain

### Warning

No check for invertibility is done for maximum computational efficiency since this function is used extensively in the numerical optimization of the AR loglikelihood function in FitAR. Use InvertibleQ to test for invertible AR coefficients.

### Author(s)

A.I. McLeod

### References

McLeod, A.I. and Zhang, Y. (2006). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

### See Also

InvertibleQ, PacfToAR

### Examples

```
somePACF<-c(0.5,0.6,0.7,0.8,-0.9,-0.8)
#PacfToAR() transforms PACF to AR parameter coefficients.
someAR<-PacfToAR(somePACF)
test<-ARToPacf(someAR)
#This should be very small
sum(abs(test-somePACF))
```

---

## AcfPlot *Autocorrelation Plot*

---

### Description

Produces correlation plot

### Usage

```
AcfPlot(g, LagZeroQ= TRUE, ylab=NULL, main=NULL, ...)
```

### Arguments

| | |
|---|---|
| g | vector of autocorrelations at lags 1,..,length(g) |
| LagZeroQ | start plot at lag zero with g[0]=1 |
| ylab | vertical axis label |
| main | plot title |
| ... | optional graphical parameters |

### Value

No value. Plot is produced via plot function.

### Author(s)

A.I. McLeod

### See Also

acf

### Examples

```
#
#simple example, plot acf for AR(1)
 phi<-0.8
 maxLag<-20
 g<-phi^(1:maxLag)
 AcfPlot(g)
 AcfPlot(g, LagZeroQ=FALSE)
#
# Plot the sample inverse partial autocorrelations.
# On the basis of this plot, Cleveland (1972) suggested an ARp(1,2,7)
# for this data
"InverseAcf" <-
function(z, p=15){
g<-TacvfMA(GetFitAR(z-mean(z),1:p)$phiHat, lag.max=p)
g/g[1]
}
#
data(SeriesA)
AcfPlot(InverseAcf(SeriesA),LagZeroQ=FALSE)
```

---

## BackcastResidualsAR

*Innovation Residuals in AR*

---

### Description

Obtains the residuals (estimated innovations). The residuals for t=1,...,p are obtained using the backforecasting algorithm of Box and Jenkins (1970).

### Usage

```
BackcastResidualsAR(y, phi, Q = 100, demean=TRUE)
```

### Arguments

| | |
|---|---|
| y | a time series or vector |
| phi | AR coefficients, lags 1,...,p |
| Q | for backcasting, the AR is approximated by an MA(Q) |
| demean | subtract sample mean |

### Details

The backforecasting algorithm is described in detail in the book of Box and Jenkins. The idea is to expected value of the innovation assuming a high-order MA(q).

## Value

Vector of residuals

## Note

No check is done that the AR is causal-stationary.

## Author(s)

A.I. McLeod

## References

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

## See Also

InvertibleQ, FitAR

## Examples

```
#compare residuals obtained using backcasting with fitted parameters and
# the residuals extracted from output of FitAR.  They are identical.
p<-11
out<-FitAR(log(lynx), p)
phi<-out$phiHat #fitted parameters
resphi<-BackcastResidualsAR(log(lynx), phi)
sum(abs(resphi-resid(out)))
```

---

# Boot.FitAR *Simulate a Fitted AR*

---

## 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 'FitAR':
Boot(obj, R=1, ...)
```

## Arguments

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

## Value

A simulated time series with the same length as the original fitted time series is produced.

## Author(s)

A.I. McLeod

## See Also

Boot SimulateGaussianAR

## Examples

```
#Plot log(lynx) time series and simulation
#
data(lynx)
ans <- FitAR(log(lynx), 8)
z<-Boot.FitAR(ans)
par(mfrow=c(2,1))
TimeSeriesPlot(log(lynx))
title(main="log(lynx) time series")
TimeSeriesPlot(z)
title(main="Simulated AR(8), fitted to log lynx")
#par(mfrow=c(1,1)
#
#Use bootstrap to compute standard errors of parameters
#takes about 18 seconds on a 3.6 GHz PC
ptm <- proc.time() #user time
R<-100  #number of bootstrap iterations
p<-c(1,2,4,7,10,11)
ans<-FitAR(log(lynx),p)
out<-Boot(ans, R)
fn<-function(z) FitAR(z,p)$zetaHat
sdBoot<-sqrt(diag(var(t(apply(out,fn,MARGIN=2)))))
sdLargeSample<-coef(ans)[,2][1:6]
sd<-matrix(c(sdBoot,sdLargeSample),ncol=2)
dimnames(sd)<-list(names(sdLargeSample),c("Bootstrap","LargeSample"))
ptm<-(proc.time()-ptm)[1]
sd
```

---

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 |

### Details

At present, the only function implemented is Boot.FitAR.

### Value

Parametric bootstrap simulation

### Author(s)

A.I. McLeod

### See Also

Boot.FitAR

### Examples

```
data(SeriesA)
out<-FitARLS(SeriesA, c(1,2,7))
Boot(out)
```

---

## Boot.ts *Parametric Time Series Bootstrap*

---

### Description

An AR(p) model is fit to the time series using the AIC and then it is simulated.

### Usage

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

### Arguments

| | |
|---|---|
| obj | a time series, class "ts" |
| R | number of bootstrap replicates |
| ... | optional arguments |

### Value

A time series or vector.

### Note

Parametric and nonparametric time series bootstraps are discussed by Davison and Hinkley (1997, Ch.8.2).

### Author(s)

A.I. McLeod

### References

Davison, A.C. and Hinkley, D.V. (1997), Bootstrap Methods and Their Application. Cambridge University Press.

### See Also

Boot.FitAR. Nonparametric bootstrap for time series is available in the function `tsboot` in the library `boot`.

### Examples

```
data(SeriesA)
layout(matrix(c(1,2,1,2),ncol=2))
TimeSeriesPlot(SeriesA)
TimeSeriesPlot(Boot(SeriesA),main="Bootstrap of Series A")
```

---

## BoxCox.Arima — *Box-Cox Analysis for "Arima" Objects*

---

### Description

Implements Box-Cox analysis for "Arima" class objects, the output from `arima`. Variance change in time series is an important topic. In some cases using a Box-Cox transformation will provide a much simpler analysis than the much more complex ARMA-GARCH approach. See US Tobacco series example given below for an example.

### Usage

```
## S3 method for class 'Arima':
BoxCox(object, interval = c(-1, 1), type = "BoxCox", InitLambda = "none", ...)
```

### Arguments

| | |
|---|---|
| `object` | output from arima |
| `interval` | interval to be searched for optimal transformation |
| `type` | Ignored unless, InitLambda!="none". Type of transformation, default is "BoxCox". Otherwise a simple power transformation. |
| `InitLambda` | default "none". Otherwise a numerical value giving the transformation parameter. |
| `...` | optional arguments passed to optimize |

### Details

If no transformation is used on the data, then the original data is used. But if a transformation has already been used, we need to inverse transform the data to recover the untransformed data.

For $\lambda \neq 0$, the Box-Cox transformation is of x is $(x-1)^{\lambda}/\lambda$ whereas the regular power transformation is simply $x^{\lambda}$. When $\lambda = 0$, it is log in both cases.

The log of the Jacobian is $(\lambda - 1)\sum_{t=D+1}^{n} \log(z_t)$ (lambda-1)*sum(log(z[(D+1):n]), where $\lambda$ is the transformation, n=length(z), z is the vector of data and D = d + ds*s, where d is the degree of regular differencing, ds is the degree of seasonal differencing and s is the seasonal period. The correct expression for the loglikelihood function was first given in Hipel and McLeod (1977, eqn. 10). Using the wrong expression for the Jacobian has a disasterous effect in many situations. For example with the international airline passenger time series, the MLE for lambda would be about 1.958 instead of close to zero.

If the minimum data value is <= 0, a small positive constant, equal to the negative of the minimum plus 0.25, is added to all the data values.

## Value

No value returned. Graphical output produced as side-effect. The plot shows relative likelihood function as well as the MLE and a confidence interval.

## Note

The MASS package has a similar function `boxcox` but this is implemented only for regression and analysis of variance.

## Author(s)

A.I. McLeod

## References

Hipel, K.W. and McLeod, A.I. (1977). Advances in Box-Jenkins Modelling. Part 1, Model Construction. Water Resources Research 13, 567-575.

## See Also

`arima`, `BoxCox`, `BoxCox.FitAR`

## Examples

```
#Tobacco Production
 data(USTobacco)
 plot(USTobacco)
 win.graph()
 USTobacco.arima<-arima(USTobacco,order=c(0,1,1))
 BoxCox(USTobacco.arima)
#
 air.arima<-arima(AirPassengers, c(0,1,1), seasonal=list(order=c(0,1,1), period=12))
 BoxCox(air.arima)
#
#In this example, we fit a model to the square-root of the sunspots and
#back transform in BoxCox.
sqrtsun.arima<-arima(sqrt(sunspot.year),c(2,0,0))
BoxCox(sqrtsun.arima, InitLambda=0.5, type="power")
#
#Back transform with AirPassengers
Garima<-arima(log(AirPassengers), c(0,1,1), seasonal=list(order=c(0,1,1),period=12))
BoxCox(Garima, InitLambda=0)
```

## BoxCox.FitAR — *Box-Cox Analysis for "FitAR" Objects*

### Description

This is a methods function to do a Box-Cox analysis for models fit using FitAR and FitARLS.

### Usage

```
## S3 method for class 'FitAR':
BoxCox(object, interval = c(-1, 1), type = "BoxCox", InitLambda = "none", ...)
```

### Arguments

| | |
|---|---|
| `object` | output from FitAR or FitARLS |
| `interval` | interval to be searched for optimal transformation |
| `type` | Ignored unless, InitLambda!="none". Type of transformation, default is "Box-Cox". Otherwise a simple power transformation. |
| `InitLambda` | default "none". Otherwise a numerical value giving the transformation parameter. |
| `...` | optional arguments passed to optimize |

### Details

If no transformation is used on the data, then the original data is used. But if a transformation has already been used, we need to inverse transform the data to recover the untransformed data.

For $\lambda \neq 0$, the Box-Cox transformation is of x is $(x-1)^{\lambda}/\lambda$ whereas the regular power transformation is simply $x^{\lambda}$. When $\lambda = 0$, it is log in both cases.

If the minimum data value is <= 0, a small positive constant, equal to the negative of the minimum plus 0.25, is added to all the data values.

### Value

No value returned. Graphical output produced as side-effect. The plot shows relative likelihood function as well as the MLE and a confidence interval.

### Note

The MASS package has a similar function `boxcox` but this is implemented only for regression and analysis of variance.

### Author(s)

A.I. McLeod

### References

Box, G. E. P. and Cox, D. R. (1964) An analysis of transformations. Journal of Royal Statistical Society, Series B, vol. 26, pp. 211-246.

McLeod, A.I. and Zhang, Y. (2006a). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

McLeod, A.I. and Zhang, Y. (2006b, under review). Subset Autoregression Modelling. Journal of Statistical Sofware.

### See Also

BoxCox, BoxCox.Arima

### Examples

```
#lynx time series. ARp subset model.
out<-FitARLS(lynx, c(1,2,4,10,11))
BoxCox(out)
#
#sunspots. ARz subset model.
z<-sunspot.year+0.25
p<-SelectModel(z, SubsetModel="z", lag.max=25, Best=1)
out<-FitAR(z, p)
BoxCox(out)
#
#compare with AR(10)
z<-sunspot.year+0.25
out<-FitAR(z, 10)
BoxCox(out)
#
#Back transform after fitting model to log(lynx)
p<-SelectModel(log(lynx),SubsetModel="z",Best=1)
ans<-FitAR(log(lynx), p)
BoxCox(ans, InitLambda=0)
#
#again with ARp subset model
p<-SelectModel(log(lynx),SubsetModel="p",Best=1)
ans<-FitARLS(log(lynx), p)
BoxCox(ans, InitLambda=0)
```

---

## BoxCox *Generic Box-Cox Analysis Function*

---

### Description

Generic function

### Usage

```
BoxCox(object, ...)
```

### Arguments

| | |
|---|---|
| `object` | model object |
| `...` | optional arguments |

### Value

No value returned. Graphical output produced as side-effect. The plot shows relative likelihood function as well as the MLE and a confidence interval.

### Note

The MASS package has a similar function boxcox but this is implemented only for regression and analysis of variance.

### Author(s)

A.I. McLeod

### See Also

`BoxCox.Arima`, `BoxCox.FitAR`, `BoxCox.ts`, `BoxCox.numeric`

### Examples

```
BoxCox(lynx)
out<-FitARLS(lynx, c(1,2,4,10,11))
BoxCox(out)
out<-FitAR(lynx, c(1,2,4,5,7,8,10,11,12))
BoxCox(out)
```

---

## `BoxCox.numeric` *Box-Cox Analysis for a Time Series*

---

### Description

An AR(p) model is selected using AIC and then the best Box-Cox transformation is determined. Requires package FitAR.

### Usage

```
BoxCox.numeric(object, interval = c(-1, 1), IIDQ = FALSE, ...)
```

### Arguments

| | |
|---|---|
| `object` | a vector of time series values |
| `interval` | interval to be searched |
| `IIDQ` | If true, IID is assumed, ie. p=0. If FALSE, AR(p) is fit with p determined using AIC. |
| `...` | optional arguments |

### Details

If the minimum data value is <= 0, a small positive constant, equal to the negative of the minimum plus 0.25, is added to all the data values. If length(object) < 20, no AR model is used, that is, p=0.

### Value

No value returned. Graphical output produced as side-effect. The plot shows relative likelihood funciton as well as the MLE and a confidence interval.

### Note

The MASS package has a similar function boxcox but this is implemented only for regression and analysis of variance.

### Author(s)

A.I. McLeod

### References

Box, G. E. P. and Cox, D. R. (1964) An analysis of transformations. Journal of Royal Statistical Society, Series B, vol. 26, pp. 211-246.

### See Also

BoxCox.FitAR, BoxCox.Arima, BoxCox.ts

### Examples

```
#
#annual sunspot series
BoxCox(sunspot.year, IIDQ=FALSE)
#
#non-time series example, lengths of rivers
BoxCox(rivers)
```

---

## BoxCox.ts *Box-Cox Analysis for a Time Series*

---

### Description

The time series is converted to a vector and BoxCox.numeric is used.

### Usage

```
BoxCox.ts(object, interval = c(-1, 1), ...)
```

### Arguments

| | |
|---|---|
| object | a vector of time series values |
| interval | interval to be searched |
| ... | optional arguments |

## Details

If the minimum data value is <= 0, a small positive constant, equal to the negative of the minimum plus 0.25, is added to all the data values.

## Value

No value returned. Graphical output produced as side-effect. The plot shows relative likelihood function as well as the MLE and a confidence interval.

## Warning

It is important not to transform the data when fitting it with AR since the optimal transformation would be found for the transformed data – not the original data. Normally this would not be a sensible thing to do.

## Note

The MASS package has a similar function boxcox but this is implemented only for regression and analysis of variance.

## Author(s)

A.I. McLeod

## References

Box, G. E. P. and Cox, D. R. (1964) An analysis of transformations. Journal of Royal Statistical Society, Series B, vol. 26, pp. 211-246.

## See Also

BoxCox.FitAR, BoxCox.Arima, BoxCox.numeric

## Examples

```
#
BoxCox(sunspot.year)
```

---

# ChampernowneD *Champernowne Matrix*

---

## Description

Computes sufficient statistics for AR

## Usage

```
ChampernowneD(z, p, MeanZero = FALSE)
```

## Arguments

| | |
|---|---|
| `z` | time series data |
| `p` | order of the AR |
| `MeanZero` | Assume mean is zero. Default is FALSE so the sample mean is subtracted from the data first. Otherwise no sample mean correction is made. |

## Details

This matrix is defined in McLeod & Zhang (2006)

## Value

The matrix D defined following eqn. (3) of McLeod & Zhang (2006) is computed.

## Note

This function is used by GetFitAR. It may be used to compute the exact loglikelihood for an AR.

## Author(s)

A.I. McLeod

## References

McLeod, A.I. and Zhang, Y. (2006). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

## See Also

`GetFitAR`, `FastLoglikelihoodAR`, `FitAR`

## Examples

```
#compute the exact concentrated loglikelihood function, (McLeod & Zhang, 2006, eq.(6)),
# for AR(p) fitted by Yule-Walker to logged lynx data
#
p<-8
CD<-ChampernowneD(log(lynx), p)
n<-length(lynx)
phi<-ar(log(lynx), order.max=p, aic=FALSE, method="yule-walker")$ar
LoglYW<-FastLoglikelihoodAR(phi,n,CD)
phi<-ar(log(lynx), order.max=p, aic=FALSE, method="burg")$ar
LoglBurg<-FastLoglikelihoodAR(phi,n,CD)
phi<-ar(log(lynx), order.max=p, aic=FALSE, method="ols")$ar
LoglOLS<-FastLoglikelihoodAR(phi,n,CD)
phi<-ar(log(lynx), order.max=p, aic=FALSE, method="mle")$ar
LoglMLE<-FastLoglikelihoodAR(phi,n,CD)
ans<-c(LoglYW,LoglBurg,LoglOLS,LoglMLE)
names(ans)<-c("YW","Burg","OLS","MLE")
ans
#compare the MLE result given by ar with that given by FitAR
FitAR(log(lynx),p)
```

`DetAR` *Covariance Determinant of AR(p)*

## Description

Computes the covariance determinant of p successive observations from an AR(p) process with unit innovation variance.

## Usage

```
DetAR(phi)
```

## Arguments

| | |
|---|---|
| `phi` | vector of AR coefficients |

## Details

The AR coefficients are transformed to PACF and then the determinant is computed as a product of PACF terms as given in McLeod and Zhang (2006, eqn. 4).

## Value

Determinant

## Author(s)

A.I. McLeod

## References

McLeod, A.I. and Zhang, Y. (2006). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

## See Also

`FastLoglikelihoodAR`

## Examples

```
DetAR(c(0.1,0.1,0.1))
```

FastLoglikelihoodAR

*Fast Computation of the Loglikelihood Function in AR*

## Description

Computation of the loglikelihood is O(1) flops in repeated evaluations of the loglikelihood holding the data fixed and varying the parameters. This is useful in exact MLE estimation.

## Usage

```
FastLoglikelihoodAR(phi, n, CD)
```

## Arguments

| | |
|---|---|
| phi | AR coefficients |
| n | length of series |
| CD | Champernowne matrix |

## Details

The details of this computation are described in McLeod and Zhang (2006).

## Value

loglikelihood

## Author(s)

A.I. McLeod

## References

McLeod, A.I. and Zhang, Y. (2006). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

## See Also

ChampernowneD, LoglikelihoodAR

## Examples

```
#Compute the loglikelihood using the direct method as implemented
# in LoglikelihoodAR and using the fast method
data(SeriesA)
phi<-PacfToAR(rep(0.5,10))
p<-length(phi)
z<-SeriesA-mean(SeriesA)
n<-length(z)
L1<-LoglikelihoodAR(phi, z)
cd<-ChampernowneD(z,p,MeanZero=TRUE)
L2<-FastLoglikelihoodAR(phi,n,cd)
```

```
out<-c(L1,L2)
names(out)<-c("direct","fast")
out
```

FitAR *Exact MLE for AR(p) and Subset AR*

## Description

The estimation algorithm in McLeod & Zhang (2006a) is implemented.

## Usage

```
FitAR(z, p, SubsetQ=FALSE,  demean = TRUE, MeanMLEQ = FALSE, lag.max = "default"
```

## Arguments

| | |
|---|---|
| `z` | time series, vector or ts object. |
| `p` | p specifies the model. If length(p) is 1, an AR(p) is assumed and if p has length greater than 1, an ARz subset model with lags indicated by p is assumed. If you wish to fit a subset model with only one lag, say for example, 4, then set p to be c(0,0,0,4). For p=0, white noise model. |
| `SubsetQ` | default FALSE. Ignored if p has length > 1. Need set to TRUE for ambiguous cases – see Note. |
| `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 MLE for AR(p) and subset AR(p) are using methods described in McLeod and Zhang (2006a). In addition the exact MLE for the mean can be computed using an iterative backfitting approach described in McLeod and Zhang (2006b).

The default for lag.max is ceiling(min(length(z)/4, min(max(length(z)/4, 30), 100))).

## Value

A list with class name "FitAR" and components:

| | |
|---|---|
| `loglikelihood` | value of the loglikelihood |
| `phiHat` | coefficients in AR(p) – including 0's |
| `sigsqHat` | innovation variance estimate |
| `muHat` | estimate of the mean |
| `covHat` | covariance matrix of the coefficient estimates |
| `zetaHat` | transformed parameters, length(zetaHat)=# coefficients estimated |

| | |
|---|---|
| `RacfMatrix` | residual autocorrelations and sd for lags 1...lag.max |
| `LjungBox` | table of Ljung-Box portmanteau test statistics |
| `SubsetQ` | parameters in AR(p) – including 0's |
| `res` | innovation residuals, same length as z |
| `fits` | fitted values, same length as z |
| `lags` | lags used in AR model |
| `demean` | TRUE if mean estimated otherwise assumed zero |
| `FitMethod` | "MLE" for this function |
| `IterationCount` | number of iterations in mean mle estimation |
| `convergence` | value returned by optim – should be 0 |
| `MLEMeanQ` | TRUE if mle for mean algorithm used |
| `tsp` | tsp(z) |
| `call` | result from match.call() showing how the function was called |
| `ModelTitle` | description of model |
| `DataTitle` | returns attr(z,"title") |

## Note

The SubsetQ parameter is used to distinguish models such as seasonal lag models from full AR models. For large p, this algorithm is faster than the built-in R function. See example below. There are generic print, summary, coef and resid functions for class "FitAR".

## Author(s)

A.I. McLeod

## References

McLeod, A.I. and Zhang, Y. (2006a). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

McLeod, A.I. and Zhang, Y. (2006b, Accepted). Faster ARMA Maximum Likelihood Estimation, Computational Statistics and Data Analysis.

## See Also

`FitARLS`, `RacfPlot`

## Examples

```
#First Example: Fit exact MLE to AR(4) using FitAR and arima
set.seed(3323)
phi<-c(2.7607,-3.8106,2.6535,-0.9238)
z<-SimulateGaussianAR(phi,1000)
ans1<-arima(z, order=c(4,0,0))
ans2<-FitAR(z,4,MeanMLEQ=TRUE)
ans1
coef(ans2)

#Second Example: compare with sample mean result
```

```
ans3<-FitAR(z,4)
coef(ans3)

#Third Example: Fit subset AR model to lynx series
z<-log(lynx)
FitAR(z, c(1,2,4,7,10,11))
#now obain exact MLE for Mean as well
FitAR(z, c(1,2,4,7,10,11), MeanMLE=TRUE)

#Fourth Example: Some Timings
t1<-system.time(ans1<-ar(sunspot.month, order.max=25, aic=FALSE, method="mle"))[[1]]
t2<-system.time(ans2<-FitAR(sunspot.month, 25))[[1]]
#uncomment lines below to run full examples but it takes about
# 4 minutes on a 3.6GHz PC
#
#t3<-system.time(ans3<-FitAR(sunspot.month, 25, MeanMLE=TRUE))[[1]]
#times<-c(t1,t2,t3)
#names(times)<-c("ar","FitAR", "FitAR-MeanMLE")
```

---

`FitARLS` *Fit Subset AR Using OLS*

---

## Description

The usual subset AR model is defined by selecting a subset of the AR coefficients and setting all other coefficients to zero. This subset model is fit using least squares.

## Usage

```
FitARLS(z, p, SubsetQ = FALSE, lag.max = "default")
```

## Arguments

| | |
|---|---|
| `z` | time series |
| `p` | Vector specifying which lags to use in the AR. If SubsetQ=FALSE, p should be a scalar and a regular AR(p) will be fit. For p=0, white noise. |
| `SubsetQ` | default FALSE. Ignored if p has length > 1. Need set to TRUE for ambiguous cases – see Note. |
| `lag.max` | The maximum lag to be used in the Portmanteau Test |

## Details

The design matrix, X, is formed by concatenating together the needed columns and then the R function lsfit is used. An intercept term is used with lsfit. The residuals are computed using BackcastResidualsAR. The exact loglikelihood is computed using LoglikelihoodAR.

## Value

A list with class name "FitAR" and components:

| | |
|---|---|
| `loglikelihood` | value of the loglikelihood using LoglikelihoodAR |
| `phiHat` | coefficients in AR(p) – including 0's |
| `sigsqHat` | innovation variance estimate |
| `muHat` | estimate of the mean |
| `covHat` | covariance matrix of the coefficient estimates |
| `zetaHat` | transformed parameters, length(zetaHat)=# coefficients estimated |
| `RacfMatrix` | residual autocorrelations and sd for lags 1...lag.max |
| `LjungBox` | table of Ljung-Box portmanteau test statistics |
| `SubsetQ` | parameters in AR(p) – including 0's |
| `res` | innovation residuals, same length as z |
| `fits` | fitted values, same length as z |
| `lags` | lags used in AR model |
| `demean` | TRUE if mean estimated otherwise assumed zero |
| `FitMethod` | "LS" for this function |
| `tsp` | tsp(z) |
| `call` | result from match.call() showing how the function was called |
| `yX` | the dependent variable column prepended to the columns of independent variables |
| `ModelTitle` | description of model |
| `DataTitle` | returns attr(z,"title") |

## Note

If SubsetQ=FALSE, this is equivalent to the built-in function ar( ..., method="OLS"). Note that least-squares residuals are not used. The residuals are calculated using BackcastResidualsAR and the loglikelihood is calculated using LoglikelihoodAR.

## Author(s)

A.I. McLeod

## References

Tong, H. (1977). Some comments on the Canadian lynx data. Journal of the Royal Statistical Society A 140, 432-436.

## See Also

`FitAR`, `LoglikelihoodAR`, `BackcastResidualsAR`, `ar`

### Examples

```
#Compare the fit achieved by the two types of subset models
z<-log(lynx)
pvec<-SelectModel(z, SubsetModel="z", Criterion="BIC", lag.max=12, Best=1)
pvec
FitARLS(z, pvec)
FitAR(z, pvec)
```

---

## FromSymmetricStorageUpper *Converts a Matrix from Symmetric Storage Mode to Regular Format*

---

### Description

Utility function.

### Usage

```
FromSymmetricStorageUpper(x)
```

### Arguments

| | |
|---|---|
| x | a vector which represents a matrix in upper triangular form |

### Value

symmetric matrix

### Author(s)

A.I. McLeod

### Examples

```
FromSymmetricStorageUpper(1:5)
```

---

## Get1G *Internal Utility Function: BLUE mean*

---

### Description

This function is not normally used directly by the user. It is used in the exact mle for mean.

### Usage

```
Get1G(phi, n)
```

### Arguments

| | |
|---|---|
| phi | vector of AR coefficients |
| n | length of series |

### Details

### Value

A vector used in the mle computation of the mean

### Author(s)

A.I. McLeod

### See Also

GetARMeanMLE

### Examples

```
#Simulate an AR(2) and compute the exact mle for mean
set.seed(7771111)
n<-50
phi<-c(1.8,-0.9)
z<-SimulateGaussianAR(phi, n)
g1<-Get1G(phi, length(z))
sum(g1*z)/sum(g1)
#sample mean
mean(z)
#more directly with getArMu
GetARMeanMLE(z,phi)
```

---

## GetARMeanMLE *Exact MLE for Mean in AR(p)*

---

### Description

Details of this algorithm are given in McLeod and Zhang (2007).

### Usage

```
GetARMeanMLE(z, phi)
```

### Arguments

| | |
|---|---|
| z | vector of length n containing the time series |
| phi | vector of AR coefficients |

### Value

Estimate of mean

### Author(s)

A.I. McLeod

## References

McLeod, A.I. and Zhang, Y. (2006). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

## See Also

mean

## Examples

```
#Simulate a time series with mean zero and compute the exact
#mle for mean and compare with sample average.
 set.seed(3323)
 phi<-c(2.7607,-3.8106,2.6535,-0.9238)
 z<-SimulateGaussianAR(phi,1000)
 ans1<-mean(z)
 ans2<-GetARMeanMLE(z,phi)
# define a direct MLE function
"DirectGetMeanMLE" <-
function(z, phi){
    GInv<-solve(toeplitz(TacvfAR(z, length(z)-1)))
    g1<-colSums(GInv)
    sum(g1*z)/sum(g1)
}
ans3<-DirectGetMeanMLE(z,phi)
ans<-c(ans1,ans2,ans3)
names(ans)<-c("mean", "GetARMeanMLE","DirectGetMeanMLE")
ans
```

---

# GetB *Internal Utility Function*

---

## Description

The user would not normally use this function. The function is needed for exact mle for mean. Used in Get1G which is called from GetARMeanMLE.

## Usage

```
GetB(phi)
```

## Arguments

| | |
|---|---|
| phi | vector of AR coefficients |

GetFitAR *Fit AR(p)*

## Description

Obtains the exact MLE for AR(p) or subset AR model. This function is used by FitAR. One might prefer to use GetFitAR for applications such as bootstrapping since it is faster than FitAR.

## Usage

```
GetFitAR(z, pvec, MeanValue=0, ...)
```

## Arguments

| | |
|---|---|
| `z` | time series |
| `pvec` | lags included in AR model. If pvec=0, white noise model assumed. |
| `MeanValue` | by default it is assumed the mean of z is 0 |
| `...` | optional arguments passed through to optim |

## Details

The built-in function optim is used to obtain the MLE estimates for an AR or subset AR.

## Value

| | |
|---|---|
| `loglikelihood` | value of maximized loglikelihood |
| `zetaHat` | estimated zeta parameters |
| `phiHat` | estimated phi parameters |
| `convergence` | result from optim |

## Author(s)

A.I. McLeod

## References

McLeod, A.I. and Zhang, Y. (2006). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

## See Also

`FitAR`

## Examples

```
#compare results from GetFitAR and FitAR
z<-log(lynx)
z<-z - mean(z)
GetFitAR(z, c(1,2,8))
out<-FitAR(log(lynx), c(1,2,8))
out
coef(out)
```

GetFitARLS *Fit AR Using Least-Squares*

## Description

For ARp subset models, the least squares estimates are computed. The exact loglikelihood is then determined. The estimated parameters are checked to see if they are in the AR admissible region.

## Usage

```
GetFitARLS(z, pvec)
```

## Arguments

| | |
|---|---|
| `z` | vector or ts object, the time series |
| `pvec` | lags included in subset AR. If pvec=0, white noise assumed. |

## Details

The R function `lsfit` is used.

## Value

a list with components:

| | |
|---|---|
| `loglikeliihood` | the exact loglikelihood |
| `phiHat` | estimated AR parameters |
| `constantTerm` | constant term in the linear regression |
| `lags` | estimated AR parameters |
| `res` | the least squares regression residuals |
| `yX` | the dependent variable column prepended to the columns of independent variables |
| `InvertibleQ` | True, if the estimated parameters are in the AR admissible region. |

## Note

This is a helper function for `FitARLS`. Normally the user would `FitARLS` since this function provides generic print, summary, resid and plot methods but GetFitARLS is sometimes useful in iterative computations like bootstrapping since it is faster.

## Author(s)

A.I. McLeod

## References

McLeod, A.I. and Zhang, Y. (2006). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

## See Also

FitARLS, FitAR, GetFitAR, LoglikelihoodAR SelectModel

## Examples

```
#Fit subset AR using GetFitARLS and FitARLS
data(SeriesA)
ansLS<-GetFitARLS(SeriesA, c(1,2,7))
ans<-FitARLS(SeriesA, c(1,2,7))
ansLS
ans
summary(ans)
summary(resid(ans))
plot(ans)
```

---

`GetKappa` *Internal Utility Function*

---

## Description

Used by Get1G.

## Usage

```
GetKappa(phi)
```

## Arguments

| | |
|---|---|
| `phi` | ARCoefficients |

---

`GetLeapsAR` *Select lags for Best Subset ARp Model*

---

## Description

The subset ARp model is the usual subset model, for example see Tong (1977). This function is used by SelectModel for model identification for ARp models.

## Usage

```
GetLeapsAR(z, lag.max = 15, Criterion = "UBIC", Best = 3, Candidates=5)
```

## Arguments

| | |
|---|---|
| `z` | ts object or vector containing time series |
| `lag.max` | maximum order of the AR |
| `Criterion` | default UBIC, other choices are "AIC" or "BIC" |
| `Best` | the number of based selected based on highest AIC/BIC |
| `Candidates` | number of models initially selected using the approximate criterion |

## Details

The R function leaps in the R package leaps is used to compute the subset regression model with the smallest residual sum of squares containing 1,...,lag.max parameters. The mean is always included, so the only parameters considered are the phi coefficients. After the best models containing 1,..,lag.max parameters are selected the models are individually refit using GetFitARLS to determine the exact likelihood function for each selected model. Based on this likelihood the UBIC/BIC/AIC is computed and then the best models are selected. The UBIC criterion was developed by Chen and Chen (2007).

## Value

a list with components

`NumParameters`

`UBIC`

`AIC`

`BIC`

`p`

lags present

## Warning

AIC and BIC values produced are not comparable to AIC and BIC produced by SelectModel for ARz models. However comparable AIC/BIC values are produced when the selected models are fit by FitARLS or FitAR respectively.

## Note

Requires leaps package

## Author(s)

A.I. McLeod

## References

Tong, H. (1977) Some comments on the Canadian lynx data. Journal of the Royal Statistical Society A 140, 432-436.

Chen, J. and Chen, Z. (2007). Extended Bayesian Information Criteria for Model Selection with Large Model Space. Preprint.

## See Also

`SelectModel`, `GetFitARLS`, `leaps`

### Examples

```
#for the log(lynx) Tong (1977) selected an ARp(1,2,4,10,11)
#using the AIC and a subset selection algorithm. Our more exact
#approach shows that the ARp(1,2,3,4,10,11) has slightly lower
#AIC (using exact likelihood evaluation).
z<-log(lynx)
GetLeapsAR(z, lag.max=11)
GetLeapsAR(z, lag.max=11, Criterion="BIC")
```

---

## InformationMatrixAR *Information Matrix for AR(p)*

---

### Description

The Fisher large-sample information matrix per observation for the p coefficients in an AR(p) is computed.

### Usage

```
InformationMatrixAR(phi)
```

### Arguments

| | |
|---|---|
| phi | vector of length p corresponding to the AR(p) coefficients |

### Details

The Fisher information matrix is computed as the covariance matrix of an AR(p) process with coefficients given in the argument `phi` and with unit innovation variance. The `TacvfAR` function is used to compute the necessary autocovariances. `FitAR` uses `InformationMatrixAR` to obtain estimates of the standard errors for the estimated parameters in the case of the full AR(p) model.

### Value

a p-by-p Toeplitz matrix, p=length(phi)

### Author(s)

A.I. McLeod

### References

McLeod, A.I. and Zhang, Y. (2006). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

### See Also

`FitAR`, `InformationMatrixARp`, `TacvfAR`, `InformationMatrixARz`

### Examples

```
InformationMatrixAR(c(1.8,-0.6))
```

`InformationMatrixARp` *Information Matrix for ARp*

## Description

The large-sample information matrix per observation is computed in a subset AR with the usual parameterization, that is, a subset of the AR coefficients.

## Usage

```
InformationMatrixARp(phi, lags)
```

## Arguments

| | |
|---|---|
| `phi` | vector of coefficients in the subset AR |
| `lags` | vector indicating lags present in `phi` |

## Details

The subset information matrix is obtained simply by selecting the appropriate rows and columns from the full information matrix. This function is used by `FitARLS` to obtain the estimated standard errors of the parameter estimates.

## Value

a p-by-p Toeplitz matrix, p=length(phi)

## Author(s)

A.I. McLeod

## References

McLeod, A.I. and Zhang, Y. (2006). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

## See Also

`InformationMatrixAR`, `FitARLS`, `InformationMatrixARz`

## Examples

```
#variances of parameters in a subset ARp(1,2,6)
fi<-InformationMatrixARp(c(0.36,0.23,0.23),c(1,2,6))
sqrt(diag(solve(fi*197)))
```

InformationMatrixARz
*Information Matrix ARz*

### Description

Computes the large-sample Fisher information matrix per observation for the AR coefficients in a subset AR when parameterized by the partial autocorrelations.

### Usage

```
InformationMatrixARz(zeta, lags)
```

### Arguments

| | |
|---|---|
| `zeta` | vector of coefficients, ie. partial autocorrelations at lags specified in the argument `lags` |
| `lags` | lags in subset model, same length as zeta argument |

### Details

The details of the computation are given in McLeod and Zhang (2006, eqn 13). `FitAR` uses `InformationMatrixARz` to obtain estimates of the standard errors of the estimated parameters in the subset AR model when partial autocorrelation parameterization is used.

### Value

a p-by-p Toeplitz matrix, p=length(zeta)

### Author(s)

A.I. McLeod

### References

McLeod, A.I. and Zhang, Y. (2006). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

### See Also

`FitAR`, `InformationMatrixAR`, `InformationMatrixARp`

### Examples

```
#Information matrix for ARz(1,4) with parameters 0.9 and 0.9.
InformationMatrixARz(c(0.9, 0.9), lags=c(1,4))
```

`InvertibleQ` *Invertibility Test*

## Description

Tests if the polynomial

$$1 - \phi(1)B \ldots - \phi(p)B^p,$$

where p=length[phi] has all roots outside the unit circle. This is the invertibility condition for the polynomial.

## Usage

```
InvertibleQ(phi)
```

## Arguments

| | |
|---|---|
| `phi` | a vector of AR coefficients |

## Details

The PACF is computed for lags 1,...,p using eqn. (1) in McLeod and Zhang (2006). The invertibility condition is satisfied if and only if all PACF values are less than 1 in absolute value.

## Value

TRUE, if invertibility condition is satisfied. FALSE, if not invertible.

## Author(s)

A.I. McLeod

## References

McLeod, A.I. and Zhang, Y. (2006). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

## See Also

`ARToPacf`

## Examples

```
#simple examples
InvertibleQ(0.5)
#find the area of the invertible region for AR(2).
#We assume that the parameters must be less than 2 in absolute value.
#From the well-known diagram in the book of Box and Jenkins (1970),
#this area is exactly 4.
NSIM<-10^4
phi1<-runif(NSIM, min=-2, max=2)
phi2<-runif(NSIM, min=-2, max=2)
k<-sum(apply(matrix(c(phi1,phi2),ncol=2), MARGIN=1, FUN=InvertibleQ))
area<-16*k/NSIM
area
```

`Jacobian` *Jacobian AR-coefficients to Partial Autocorrelations*

## Description

This is more or less and internal routine used by InformationMatrixZeta but it is described in more detail since it may be useful in other computations.

## Usage

```
Jacobian(zeta)
```

## Arguments

| | |
|---|---|
| `zeta` | partial autocorrelation parameters |

## Details

The computation is described in detail in McLeod and Zhang (2006, Section 2.2)

## Value

square matrix of order length(zeta)

## Author(s)

A.I. McLeod

## References

McLeod, A.I. and Zhang, Y. (2006). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

## See Also

`InformationMatrixARz`

## Examples

```
#In McLeod and Zhang (2006, p.603) a symbolic example is given
# for the AR(4).
#
 Jacobian(rep(0.8,4))
```

## JacobianK — *Internal Utility Function*

### Description

The matrix defined in eqn. (10) of McLeod and Zhang (2006). Used by the function `Jacobian`.

### Usage

```
JacobianK(zeta, k)
```

### Arguments

| | |
|---|---|
| `zeta` | partial autocorrelations |
| `k` | k-th Jacobian |

### Value

matrix

### Author(s)

A.I. McLeod

### References

McLeod, A.I. and Zhang, Y. (2006). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

### See Also

`Jacobian`

### Examples

```
JacobianK(rep(0.8,4),3)
```

## JarqueBeraTest — *Jarque-Bera Normality Test*

### Description

A powerful omnibus test for normality.

### Usage

```
JarqueBeraTest(z)
```

### Arguments

| | |
|---|---|
| `z` | vector of data |

### Details

This test is derived as a Lagrange multiplier test for normal distribution in the family of Pearson distributions (Jarque and Bera, 1987 ).

### Value

| | |
|---|---|
| `LM` | value of the LM statistic |
| `pvalue` | p-value |

### Author(s)

A.I. McLeod

### References

Jarque, C.M. and Bera, A.K. (1987). A Test for Normality of Observations and Regression Residuals. International Statistical Review 55, 163-172

### Examples

```
#some normal data
z<-rnorm(100)
JarqueBeraTest(z)
#some skewed data
z<-rexp(100)
JarqueBeraTest(z)
#some thick tailed data
z<-rt(100,5)
JarqueBeraTest(z)
```

---

## `LBQPlot` *Plot Ljung-Box Test P-value vs Lag*

---

### Description

The Ljung-Box portmanteau p-value is plotted vs lag.

### Usage

```
LBQPlot(obj, SquaredQ=FALSE)
```

### Arguments

| | |
|---|---|
| `obj` | output from FitAR or FitARLS |
| `SquaredQ` | default, SquaredQ=FALSE, regular autocorrelations. If SquaredQ=TRUE use autocorrelations of squared residuals. |

### Value

Plot is produced as a side-effect. No output

### Note

This function is normally invoked when plot.FitAR is used.

### Author(s)

A.I. McLeod

### References

Ljung, G.M. and Box, G.E.P. (1978) On a measure of lack of fit in time series models. Biometrika 65, 297-303.

McLeod, A.I. and Zhang, Y. (2006). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

### See Also

`plot.FitAR`, `FitAR`, `FitARLS`

### Examples

```
#fit subset AR and plot diagnostic check
 data(SeriesA)
 out<-FitARLS(SeriesA, c(1,2,7))
 LBQPlot(out)
#note that plot produces LBQPlot and RacfPlot
 plot(out)
```

---

## `LjungBoxTest` *Ljung-Box Test for Randomness*

---

### Description

The Ljung-Box Portmanteau test for the goodness of fit of ARIMA models is implemented.

### Usage

```
LjungBoxTest(res, k=0, lag.max=30, StartLag=10, SquaredQ=FALSE)
```

### Arguments

| | |
|---|---|
| `res` | residuals |
| `k` | number of ARMA parameters, default k=0 |
| `lag.max` | maximum lag, default MaxLag=30 |
| `StartLag` | test is done for lags m=StartLag:MaxLag, default StartLag=10 |
| `SquaredQ` | if TRUE, use squared residuals for ARCH test, default Squared=FALSE |

### Details

This test is described in detail in Wei (2006, p.153, eqn. 7.5.1).

## Value

A matrix with columns labelled m, Qm, pvalue, where m is the lag and Qm is the Ljung-Box Portmanteau statistic and pvalue its p-value. A powerful test for ARCH and other nonlinearities is obtained by using squared values of the series to be tested (McLeod & Li, 1983). Note that if Squared=TRUE is used the data "res" is centered by sample mean correction before squaring.

## Note

This test may also be used to test a time series for randomness taking k=0.

## Author(s)

A.I. McLeod

## References

W.W.S. Wei (2006, 2nd Ed.), *Time Series Analysis: Univariate and Multivariate Methods*.

A.I. McLeod. & W.K. Li (1983), Diagnostic checking ARMA time series models using squared-residual autocorrelations, *Journal of Time Series Analysis* **4**, 269–273.

## See Also

Box.test

## Examples

```
#test goodness-of-fit of AR(2) model fit to log(lynx)
data(lynx)
z<-log(lynx)
ans<-FitAR(z, 1:2)
#notice that the test is also available as a component of the output of FitAR (FitARLS al
ans$LjungBox
#a plot of the test is produced
plot(ans)
#doing the test manually
res<-resid(ans)
LjungBoxTest(res, k=2, lag.max=20, StartLag=5)

#test for subset case
z<-log(lynx)
pvec<-SelectModel(z, SubsetModel="z", Criterion="BIC", lag.max=10, Best=1)
ans<-FitAR(z, pvec)
plot(ans)
res<-resid(ans)
LjungBoxTest(res, k=length(pvec), lag.max=20, StartLag=11)
#test for ARCH effect,
LjungBoxTest(res,SquaredQ=TRUE)
```

`LoglikelihoodAR` *Exact Loglikelihood for AR*

### Description

The exact loglikelihood function, defined in eqn. (6) of McLeod & Zhang (2006) is computed. Requires O(n) flops, n=length(z).

### Usage

```
LoglikelihoodAR(phi, z, MeanValue = 0)
```

### Arguments

| | |
|---|---|
| `phi` | AR parameters |
| `z` | time series data, not assumed mean corrected |
| `MeanValue` | usually this is mean(z) but it could be another value for example the MLE of the mean |

### Details

Eqn (6) of McLeod and Zhang (2006) may be written

$$-(n/2)\log(\hat{\sigma}_a^2) - (1/2)\log(g_p),$$

where $\hat{\sigma}_a^2$ is the residual variance and $g_p$ is the covariance determinant.

### Value

The value of the loglikelihood is returned

### Warning

No check is done for stationary-causal process

### Note

For MLE computation it is better to use `FastLoglikelihoodAR` since for repeated likelihood evaluations this requires only O(1) flops vs O(n) flops, where n=length(z).

### Author(s)

A.I. McLeod

### References

McLeod, A.I. and Zhang, Y. (2006). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

### See Also

`FastLoglikelihoodAR`

### Examples

```
#Fit a subset model to Series A and verify the loglikelihood
 data(SeriesA)
 out<-FitAR(SeriesA, c(1,2,7))
 out
#either using print.default(out) to see the components in out
#or applying LoglikelihoodAR () by first obtaining the phi parameters as out$phiHat.

#
 LoglikelihoodAR(out$phiHat, SeriesA, MeanValue=mean(SeriesA))
```

---

## Ninemile — *Douglas Fir Treerings, Nine Mile Canyon, Utah, 1194-1964*

---

### Description

A treering time series comprises of 771 values showing a periodicity of around 10 years.

### Usage

```
data(Ninemile)
```

### Format

ts object with title attribute

### Source

Hipel, K.W. and McLeod, A.I. (2006). Time Series Modelling of Water Resources and Environmental Systems.

### References

McLeod, A.I. and Zhang, Y. (2006). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

### Examples

```
data(Ninemile)
ans<-FitAR(Ninemile, c(1,2,9))
summary(ans)
```

`PacfDL` *Partial Autocorrelations via Durbin-Levinson*

### Description

Given autocovariances, the partial autocorrelations and/or autoregressive coefficients in an AR may be determined using the Durbin-Levinson algorithm. If the autocovariances are sample autocovariances, this is equivalent to using the Yule-Walker equations. But as noted below our function is more general than the built-in R functions.

### Usage

```
PacfDL(c, LinearPredictor = FALSE)
```

### Arguments

`c` autocovariances at lags 0,1,...,p=length(c)-1

`LinearPredictor` if TRUE, AR coefficients are also determined using the Yule-Walker method

### Details

The Durbin-Levinson algorithm is described in many books on time series and numerical methods, for example Percival and Walden (1993, eqn 403).

### Value

If LinearPredictor=FALSE, vector of length p=length(c)-1 containing the partial autocorrelations at lags 1,...,p. Otherwise a list with components:

`Pacf` vector of partial autocorrelations

`ARCoefficients` vector of AR coefficients

`ResidualVariance` residual variance for AR(p)

### Warning

Stationarity is not tested.

### Note

Sample partial autocorrelations can also be computed with the `acf` function and Yule-Walker estimates can be computed with the `ar` function. Our function `PacfDL` provides more flexibility since then input c may be any valid autocovariances not just the usual sample autocovariances. For example, we can determine the minimum mean square error one-step ahead linear predictor of order p for theoretical autocovariances from a fractional arma or other linear process.

### Author(s)

A.I. McLeod

### References

Percival, D.B. and Walden, A.T. (1993). Spectral Analysis For Physical Applications, Cambridge University Press.

### See Also

acf, ar

### Examples

```
#first define a function to compute the Sample Autocovariances
 sacvf<-function(z, lag.max){
 c(acf(z, plot=FALSE, lag.max=lag.max)$acf)*(length(z)-1)/length(z)
 }
#now compute PACF and also fit AR(7) to SeriesA
 data(SeriesA)
 ck<-sacvf(SeriesA, 7)
 PacfDL(ck)
 PacfDL(ck, LinearPredictor = TRUE)
#compare with built-in functions
 pacf(SeriesA, lag.max=7, plot=FALSE)
 ar(SeriesA, lag.max=7, method="yw")
#fit an optimal linear predictor of order 10 to MA(1)
 g<-TacvfMA(0.8,5)
 PacfDL(g, LinearPredictor=TRUE)
#
#Compute the theoretical pacf for MA(1) and plot it
 ck<-c(1,-0.4,rep(0,18))
 AcfPlot(PacfDL(ck)$Pacf)
 title(main="Pacf of MA(1), r(1)=-0.4")
```

---

## PacfPlot — *Plot Partial Autocorrelations and Limits*

---

### Description

The partial autocorrelations and their individual 95 percent confidence intervals are plotted under the assumption the model is contained in an AR(P), where P is a specified upper limit.

### Usage

```
PacfPlot(z, lag.max = 15, ...)
```

### Arguments

| | |
|---|---|
| `z` | time series |
| `lag.max` | maximum lag, P |
| `...` | optional parameters passed through to plot. |

### Details

The Burg algorithm is used to estimate the PACF.

### Value

No value is returned. Graphical output is produced as side-effect.

### Author(s)

A.I. McLeod

### References

McLeod, A.I. and Zhang, Y. (2006). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

### See Also

ar.burg pacf

### Examples

```
#For the log(lynx) series and taking lag.max=15, the PacfPlot and
# the minimum BIC subset selection produce the same result.
z<-log(lynx)
PacfPlot(z)
SelectModel(z,lag.max=15,SubsetModel="z",Best=1,Criterion="BIC")
```

---

## PacfToAR *Transform from PACF Parameters to AR Coefficients*

---

### Description

Transforms AR partical autocorrelation function (PACF) parameters to AR coefficients based on the Durbin-Levinson recursion.

### Usage

```
PacfToAR(zeta)
```

### Arguments

| | |
|---|---|
| zeta | vector of AR PACF parameters |

### Details

see Mcleod and Zhang (2006)

### Value

vector of AR coefficients

### Author(s)

A.I. McLeod

### References

McLeod and Zhang (2006)

### See Also

InvertibleQ, PacfToAR

### Examples

```
somePACF<-c(0.5,0.6,0.7,0.8,-0.9,-0.8)
someAR<-PacfToAR(somePACF)
test<-ARToPacf(someAR)
#this should be very small
sum(abs(test-somePACF))
```

---

PlotARSdf *Plot AR or ARMA Spectral Density*

---

### Description

Constructs a plot of the AR spectral density function

### Usage

```
PlotARSdf(phi = NULL, theta = NULL, units = "radial", logSdf = FALSE, Innovation
```

### Arguments

| | |
|---|---|
| phi | AR Coefficients |
| theta | MA Coefficients |
| units | default is "radial" |
| logSdf | default is FALSE otherwise log sdf is plotted |
| InnovationVariance | innovation variance, default is 1 |
| main | optional plot title |
| sub | optional subtitle |
| ... | optional arguments |

### Details

The spectral density function is symmetric and defined in (-pi,pi) but plotted over (0,pi). If units are not "radial", it is plotted over (0, 0.5).

### Value

No value. Plot is generated as product by using R plot.

### See Also

ARSdf

## Examples

```
#AR(1)
PlotARSdf(0.8)
#MA(1)
PlotARSdf(theta=0.8)
#ARMA(1,1)
PlotARSdf(0.9,0.5)
#white noise
PlotARSdf()
```

---

`RacfPlot` *Residual Autocorrelation Plot*

---

## Description

Residual autocorrelation plot for "FitAR" objects. This plot is useful for diagnostic checking models fit with the functions `FitAR` and `FitARLS`

## Usage

```
RacfPlot(obj, lag.max = 1000, SquaredQ=FALSE)
```

## Arguments

| | |
|---|---|
| `obj` | output from FitAR or FitARLS |
| `lag.max` | maximum lag. Set to 1000 since minimum of this value and the value in the `obj` is used. |
| `SquaredQ` | default is FALSE. For squared residual autocorrelations, set to TRUE |

## Details

The standard deviations of the residual autocorrelations are obtained from McLeod (1978, eqn.16) or McLeod and Zhang (2006, eqn.16). Simultaneous confidence bounds are shown and constructed using the Bonferonni approximation as suggested by Hosking and Ravishanker (1993)

## Value

Plot is produced as a side-effect. No output

## Note

This function is normally invoked when plot.FitAR is used.

## Author(s)

A.I. McLeod

## References

Hosking, J.R.M. and Ravishanker, N. (1993) Approximate simultaneous significance intervals for residual autocorrelations of autoregressive-moving average time series models. Journal of Time Series Analysis 14, 19-26.

McLeod, A.I. (1978), On the distribution and applications of residual autocorrelations in Box-Jenkins modelling, Journal of the Royal Statistical Society B 40, 296-302.

McLeod, A.I. and Zhang, Y. (2006). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

## See Also

plot.FitAR, FitAR, FitARLS

## Examples

```
#fit subset AR and plot diagnostic check
 data(SeriesA)
 out<-FitARLS(SeriesA, c(1,2,7))
 RacfPlot(out)
#note that plot produces LBQPlot and RacfPlot
 plot(out)
#check squared residuals
 RacfPlot(out, SquaredQ=TRUE)
```

---

Readts *Input a Time Series*

---

## Description

This function inputs time series stored in ASCII in a format that the first line in the file is a title, next few lines, beginning with a #, are comments, and the remaining lines contain the data. Here is an example:

```
 Changes In Global Temperature, Annual, 1880-1985
cr #Surface air "temperature change" for the globe, 1880-1985.
cr #Degrees Celsius.  "Temperature change" actually means temperature
cr #Surface Air Temperature", `Journal of Geophysical Research`, Vol.
92,
cr -.40 -.37 -.43 ....
cr ...............
cr .27 .42 .02 .30 .09 .05
cr
```

## Usage

```
Readts(file = "", freq = 1, start = 1, VerboseQ=TRUE)
```

### Arguments

| | |
|---|---|
| `file` | location for input file |
| `freq` | tsp parameter, =1, annual, =12 monthly etc |
| `start` | tsp parameter |
| `VerboseQ` | normally prompt for arguments but set VerboseQ=FALSE to automate |

### Value

ts object with attribute 'title'

### Author(s)

A.I. McLeod

### See Also

`scan`, `ts`,

### Examples

```
#You will need to change save the data given above in a file
#and change the directory as appropriate
#z<-Readts(file="d:/datasets/mhsets/annual/globtp.1", start=1880, VerboseQ=FALSE)
```

---

## `SelectModel` *Select Best AR, ARz or ARp Model*

---

### Description

The AIC/BIC/UBIC criterion is used to select the best fitting AR or subset AR model. The result may be plotted using `plot`.

### Usage

```
SelectModel(z, lag.max = 15, SubsetModel = c("n", "p", "z"), Criterion = "defaul
```

### Arguments

| | |
|---|---|
| `z` | time series data |
| `lag.max` | maximum order of autoregression |
| `SubsetModel` | default is no subset, SubsetModel="n". Alternatives are ARp, SubsetModel="p" or ARz, SubsetModel="z" |
| `Criterion` | default is "UBIC". Options: "BIC" and "AIC". |
| `Best` | final number of models to be selected |
| `Candidates` | number of models initially selected using the approximate criterion |

## Details

McLeod and Zhang (2006) outline an approximate AIC/BIC selection algorithm. This algorithm is a refinement of that method. The refinement consists of automatically look for the best k candidates, where k=`Candidates`. Then the exact likelihood is evaluated for all k candidates. Out of these k candidates, the best q = `Best` are then selected. This two-step procedure is needed because if k is too low, the approximate AIC/BIC rankings may not agree with the exact rankings. This strategy is used for model selection for AR, ARz and ARp models. A plot method is available to graph the output. The UBIC developed by Chen and Chen (2007) is an important extension of the BIC criterion for subset selection. In the non-subset case UBIC is equivalent to BIC.

## Value

When `Best` $= 1$, a vector is returned indicated the lag or lags included in the model. The null model is indicated by returning 0 for the lag. An object with class "Selectmodel" is returned when `Best` $> 1$. If SubsetQ = FALSE, a matrix is return whose first column shows p and second AIC or BIC. Otherwise for subset selection, the result is a list with q components, where q=BEST. When Criterion="UBIC", the components in this list are:

| | |
|---|---|
| `p` | lags present, a 0 indicates the null model |
| `UBIC` | exact UBIC |

Similarly for the AIC/BIC case.

The components are arranged in order of the criterion used.

When SubsetModel="p" or "z", an attribute "model" indicating "ARp" or "ARz" is included.

## Warning

Setting `Candidates` too low can result in anomalous results. For example if `Candidates`=1, we find that the top ranking model may depend on how large `Best` is set. This phenomenon is due to the fact that among the best AIC/BIC models there is sometimes very little difference in their AIC/BIC scores. Since the initial ranking of the Candidates is done using the approximate likelihood, the final ranking using the exact likelihood may change.

## Note

For white noise, the best model is the null model, containing no lags. This is indicating by setting the model order, $p = 0$.

## Author(s)

A.I. McLeod

## References

McLeod, A.I. and Zhang, Y. (2006). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

Chen, J. and Chen, Z. (2007). Extended Bayesian Information Criteria for Model Selection with Large Model Space. Preprint.

## See Also

`plot.Selectmodel`, `PacfPlot`, `PacfPlot`, `FitAR`

## Examples

```
#Example 1: Find an ARp subset model for lynx data using BIC
z<-log(lynx)
out<-SelectModel(z, SubsetModel="p", Criterion="BIC", Best=5)
plot(out)
#
#Example 2: Find an ARz subset model for lynx data using BIC
out<-SelectModel(z, SubsetModel="z", Criterion="BIC", Best=5)
plot(out)
#
#Example 3: Select an AR(p) model
out<-SelectModel(z, Criterion="BIC", Best=5)
out
plot(out)
#
#Example 4: Fit subset models to lynx series
z<-log(lynx)
#requires library leaps. Should be automatically when FitAR package is loaded.
pvec <- SelectModel(z, lag.max=11, SubsetModel="p", Criterion="AIC", Best=1)
ans1 <- FitARLS(z, pvec)
pvec <- SelectModel(z, lag.max=11, SubsetModel="z", Criterion="AIC", Best=1)
ans2<-FitAR(z, pvec)
summary(ans1)
summary(ans2)
```

---

# SeriesA    *Series A, Chemical Process Concentration Readings*

---

## Description

Chemical process concentration readings for every 2 hours.

## Usage

```
data(SeriesA)
```

## Format

ts object with attribute "title"

## Details

Box and Jenkins (1970) fit an ARMA(1,1) and ARIMA(0,1,1) to this series. Cleveland() suggested a subset AR(1,2,7). McLeod and Zhang (2006) fit a subset ARz(1,2,6,7) parameterized with the partial autocorrelations.

## Source

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

## References

Cleveland, W.S. (1971) The inverse autocorrelations of a time series and their applications. Technometrics 14, 277-298.

McLeod, A.I. and Zhang, Y. (2006). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

## Examples

```
data(SeriesA)
#fit subset models
FitARLS(SeriesA, c(1,2,7))
FitAR(SeriesA, c(1,2,6,7))
```

---

# SiddiquiMatrix — *Covariance Matrix of MLE Parameters in an AR(p)*

---

## Description

A direct method of computing the inverse of the covariance matrix of p successive observations in an AR(p) with unit innovation variance given by Siddiqui (1958) is implemented. This matrix, divided by n = length of series, is the covariance matrix for the MLE estimates in a regular AR(p).

## Usage

```
SiddiquiMatrix(phi)
```

## Arguments

| | |
|---|---|
| `phi` | coefficients in a regular AR(p) |

## Value

Matrix, covariance matrix of MLE estimates

## Note

No check on whether the parameters are in the stationary region is done. It has been shown a necessary and sufficient condition for the parameters to be in the stationary region is that this matrix should be positive-definite (Pagano, 1973). But computationally it is probably better to test for stationarity by using `ARToPacf` to transform to the PACF and then check that the absolute value of all partial autocorrelations are less than 1.

## Author(s)

A.I. McLeod

## References

Siddiqui, M.M. (1958) On the inversion of the sample covariance matrix in a stationary autoregressive process. Annals of Mathematical Statistics 29, 585-588.

Pagano, M. (1973), When is an autoregressive scheme stationary? Communications in Statistics A 1, 533-544.

### See Also

FitAR, FitARLS

### Examples

```
#compute the inverse directly and by Siddiqui's method and compare:
phi<-PacfToAR(rep(0.8,5))
A<-SiddiquiMatrix(phi)
B<-solve(toeplitz(TacvfAR(phi, lag.max=length(phi)-1)))
max(abs(A-B))
```

---

## SimulateGaussianAR *Autoregression Simulation*

---

### Description

Simulate a mean-zero stationary Gaussian AR(p) time series.

### Usage

```
SimulateGaussianAR(phi, n = 100, InnovationVariance = 1)
```

### Arguments

| | |
|---|---|
| phi | vector containing AR coefficients |
| n | length of time series |
| InnovationVariance | innovation variance |

### Details

The p initial values are simulated using the appropriate multivariate distribution as was suggested in McLeod (1975). The R function rnorm() is used.

### Value

A vector of length n, the simulated series

### Note

If the process is non-stationary, then random initial values are used determined by the first p innovations.

### Author(s)

A.I. McLeod

### References

McLeod, A.I. (1975), Derivation of the theoretical autocorrelation function of autoregressive moving-average time series, *Applied Statistics* **24**, 255–256.

Percival, D.B. and Walden, A.T. (1993), *Spectral Analysis for Physical Applications*.

### See Also

Boot.FitAR

### Examples

```
#Percival and Walden (1993, p.46) illustrated a time series with a
#very peaked spectrum with the AR(4) with coefficients
#c(2.7607,-3.8106,2.6535,-0.9238) with NID(0,1) innovations.
#
z<-SimulateGaussianAR(c(2.7607,-3.8106,2.6535,-0.9238),1000)
library(lattice)
TimeSeriesPlot(z)
```

---

## TacvfAR — *Theoretical Autocovariance Function of AR*

---

### Description

The theoretical autocovariance function of an AR(p) with unit variance is computed. This algorithm has many applications. In this package it is used for the computation of the information matrix, in simulating p initial starting values for AR simulations and in the computation of the exact mle for the mean.

### Usage

```
TacvfAR(phi, lag.max = 20)
```

### Arguments

| | |
|---|---|
| `phi` | vector of AR coefficients |
| `lag.max` | computes autocovariances lags 0,1,...,maxlag |

### Details

The algorithm given by McLeod (1975) is used.

The built-in R function ARMAacf could also be used but it is quite complicated and apart from the source code, the precise algorithm used is not described. The only reference given for ARMAacf is the Brockwell and Davis (1991) but this text does not give any detailed exact algorithm for the general case.

Another advantage of TacvfAR over ARMAacf is that it will be easier for to translate and implement this algorithm in other computing environments such as MatLab etc. since the code is entirely written in R.

### Value

Vector of length lag.max+1 containing the autocovariances at lags 0,...,lag.max is returned.

### Author(s)

A.I. McLeod

### References

McLeod, A.I. (1975), Derivation of the theoretical autocorrelation function of autoregressive moving-average time series. Applied Statistics, 24, 255-256.

### See Also

ARMAacf, InformationMatrixAR, GetARMeanMLE, SimulateGaussianAR

### Examples

```
#calculate and plot the autocorrelations from an AR(2) model
# with parameter vector c(1.8,-0.9).
 g<-TacvfAR(c(1.8,-0.9),20)
 AcfPlot(g/g[1], LagZeroQ=FALSE)
```

---

## TacvfMA — *Theoretical Autocovariances for Moving Average Process*

---

### Description

The theoretical autocovariance function of a MA(q) with unit variance is computed.

### Usage

```
TacvfMA(theta, lag.max = 20)
```

### Arguments

| | |
|---|---|
| `theta` | q parameters in MA(q) |
| `lag.max` | number of lags required. |

### Details

The first q+1 values are determined using a matrix multiplication - avoiding a loop. The remaining values set to zero.

### Value

Vector of length q+1 containing the autocovariances at lags 0,1,...,lag.max

### Note

See Details in TacvfAR for why we prefer to use this algorithm instead of ARMAacf

### Author(s)

A.I.McLeod

### References

McLeod, A.I. and Zhang, Y. (2006), Partial autocorrelation parameterization for subset autoregression. *Journal of Time Series Analysis*, **27**, 599-612.

### See Also

ARMAacf, TacvfAR

### Examples

```
TacvfMA(c(1.8,,-0.9),10)
```

---

TimeSeriesPlot *Multi-Panel or Single-Panel Time Series Plot with Aspect-Ratio Control*

---

### Description

Cleveland pointed out that the aspect-ratio is important in graphically showing the rate-of-change or shape information. For many time series, it is preferably to set this ratio to 0.25 than the default. In general, Cleveland shows that the best choice of aspect-ratio is often obtained by if the average apparent absolute slope in the graph is about 45 deg. But for many stationary time series, this would result in an aspect-ratio which would be too small. As a comprise we have chosen a default of 0.25 but the user can select other choices.

### Usage

```
TimeSeriesPlot(z, SubLength = Inf, aspect = 0.25, type="l", ylab=NULL, main=NULL
```

### Arguments

| | |
|---|---|
| `z` | ts object or vector, time series data |
| `SubLength` | maximum number of data points per panel. Default SubLength=Inf and regular graphics. For trellis graphics, set SubLength to a finite value. |
| `aspect` | optional setting for the aspect-ratio |
| `type` | plot type, default type="l" join points with lines |
| `ylab` | optional label for vertical axis |
| `main` | optional title |
| `...` | optional arguments passed to `xyplot` |

### Details

If z has attribute "title" containing a character string, this is used on the plot. Time series input using the function Readts always have this attribute set.

### Value

If `SubLength` is finite, the lattice package is used and a graphic object of class trellis is produced. Otherwise, the standard R graphics system is used and the plot is produced as a side-effect and there is no output.

### Note

Requires `lattice` library

### Author(s)

A.I. McLeod

### References

W.S. Cleveland (1993), *Visualizing Data*.

### See Also

plot.ts, Readts

### Examples

```
#from built-in datasets
 TimeSeriesPlot(AirPassengers)
 title(main="Monthly number of trans-Atlantic airline passengers")
#
#compare plots for lynx series
plot(lynx)
win.graph()
TimeSeriesPlot(lynx, type="o", pch=16, ylab="# pelts", main="Lynx Trappings")
#
#lattice style plot
data(Ninemile)
TimeSeriesPlot(Ninemile, SubLength=200)
```

---

## USTobacco *U.S. Tobacco Production, 1871-1984*

---

### Description

Annual U.S. tobacco production, 1871-1984, in millions of pounds.

### Usage

```
data(USTobacco)
```

### Format

The format is: Time-Series [1:114] from 1871 to 1984: 327 385 382 217 609 466 621 455 472 469 ... - attr(*, "title")= chr "Tobacco production,US, 1871-1984"

### Details

Wei (2006, p.120, Example 6.6) fits an ARIMA(0,1,1) to the logarithms. But a more accurate Box-Cox analysis indicates a square-root transformation should be used. A more complex ARIMA-GARCH model is also suggested by Wei (2006).

### Source

Wei, W.W.S. (2006, Series W6, p.570), *Time Series Analysis: Univariate and Multivariate Methods*. 2nd Ed., New York: Addison-Wesley.

### Examples

```
#From a plot of the series, we see that the variance is increasing with level.
#From the acf of the first differences an ARIMA(0,1,1) is suggested.
 data(USTobacco)
# layout(matrix(c(1,2,1,2),ncol=2))
 plot(USTobacco)
 lines(lowess(time(USTobacco), USTobacco), lwd=2, col="blue")
 acf(diff(USTobacco, differences=1))
```

---

## VarianceRacfAR *Covariance Matrix Residual Autocorrelations for AR*

---

### Description

Computes the variance-covariance matrix for the residual autocorrelations in an AR(p).

### Usage

```
VarianceRacfAR(phi, MaxLag, n)
```

### Arguments

| | |
|---|---|
| phi | vector of AR coefficients |
| MaxLag | covariance matrix for residual autocorrelations at lags 1,...,m, where m=MaxLag is computes |
| n | length of time series |

### Details

The covariance matrix for the residual autocorrelations is derived in McLeod (1978, eqn. 15) for the general ARMA case. With this function one can obtain the standard deviations of the residual autocorrelations which can be used for diagnostic checking with RacfPlot.

### Value

The m-by-m covariance matrix of residual autocorrelations at lags 1,...,m, where m=MaxLag.

### Note

The derivation assumes normality of the innovations, mle estimation of the parameters and a known mean-zero time series. It is easily seen that the same result still holds for IID innovations with mean zero and finite variance, any first-order efficient estimates of the parameters including the AR coefficients and mean.

### Author(s)

A.I. McLeod

### References

McLeod, A.I. (1978), On the distribution and applications of residual autocorrelations in Box-Jenkins modelling, *Journal of the Royal Statistical Society B*, **40**, 296–302

### See Also

`VarianceRacfARp`, `VarianceRacfARz`, `RacfPlot`

### Examples

```
VarianceRacfAR(0.5,5,100)
```

---

## `VarianceRacfARp` *Covariance Matrix Residual Autocorrelations for ARp*

---

### Description

The ARp subset model is defined by taking a subset of the parameters in the regular AR(p) model. With this function one can obtain the standard deviations of the residual autocorrelations which can be used for diagnostic checking with `RacfPlot`.

### Usage

```
VarianceRacfARp(phi, lags, MaxLag, n)
```

### Arguments

| | |
|---|---|
| `phi` | vector of AR coefficients |
| `lags` | lags in subset AR |
| `MaxLag` | covariance matrix for residual autocorrelations at lags 1,...,m, where m=MaxLag is computes |
| `n` | length of time series |

### Details

The covariance matrix for the residual autocorrelations is derived in McLeod (1978, eqn. 15) for the general ARMA case. McLeod (1978, eqn. 35) specializes this result to the subset case.

### Value

The m-by-m covariance matrix of residual autocorrelations at lags 1,...,m, where m=MaxLag.

### Author(s)

A.I. McLeod

### References

McLeod, A.I. (1978), On the distribution and applications of residual autocorrelations in Box-Jenkins modelling, *Journal of the Royal Statistical Society B*, **40**, 296-302.

### See Also

VarianceRacfAR, VarianceRacfARz, RacfPlot

### Examples

```
#the standard deviations of the first 5 residual autocorrelations
#to a subset AR(1,2,6) model fitted to Series A is
v<-VarianceRacfARp(c(0.36,0.23,0.23),c(1,2,6), 5, 197)
sqrt(diag(v))
```

---

## VarianceRacfARz *Covariance Matrix Residual Autocorrelations for ARz*

---

### Description

The ARz subset model is defined by taking a subset of the partial autocorrelations (zeta parameters) in the AR(p) model. With this function one can obtain the standard deviations of the residual autocorrelations which can be used for diagnostic checking with RacfPlot.

### Usage

```
VarianceRacfARz(zeta, lags, MaxLag, n)
```

### Arguments

| | |
|---|---|
| zeta | zeta parameters (partial autocorrelations) |
| lags | lags in model |
| MaxLag | covariance matrix for residual autocorrelations at lags 1,...,m, where m=MaxLag is computes |
| n | length of time series |

### Details

The covariance matrix of the residual autocorrelations in the subset ARz case is derived in McLeod and Zhang (2006, eqn. 16)

### Value

The m-by-m covariance matrix of residual autocorrelations at lags 1,...,m, where m=MaxLag.

### Author(s)

A.I. McLeod

### References

McLeod, A.I. and Zhang, Y. (2006). Partial autocorrelation parameterization for subset autoregression. *Journal of Time Series Analysis*, **27**, 599-612.

### See Also

VarianceRacfAR, VarianceRacfARz, RacfPlot

## Examples

```
#the standard deviations of the first 5 residual autocorrelations
#to a subset AR(1,2,6) model fitted to Series A is
v<-VarianceRacfARp(c(0.36,0.23,0.23),c(1,2,6), 5, 197)
sqrt(diag(v))
```

---

Willamette *Willamette Riverflow Time Series*

---

## Description

Monthly flows of the Willamette River, Salem, Oregon, Oct. 1950 - Aug. 1983. Percival and Walden (1993, Ch. 10.15) fit high-order AR models to estimate the spectral density function.

## Usage

```
data(Willamette)
```

## Format

The format is: Time-Series [1:395] from 1951 to 1984: 8.95 9.49 10.19 10.96 11.08 ... - attr(*, "title")= chr "Willamette river, Monthly, Salem, Oregon, Oct. 1950 - Aug. 1983"

## Source

http://faculty.washington.edu/dbp/sapabook.html

## References

Percival, D.B. and Walden, A.T. (1993), *Spectral Analysis for Physical Applications*. Cambridge University Press.

## Examples

```
#Percival and Walden (1993) fit an AR(27).
#Compare spectral densities with subset AR's.
data(Willamette)
pmax<-27
sdfplot(FitAR(log(Willamette), pmax))
win.graph()
p<-SelectModel(log(Willamette), SubsetModel="z", lag.max=pmax, Best=1)
sdfplot(FitAR(log(Willamette), p))
win.graph()
p<-SelectModel(log(Willamette), SubsetModel="p", lag.max=pmax, Best=1)
sdfplot(FitARLS(log(Willamette), p))
```

## bxcx — *Box-Cox Transformation and its Inverse*

### Description

Box-Cox or power transformation or its inverse. For $lambda \neq 0$, the Box-Cox transformation of x is $(x-1)^{\lambda}/\lambda$ whereas the regular power transformation is simply $x^{\lambda}$. When $\lambda = 0$, it is log in both cases. The inverse of the Box-Cox and the power transform can also be obtained.

### Usage

```
bxcx(x, lambda, InverseQ = FALSE, type = "BoxCox")
```

### Arguments

| | |
|---|---|
| `x` | a vector or time series |
| `lambda` | power transformation parameter |
| `InverseQ` | if TRUE, the inverse transformation is done |
| `type` | either "BoxCox" or "power" |

### Value

a vector or time series of the transformed data

### Author(s)

A.I. McLeod

### References

Box, G. E. P. and Cox, D. R. (1964) An analysis of transformations. Journal of Royal Statistical Society, Series B, vol. 26, pp. 211-246.

### See Also

BoxCox

### Examples

```
#lambda=0.5
z<-AirPassengers; lambda<-0.5
y<-bxcx(z, lambda)
z2<-bxcx(y, lambda, InverseQ=TRUE)
sum(abs(z2-z))
#
z<-AirPassengers; lambda<-0.0
y<-bxcx(z, lambda)
z2<-bxcx(y, lambda, InverseQ=TRUE)
sum(abs(z2-z))
```

`coef.FitAR` *Display Estimated Parameters from Output of FitAR*

### Description

Method function to display fitted parameters, their standard errors and Z-ratio for AR models fit with FitAR and FitARLS.

### Usage

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

### Arguments

| | |
|---|---|
| `object` | `obj` the output from `FitAR` or `FitARLS` |
| `...` | 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. and Zhang, Y. (2006). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

### Examples

```
# Fit subset AR to SeriesA
 data(SeriesA)
 outA<-FitAR(SeriesA, c(1,2,7))
 coef(outA)
#
 outALS<-FitARLS(SeriesA, c(1,2,7))
 coef(outALS)
```

## fitted.FitAR *Fitted Values from "FitAR" Object*

### Description

Method function, extracts fitted values from `FitAR` object.

### Usage

```
## S3 method for class 'FitAR':
fitted(object, ...)
```

### Arguments

| | |
|---|---|
| `object` | object of class "FitAR" |
| `...` | optional arguments |

### Value

vector of fitted values

### Author(s)

A.I. McLeod

### See Also

`FitAR`, `FitARLS`,

### Examples

```
data(SeriesA)
out<-FitAR(SeriesA, c(1,2,6,7))
fitted(out)
```

## plot.FitAR *Plot Method for "FitAR" Object*

### Description

Diagnostic plots: portmanteau p-values; residual autocorrelation plot; normal probability plot and Jarque-Bera test; spectral density function

### Usage

```
plot.FitAR(x, clearGraphics=TRUE, terse=TRUE, ...)
```

### Arguments

| | |
|---|---|
| `x` | object of class "FitAR" |
| `clearGraphics` | close all graphics windows |
| `terse` | if TRUE, only one graph is produced, otherwise many diagnostic plots. |
| `...` | optional arguments |

### Value

No value is returned. Plots are produced as side-effect.

### Author(s)

A.I. McLeod

### References

McLeod, A.I. and Zhang, Y. (2006). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

### See Also

`summary.FitAR`, `FitAR`, `FitARLS` `JarqueBeraTest` `RacfPlot` `LBQPlot`

### Examples

```
data(SeriesA)
obj<-FitAR(SeriesA, c(1,2,6,7))
plot(obj)
```

---

## `plot.Selectmodel` *Subset AR Graph for "Selectmodel" Object*

---

### Description

A graphical depiction is given of the output from `SelectModel`.

### Usage

```
plot.Selectmodel(x, ...)
```

### Arguments

| | |
|---|---|
| `x` | out from `SelectModel` |
| `...` | optional arguments |

### Details

The relative plausibility of Model A vs the best Model B, is defined as $R = e^{(AIC_B - AIC_A)/2}$. Values of R less than 1 R is defined similarly if the BIC/UBIC criterion is used.

### Value

No value. Plot produced as side-effect.

### Author(s)

A.I. McLeod

### See Also

`SelectModel`

### Examples

```
data(Willamette)
out<-SelectModel(log(Willamette),lag.max=150,SubsetModel="n",Best=5,Criterion="AIC")
plot(out)
```

---

## print.FitAR *Print Method for "FitAR" Object*

---

### Description

A terse summary is given

### Usage

```
print.FitAR(x, ...)
```

### Arguments

| | |
|---|---|
| x | object of class "FitAR" |
| ... | optional arguments |

### Value

A terse summary is displayed

### Author(s)

A.I. McLeod

### References

McLeod, A.I. and Zhang, Y. (2006). Partial autocorrelation parameterization for subset autoregression. Journal of Time Series Analysis, 27, 599-612.

### See Also

`summary.FitAR`, `FitARLS`

### Examples

```
data(SeriesA)
FitAR(SeriesA, c(1,2,6,7))
```

## residuals.FitAR *Extract Residual from "FitAR" Object*

### Description

Method function.

### Usage

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

### Arguments

| | |
|---|---|
| `object` | object of class "FitAR" |
| `...` | optional arguments |

### Value

Vector of residuals

### Author(s)

A.I. McLeod

### See Also

FitAR, FitARLS,

### Examples

```
data(SeriesA)
out<-FitAR(SeriesA, c(1,2,6,7))
resid(out)
```

## sdfplot.Arima *Spectral Density of Fitted ARIMA Model*

### Description

Method for class "Arima" for sdfplot.

### Usage

```
sdfplot.Arima(obj, ...)
```

### Arguments

| | |
|---|---|
| `obj` | object class "Arima" |
| `...` | optional arguments |

### Value

None. Plot is produced using plo.t

### Author(s)

A.I. McLeod

### See Also

ARSdf, sdfplot, sdfplot.FitAR, sdfplot.ts

### Examples

```
data(SeriesA)
sdfplot(SeriesA,c(1,0,1))
```

---

sdfplot.FitAR *Autoregressive Spectral Density Estimation for "FitAR"*

---

### Description

Methods function for sdfplot. Autoregressive spectral density function estimation using the result output from FitAR.

### Usage

```
## S3 method for class 'FitAR':
sdfplot(obj, ...)
```

### Arguments

| | |
|---|---|
| obj | object, class"FitAR" |
| ... | optional arguments |

### Value

Plot produced as side-effect. No output.

### Author(s)

A.I. McLeod

### See Also

sdfplot, FitAR, FitARLS

## Examples

```
#Use AIC to select best subset model to fit to lynx data and
#plot spectral density function
data(SeriesA)
pvec<-SelectModel(SeriesA, SubsetModel="p", lag.max=10, Best=1)
ans<-FitAR(SeriesA, pvec)
sdfplot(ans)
#
#plot sdf and put your own title
z<-c(SeriesA)
pvec<-SelectModel(z, SubsetModel="p", lag.max=10, Best=1)
ans<-FitAR(z, pvec)
sdfplot(ans)
title(main="Example SDF")
```

---

# `sdfplot` *Autoregressive Spectral Density Estimation*

---

## Description

Generic function. Methods are available for "FitAR", "ar", "Arima", "ts" and "numeric".

## Usage

```
sdfplot(obj, ...)
```

## Arguments

| | |
|---|---|
| `obj` | input object |
| `...` | optional arguments |

## Value

Plot produced as side-effect. No output.

## Author(s)

A.I. McLeod

## See Also

`sdfplot`, `FitAR`, `FitARLS`

## Examples

```
#Use AIC to select best subset model to fit to lynx data and
#plot spectral density function
data(SeriesA)
pvec<-SelectModel(SeriesA, SubsetModel="p", lag.max=10, Best=1)
ans<-FitAR(SeriesA, pvec)
sdfplot(ans)
#
# fit ARMA and plot sdf
```

```
ans<-arima(SeriesA, c(1,0,1))
sdfplot(ans)
```

---

sdfplot.ar *Autoregressive Spectral Density Estimation for "ar"*

---

### Description

Method for class "ar" for sdfplot.

### Usage

```
## S3 method for class 'ar':
sdfplot(obj, ...)
```

### Arguments

| | |
|---|---|
| `obj` | class "ar" object, output from `ar` |
| `...` | optional arguments |

### Value

None. Plot is produced using plot

### Author(s)

A.I. McLeod

### See Also

`ARSdf`, `sdfplot`, `sdfplot.FitAR`, `sdfplot.ts`

### Examples

```
#Fit AR(p) using AIC model selection and Burg estimates. Plot spectral density
#estimate
ans<-ar(lynx, lag.max=20)
sdfplot(ans)
```

## sdfplot.numeric — *Autoregressive Spectral Density Estimation for "numeric"*

### Description

Method function for vectors, class "numeric"

### Usage

```
## S3 method for class 'numeric':
sdfplot(obj, ...)
```

### Arguments

| | |
|---|---|
| `obj` | object, class"numeric", a vector |
| `...` | optional arguments |

### Value

Plot produced as side-effect. No output.

### Author(s)

A.I. McLeod

### See Also

`sdfplot`

### Examples

```
sdfplot(lynx)
```

## sdfplot.ts — *Autoregressive Spectral Density Estimation for "ts" Object*

### Description

Methods function for "ts".

### Usage

```
## S3 method for class 'ts':
sdfplot(obj, ...)
```

### Arguments

| | |
|---|---|
| `obj` | object, class"ts" |
| `...` | optional arguments |

### Value

Plot produced as side-effect. No output.

### Author(s)

A.I. McLeod

### See Also

sdfplot

### Examples

```
data(SeriesA)
sdfplot(SeriesA)
```

---

## summary.FitAR *Summary Method for "FitAR" Object*

---

### Description

summary for "FitAR" object

### Usage

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

### Arguments

| | |
|---|---|
| object | "FitAR" object |
| ... | optional arguments |

### Value

A printed summary is given

### Author(s)

A.I. McLeod

### See Also

print.FitAR, FitAR, FitARLS,

### Examples

```
data(SeriesA)
out<-FitAR(SeriesA, c(1,2,6,7))
summary(out)
```

# Index