Package ‘perARMA’

May 9, 2026

Type Package

Title Periodic Time Series Analysis

Version 1.7

Maintainer Karolina Marek <karolina.marek10@gmail.com>

Description Identification, model fitting and estimation for time series with periodic structure.
Additionally, procedures for simulation of periodic processes
and real data sets are included.
Hurd, H. L., Miamee, A. G. (2007) <doi:10.1002/9780470182833>
Box, G. E. P., Jenkins, G. M., Reinsel, G. (1994) <doi:10.1111/jtsa.12194>
Brockwell, P. J., Davis, R. A. (1991, ISBN:978-1-4419-0319-8)
Bretz, F., Hothorn, T., Westfall, P. (2010, ISBN: 9780429139543)
Westfall, P. H., Young, S. S. (1993, ISBN:978-0-471-55761-6)
Bloomfield, P., Hurd, H. L.,Lund, R. (1994)
<doi:10.1111/j.1467-9892.1994.tb00181.x>
Dehay, D., Hurd, H. L. (1994, ISBN:0-7803-1023-3)
Vecchia, A. (1985) <doi:10.1080/00401706.1985.10488076>
Vecchia, A. (1985) <doi:10.1111/j.1752-1688.1985.tb00167.x>
Jones, R., Brelsford, W. (1967) <doi:10.1093/biomet/54.3-4.403>
Makagon, A. (1999)
<https://www.math.uni.wroc.pl/~pms/files/19.2/Article/19.2.5.pdf>
Sakai, H. (1989) <doi:10.1111/j.1467-9892.1991.tb00069.x>
Gladyshev, E. G. (1961)
<https:
//www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=dan&paperid=24851>
Ansley (1979) <doi:10.1093/biomet/66.1.59>
Hurd, H. L., Gerr, N. L. (1991) <doi:10.1111/j.1467-9892.1991.tb00088.x>.

License GPL (>= 2.0)

LazyData TRUE

Imports corpcor, gnm, matlab, Matrix, signal, stats

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2023-11-17 08:20:02 UTC

Author Anna Dudek [aut],
Harry Hurd [aut],
Wioletta Wojtowicz [aut],
Karolina Marek [cre]

Contents

`perARMA-package` *Periodic Time Series Analysis and Modeling*

Description

This package provides procedures for periodic time series analysis. The package includes procedures for nonparametric spectral analysis and parametric (PARMA) identification, estimation/fitting and prediction. The package is equipped with three examples of periodic time series: `volumes` and `volumes.sep`, which record hourly volumes of traded energy, and `arosa` containing monthly ozone levels.

Details

Package:	perARMA
Type:	Package
Version:	1.6
Date:	2016-02-25
License:	GPL(>=2.0)
LazyLoad:	yes

The main routines are:
Nonparametric spectral analysis: `pgram`, `scoh`
Preliminary identification and conditioning: `permest`, `persigest`
Identification: `peracf`, `Bcoeff`, `perpacf`, `acfpacf`
Parameter estimation/fitting: `perYW`, `loglikec`, `parmaf`, `loglikef`
Prediction: `predictperYW`, `predseries`
Simulation and testing: `makeparma`, `parma_ident`

For a complete list of procedures use `library(help="perARMA")`.

Author(s)

Anna Dudek, Harry Hurd and Wioletta Wojtowicz
Maintainer: Karolina Marek <karolina.marek10@gmail.com>

References

Hurd, H. L., Miamee, A. G., (2007), Periodically Correlated Random Sequences: Spectral Theory and Practice, Wiley InterScience.

See Also

Packages for Periodic Autoregression Analysis `link{pear}`, Dynamic Systems Estimation `link{dse}` and Bayesian and Likelihood Analysis of Dynamic Linear Models `link{dlm}`.

Examples

```
## Do not run
## It could take more than one minute
#demo(perARMA)
```

`ab2phth` *Fourier representation of real matrix*

Description

The function `ab2phth` transforms an input matrix `a` of size $T \times p$ containing the sine and cosine coefficients in the real Fourier series representation, to the $T \times p$ output matrix `phi` according to $\phi_{n,j} = a_{1,j} + \sum_{k=1}^{\lfloor T/2 \rfloor} (a_{2k,j} \cos(2\pi kn/T) + a_{2k+1,j} \sin(2\pi kn/T))$ for $n = 1, \ldots, T$ and $j = 1, \ldots, p$. The inverse transformation is implemented in `phth2ab` function.

Usage

```
ab2phth(a)
phth2ab(phi)
```

Arguments

a	matrix of $a_{n,j}$ coefficients (size of $T \times p$).
phi	matrix of $\phi_{n,j}$ coefficients (size of $T \times p$).

Value

martix phi or a for ab2phth or phth2ab, respectively.

Author(s)

Harry Hurd

See Also

makepar, makeparma, parma_ident

Examples

```
m=matrix(seq(0,11),3,4)
ab<-ab2phth(m)
phi=ab$phi
phth2ab(phi)
```

---

acfpacf *Plotting usual ACF and PACF*

Description

Plots values of usual ACF and PACF functions with confidence intervals. Function acfpacf uses procedures acfpacf.acf and acfpacf.pacf, which computes values of ACF and PACF function, respectively.

Usage

```
acfpacf(x, nac, npac, datastr,...)
acfpacf.acf(x, normflg)
acfpacf.pacf(x, m)
```

Arguments

`x`	input time series, missing values are not permitted.
`nac`	number of ACF values to return (typically much less than length of `x`).
`npac`	number of PACF values to return (typically much less than length of `x`).
`datastr`	string name of data to be printed on the plot.
`normflg`	computing parameter for ACF values. These values are returned as a series `acf(k)` for `k = 0, ..., nr`, where `nr` is length of `x`. Parameter `normflg` can be equal to: 0 - `acf(k)` values will be normalized by `nr`, 1 - `acf(k)` values will be normalized by `nr` multiplied by sample variance (to obtain correlations), 2 - `acf(k)` values will be divided by `nr-k`, thus normalized by number of samples at each lag, 3 - `acf(k)` values will be divided by `nr-k` multiplied by sample variance. In `acfpacf` procedure `normflg=3` is used.
`m`	maximum lag to compute PACF values. Value for the first lag (`pacf(1)`) is equal to `acf(2)`, which is the lag 1 acf value, and then for computing values for `k = 2, ..., m` the Toeplitz structure of the projection equations is used (see Brockwell, P. J., Davis, R. A., 1991, Time Series: Theory and Methods, example 4.4.2).
`...`	other arguments: `plfg`, `acalpha`, `pacacalpha`, `valcol`, `thrcol`, `thrmhcol`, where `plfg` is plotting flag, this parameter should be positive number to plot computed `acfpacf` values, `acalpha` and `pacalpha` are p-values (or `alpha` is I type error) thresholds for ACF and PACF plots based on independent normal values, `valcol,thrcol,thrmhcol` are colors of function values, confidence interval markers on the ACF/PACF and confidence interval markers on the ACF/PACF for multiple hypothesis alpha correction on the plot. By default parameters are fixed to `plfg=1`, `acalpha=.05`, `pacacalpha=.05`, `valcol="red"`, `thrcol="green"`, `thrmhcol="blue"`.

Details

Function `acfpacf` returns plot of ACF and PACF values with two types of thresholds for input `acalpha` and `pacalpha`, respectively. The first one denoted by `thr` is given for ACF values by $Pr[acf(j) > thr] = \alpha/2$ and $Pr[acf(j) < -thr] = \alpha/2$ where `acf(k)` is the ACF value at lag `k`. This threshold corresponds to type I error for null hypothesis that `acf(k) = 0`. The plot allows to check if any of the ACF values are significantly non-zero. Actual threshold calculations are based on the following asymptotic result: if X_t is $IID(0, \sigma^2)$, then for large n, $\hat{\rho}(k)$ for $k << n$ are $IIDN(0, 1/n)$ (see Brockwell, P. J., Davis, R. A., 1991, Time Series: Theory and Methods, example 7.2.1, p. 222). Thus, under the null hypothesis, the plots for `thr = qnorm(1-acalpha/2,0,1/sqrt(nr))` should exhibit a proportion of roughly `acalpha` points that lie outside the interval `[-thr, thr]`. Threshold for PACF is based on the same results.
On the other hand we can also interpret the plots as a multiple hypothesis testing problem to compute second threshold `thrm`. Suppose, we decided to plot some number of nonzero lags (equal to `nac`) of the ACF function. If the estimated `acf` values estimates are IID then we have `nac` independent tests of `acf(k) = 0`. The probability that at least one of values lies outside the interval

[-thr, thr] is equal to 1-Pr[all lie inside], which is [1-(1-acalpha)]^nac. Finally, the last expression is approximately equal to nac*acalpha. To get a threshold thrmh such that 1-Pr[all lie inside] = acalpha we take the threshold as
thrmh = qnorm (1-(acalpha/2)/nac, 0, 1/sqrt(nr)) (for more details check the Bonferroni correction).
For the PACF, the threshold thrm calculation is based on Theorem 8.1.2 of Time Series: Theory and Methods, p. 241, which states that the PACF values for an AR sequence are asymptotically normal.

Value

No return value, called for side effects

Note

Procedure acfpacf is an implementation of the usual (meaning not periodic) ACF and PACF functions. It happens that for PC time series the usual ACF and PACF are still useful in the identification of model orders and in some cases can be more sensitive than the perodic versions. The ACF and PACF values inform about the correlations of the random part. It is possible to run acfpacf on original data which include the peridic mean as a deterministic component. But typically the periodic mean can distort our understanding (or view) of the random fluctuations, thus running acfpacf additionally on the data after removing periodic mean is suggested. It is possible to use also acfpacf.acf, acfpacf.pacf procedures to obtain values of ACF and PACF functions, respectively. When subjected to a truly PC sequence, the usual ACF and PACF produce an average of the instantaneous (time indexed) values produced by periodic ACF and periodic PACF. Depending therefore on correlations between these averaged quantities, the usual ACF and PACF may be more or less sensitive that the instantaneous ones.

Author(s)

Harry Hurd

References

Box, G. E. P., Jenkins, G. M., Reinsel, G. (1994), Time Series Analysis, 3rd Ed., Prentice-Hall, Englewood Cliffs, NJ.

Brockwell, P. J., Davis, R. A. (1991), Time Series: Theory and Methods, 2nd Ed., Springer: New York.

Bretz, F., Hothorn, T., Westfall, P. (2010), Multiple Comparisons Using R, CRC Press.

Westfall, P. H., Young, S. S. (1993), Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment, Wiley Series in Probability and Statistics.

See Also

peracf, perpacf

Examples

```
data(volumes)
# for original data
dev.set(which=1)
acfpacf(volumes,24,24,'volumes')

# for data after removing periodic mean
pmean_out<-permest(t(volumes),24, 0.05, NaN,'volumes',pp=0)
xd=pmean_out$xd
dev.set(which=1)
acfpacf(xd,24,24,'volumes')
```

---

arosa *Monthly stratospheric ozone, Arosa*

Description

A fifty-year time series of monthly stratospheric ozone readings from Arosa, Switzerland. The dataset length is equal to 684, but some of the observations are missing (denoted by NaNs).

Usage

```
data(arosa)
```

Format

The format is: Time-Series [1:684] from 1 to 684: NaN NaN NaN NaN NaN NaN 312 300 281 267 ...

References

Bloomfield, P., Hurd, H. L.,Lund, R., (1994), Periodic correlation in stratospheric ozone data. Journal of Time Series Analysis 15, 127-150.

Examples

```
data(arosa)
str(arosa)
```

`Bcoeff, Bcoeffa` *Fourier representation of covariance function*

Description

The procedure `Bcoeff` computes the complex estimators $B_k(\tau) = \frac{1}{T}\sum_{t=0}^{T-1} R(t+\tau, t)\exp(-i2\pi t/T)$ as Fourier coefficients in covariance function representation. The procedure first computes the periodic mean (with missing values ignored) and subtracts it from the series. For each specified lag τ, the values of $\hat{B}_k(\tau)$ are computed only for $k = 0, 1, \ldots, \lfloor T/2 \rfloor$, because for real series $\hat{B}_k(\tau) = \overline{\hat{B}_{T-k}(\tau)}$. Also the p-values for the test $B_k(\tau) = 0$ are returned.
The function `Bcoeffa` calculates the estimator of the real coefficients $a_k(\tau)$ which satisfy $R(t + \tau, t) = B(t, \tau) = a_1(\tau) + \sum(a_{2k}(\tau)\cos(2\pi kt/T) + a_{2k+1}(\tau)\sin(2\pi kt/T))$, for all required lags τ and k.

Usage

```
Bcoeff(x, T_t, tau, missval, datastr,...)
Bcoeffa(x, T_t, tau, missval, datastr,...)
```

Arguments

`x`	input time series.
`T_t`	period length of PC-T structure.
`tau`	vector of lag values on which estimation of $B_k(\tau)$ is performed.
`missval`	notation for missing values.
`datastr`	string name of data for printing.
`...`	other arguments: `printflg` should be a positive parameter to print, `meth` is a parameter connected to the amount of frequencies used in estimation, if `meth=0` all possible frequencies are used in estimation else if `meth > 0` then $\lfloor n/2 \rfloor$ frequencies on either side of the Fourier frequencies $2\pi k/T$ are used. By default parameters are fixed to `printflg=1, meth=0`.

Details

This procedure can be applied to the original series to obtain estimators of $B_k(\tau)$ in covariance function representation or to the normalized series (series after periodic mean removal and division by standard deviations) to obtain correlation coefficients. The p-values for test of $|B_k(\tau)|^2 = 0$ are based on the ratio of magnitude squares of amplitudes of a high resolution Fourier decompositions. Magnitudes for the frequency corresponding to index k are compared to the magnitudes of neighboring frequencies (via the F distribution) (see Hurd, H. L., Miamee, A. G., 2007, Periodically Correlated Random Sequences, pp. 272-282, 288-292).

Value

procedures (for positive `printflg` parameter) print a table containing the following columns:

- `k` index number of the coefficient $B_k(\tau)$.
- `reB_k`, `imB_k/ahat_k` real and imaginary parts of estimated coefficient $B_k(\tau)$ (`Bcoeff` procedure) / real coefficients in representation of coefficient $B_k(\tau)$ (`Bcoeffa` procedure).
- `n1` degrees of freedom associated to the estimator $SS1/n1$ of the variance at frequency $2\pi k/T$.
- `n2` degrees of freedom associated to the "background" variance estimation $SS2/n2$ based on frequencies in the neighborhood of the frequency $2\pi k/T$.
- `Fratio` the statistic $(SS1/n1)/(SS2/n2)$, which under the null hypothesis has $F(n1, n2)$ distribution.
- `pv` p-values for test of $|B_k(\tau)|^2 = 0$, based on F distribution.

If `printflg` is set to be equal to 0, above values are returned just as matrices.

Author(s)

Harry Hurd

References

Dehay, D., Hurd, H. L., (1994), Representation and Estimation for Periodically and Almost Periodically Correlated Random Processes in W. A. Gardner (ed.), Cyclostationarity in Communications and Signal Processing, IEEE Press.

Hurd, H. L., Miamee, A. G., (2007), Periodically Correlated Random Sequences: Spectral Theory and Practice, Wiley InterScience.

Examples

```
data(volumes)
Bcoeff(volumes,24,seq(0,12),NaN,'volumes')
Bcoeffa(volumes,24,seq(0,12),NaN,'volumes')
```

`loglikec` *Calculation of the logarithm of likelihood function*

Description

Function `loglikec`, given `phi`, `del`, `theta` encoded in `ptvec`, evaluates the logarithm of likelihood function from the PARMA series. Procedure returns also values of the AIC, FPE, BIC information criteria and MSE of residuals, what enables to examine residuals and evaluate godness of model fit.

Usage

```
loglikec(ptvec, x, conpars)
```

Arguments

- `ptvec` — vector of parameters for PARMA(p,q) model, containing matrix parameters `phi` (of size $T \times p$), `del` (of size $T \times 1$), `theta` (of size $T \times q$) as following `ptvec = [phi[,1],...,phi[,p],del,theta[,1], ...,theta[,q]]`.
- `x` — input time series.
- `conpars` — vector of parameters `[T, p, q, stype]`,
  `T_t` period of PC-T structure,
  `p, q` maximum PAR and PMA order, respectively,
  `stype` numeric parameter connected with covariance matrix computation, so far should be equal to 0 to use procedure `R_w_ma` (see `R_w_ma` description). In the future also other values of `stype` will be available for full covariance matrix computation.

Details

In this procedure first series `x` is filtered by matrix coefficients `phi`, `del`, `theta`. The code to compute logarithm of likelihood function must includes the computation of covariance matrix from the parameters `phi`, `del`, `theta`. Since the inverse of the computed covariance is needed for computing the likelihood, and it is sometimes ill conditioned (or even singular), the condition is improved by removing rows and columns corresponding to very small eigenvalues. This corresponds to removing input data that is highly linearly dependent on the remaining input data. The procedure contains a threshold ZTHRS (which current value is `10*eps`) that governs the discarding of rows and column corresponding to small eigenvalues (these are determined by a Cholesky decomposition). Any eigenvalue smaller than the threshold has its row and column deleted from the matrix. Then the inverse and the likelihood are computed from the reduced rank covariance matrix.

Value

list of values:

- `loglik` — logarithm of likehood function (nagative value).
- `aicval` — value of AIC criterion.
- `fpeval` — value of FPE criterion.
- `bicval` — value of BIC criterion.

Note

In the `loglikec` procedure, motivated by the possibility of deficient rank sequences, we made a variant of the Cholesky decomposition. In proposed approach upper traingular matrix eliminates data points that are lineary dependant on previous ones and removes their consideration in the likelihood value calculation. As a consequence data vector is reduced so that covariance matrix is positive definite and problem of non-invertible covariance matrix is avoided.

Author(s)

Harry Hurd

References

Box, G. E. P., Jenkins, G. M., Reinsel, G. (1994), Time Series Analysis, 3rd Ed., Prentice-Hall, Englewood Cliffs, NJ.

Brockwell, P. J., Davis, R. A. (1991), Time Series: Theory and Methods, 2nd Ed., Springer: New York.

Vecchia, A., (1985), Maximum Likelihood Estimation for Periodic Autoregressive Moving Average Models, Technometrics, v. 27, pp.375-384.

Vecchia, A., (1985), Periodic autoregressive-moving average (PARMA) modeling with applications to water resources, Water Resources Bulletin, v. 21, no. 5.

See Also

R_w_ma, parmafil

Examples

```
## Do not run
## It could take a few seconds

data(volumes)
pmean<-permest(t(volumes),24, 0.05, NaN,'volumes', pp=0)
xd=pmean$xd
estimators<-perYW(volumes,24,2,NaN)
estvec=c(estimators$phi[,1],estimators$phi[,2],estimators$del)
loglikec(estvec,xd,c(24,2,0,0))
```

`loglikef` *Calculation of the logarithm of likelihood function (using Fourier representation)*

Description

Procedure `loglikef` computes the logarithm of likelihood function from the PARMA sequence `x` for matrices `phi` (of size $T \times p$) and `theta` (of size $T \times (q+1)$) inputed in their Fourier representation as `a` and `b`, respectively.

Usage

```
loglikef(ab, x, conpars)
```

Arguments

ab
: matrix [a,b] taken as a vector, where
  a is Fourier representation of phi (use phi=ab2phth(a) to recover phi),
  b is Fourier representation of theta (use del=ab2phth(b[,1]) to recover del and theta=ab2phth(b[,2:q+1]) to recover theta).
  Vector ab contains only non-zero coefficients form a and b.

x
: input time series.

conpars
: vector of parameters [T,p,q,naf,nbf,del_mask,iaf,ibf,stype]:
  T_t period of PC-T structure,
  p, q maximum PAR and PMA order, respectively,
  naf, nbf total active coefficients in a and b, respectively,
  del_mask vector of length T (it will be used in the future, so far the user should set del_mask=matrix(1,T,1)),
  iaf, ibf linear indexes of active coefficients in a and b, respectively,
  stype numeric parameter connected with covariance matrix computation, so far should be equal to 0 to use procedure R_w_ma (see R_w_ma description). In the future also other values of stype will be available for full covariance matrix computation.

Details

This method of computation of logarithm of likelihood function makes use of the representation of the periodically varying parameters by Fourier series. This alternative parametrization of PARMA system, introduced by Jones and Breslford, can sometimes substantially reduce the number of parameters required to represent PARMA system. Mapping between phi and theta coefficients and a and b coefficients is one-to-one, so first logarithm of likelihood is computed for transformed coefficients and then these coefficients are transformed to phi and theta. Fourier series parametrization permits us to reduce the total number of parameters by constraining some frequencies to have zero amplitude. Then the code includes the computation of covariance matrix from the parameters phi, del, theta. Since the inverse of the computed covariance is needed for computing the likelihood, and it is sometimes ill conditioned (or even singular), the condition is improved by removing rows and columns corresponding to very small eigenvalues. This corresponds to removing input data that is highly linearly dependent on the remaining input data. The procedure contains a threshold ZTHRS (which current value is 10*eps) that governs the discarding of rows and column corresponding to small eigenvalues (these are determined by a Cholesky decomposition). Any eigenvalue smaller than the threshold has its row and column deleted from the matrix. Then the inverse and the likelihood are computed from the reduced rank covariance matrix.

Value

negative value of the logarithm of likelihood function: y.

Note

In the loglikef procedure, motivated by the possibility of deficient rank sequences, we made a variant of the Cholesky decomposition. In proposed approach upper traingular matrix eliminates data points that are lineary dependant on previous ones and removes their consideration in the likelihood value calculation. As a consequence data vector is reduced so that covariance matrix is

positive definite and problem of non-invertible covariance matrix is avoided.
This function is used in `parmaf` procedure, thus for more details please look also at `parmaf` code.

Author(s)

Harry Hurd

References

Box, G. E. P., Jenkins, G. M., Reinsel, G. (1994), Time Series Analysis, 3rd Ed., Prentice-Hall, Englewood Cliffs, NJ.

Brockwell, P. J., Davis, R. A., (1991), Time Series: Theory and Methods, 2nd Ed., Springer: New York.

Jones, R., Brelsford, W., (1967), Time series with periodic structure, Biometrika 54, 403-408.

Makagon, A., (1999), Theoretical prediction of periodically correlated sequences, Probability and Mathematical Statistics 19 (2), 287-322.

Sakai, H., (1989), On the spectral density matrix of a periodic ARMA process, J. Time Series Analysis, v. 12, no. 2, pp. 73-82.

Vecchia, A., (1985), Maximum Likelihood Estimation for Periodic Autoregressive Moving Average Models, Technometrics, v. 27, pp.375-384.

See Also

`R_w_ma`, `parmaresid`, `parmaf`

`makeparma` *Simulation of PARMA sequence*

Description

Procedures `makeparma` and `makepar` enable to construct PARMA and PAR type sequence of length `n` according to inputed matrices `phi`, `theta`, `del`. The optional parameter `nprep` defines the number of periods of simulated output `y` that will be discarded to let the start-up transients settle.

Usage

```
makeparma(n, phi, theta, del, nprep)
makepar(n, phi, del, nprep)
```

Arguments

- `n` — length of simulated series.
- `phi` — matrix of size $T \times p$ containing periodic AR parameters.
- `theta` — matrix of size $T \times q$ containing periodic MA parameters.
- `del` — vector of length T containing the periodic sigmas (shock weights), which are sometimes denoted also as $\sigma(t)$ or as $\theta_0(t)$.
- `nprep` — number of periods of simulated output; for `p>0` the computation of new `y` values depends on old ones through the autoregressive part. Starting a PARMA with the assumption that old values of `y` are equal to zero causes a transient in the output. This transient dies out as time goes on. So to avoid this transient problem, we compute `nprep` periods of simulated output to allow the transient to die out.
  This parameter is optional, because by default `nprep=50` (for `makeparma` procedure) or `nprep=10` (for `makepar` procedure). Later we will provide a function to compute an appropriate value of `nprep` that depends on the PARMA parameters.

Details

A vector $\xi(t)$ of independent $N(0,1)$ variates is generated by the standard random number generator `rnorm`. This vector series is filtered by `parmafil`, which parameters are set by `phi`, `theta` and `del`, to generate the filtered series (pre-iterates and the `n` desired data). The last `n` data of the filtered series are output in the vector `y`.

Value

PARMA or PAR sequence returned as `y`.

Author(s)

Harry Hurd

See Also

`parmafil`

Examples

```
##################### simulation of PARMA(2,1)
T=12
nlen=480
p=2
 a=matrix(0,T,p)
q=1
 b=matrix(0,T,q)

a[1,1]=.8
a[2,1]=.3
phia<-ab2phth(a)
phi0=phia$phi
```

```
phi0=as.matrix(phi0)

b[1,1]=-.7
b[2,1]=-.6
thetab<-ab2phth(b)
theta0=thetab$phi
theta0=as.matrix(theta0)

del0=matrix(1,T,1)

PARMA21<-makeparma(nlen,phi0,theta0,del0)
parma<-PARMA21$y
plot(ts(parma))

##################### simulation of PAR(2)
T=24
nlen=1000
p=2
a=matrix(0,T,p)
a[1,1]=.5
a[2,2]=.4

phia<-ab2phth(a)
phi0=phia$phi
phi0=as.matrix(phi0)

del0=matrix(1,T,1)

PAR1<-makepar(nlen,phi0,del0)
par<-PAR1$y
plot(ts(par))
```

---

parmaf *PARMA coefficients estimation*

Description

Procedure `parmaf` enables the estimation of parameters of the chosen representation of PARMA(p,q) model. For general PARMA we use non-linear optimization methods to obtain minimum of negative logarithm of likelihood function using `loglikef` procedure. Intitial values of parameters are computed using Yule-Walker equations.

Usage

```
parmaf(x, T_t, p, q, af, bf, ...)
```

Arguments

`x` input time series.

T_t	period length of PC-T structure.
p	maximum PAR order, which is a number of columns in af.
q	maximum PMA order, which is a number of columns in bf diminished by 1.
af	$T \times p$ logical values matrix pointing to active frequency components for phi.
bf	$T \times (q+1)$ logical matrix pointing to active frequency components for theta.
...	Other arguments: a0 starting value for a, where a is Fourier representation of phi (use phi=ab2phth(a) to recover phi); if a0 is not defined Yule Walker method is used to estimate it; b0 starting values for b, where b is Fourier representation of theta (use del=ab2phth(b[,1]) to recover del and use theta = ab2phth(b[,2:q+1]) to recover theta); if b0 is not defined Yule Walker method is used to estimate it; stype numeric parameter connected with covariance matrix computation, so far should be equal to 0 to use procedure R_w_ma (see R_w_ma description). In the future also other values of stype will be available for full covariance matrix computation.

Details

In order to obtain maximum likelihood estimates of model parameters a and b we use a numerical optimization method to minimalize value of y (as negative value of logarithm of loglikelihood function returned by loglikef) over parameter values. Internally, parameters a and b are converted to phi and theta as needed via function ab2phth. For the present we use optim function with defined method="BFGS" (see code for more details).

Value

list of values:

a	is matrix of Fourier coefficients determining phi.
b	is matrix of Fourier corfficients determining theta.
negloglik	minimum value of negative logarithm of likehood function.
aicval	value of AIC criterion.
fpeval	value of FPE criterion.
bicval	value of BIC criterion.
resids	series of residuals.

Author(s)

Harry Hurd

References

Box, G. E. P., Jenkins, G. M., Reinsel, G. (1994), Time Series Analysis, 3rd Ed., Prentice-Hall, Englewood Cliffs, NJ.

Brockwell, P. J., Davis, R. A., (1991), Time Series: Theory and Methods, 2nd Ed., Springer: New

York.

Jones, R., Brelsford, W., (1967), Time series with periodic structure, Biometrika 54, 403-408.

Vecchia, A., (1985), Maximum Likelihood Estimation for Periodic Autoregressive Moving Average Models, Technometrics, v. 27, pp.375-384.

See Also

loglikef, perYW, R_w_ma,

Examples

```
######## simulation of periodic series
T=12
nlen=480
p=2
 a=matrix(0,T,p)
q=1
 b=matrix(0,T,q)
a[1,1]=.8
a[2,1]=.3

a[1,2]=-.9
phia<-ab2phth(a)
phi0=phia$phi
phi0=as.matrix(phi0)

b[1,1]=-.7
b[2,1]=-.6
thetab<-ab2phth(b)
theta0=thetab$phi
theta0=as.matrix(theta0)
del0=matrix(1,T,1)
makeparma_out<-makeparma(nlen,phi0,theta0,del0)
y=makeparma_out$y

## Do not run
## It could take more than one minute


############ fitting of PARMA(0,1) model
p=0
q=1
af=matrix(0,T,p)
bf=matrix(0,T,q+1)
bf[1,1]=1
bf[1:3,2]=1

parmaf(y,T,p,q,af,bf)

########### fitting of PARMA(2,0) model
```

```
p=2
q=0
af=matrix(0,T,p)
bf=matrix(0,T,q+1)
af[1:3,1]=1
af[1:3,2]=1
bf[1,1]=1
parmaf(y,T,p,q,af,bf)
############ fitting of PARMA(2,1) model
p=2
q=1
af=matrix(0,T,p)
bf=matrix(0,T,q+1)
af[1:3,1]=1
af[1:3,2]=1
bf[1,1]=1
bf[1:3,2]=1
parmaf(y,T,p,q,af,bf)
```

`parmafil` *PARMA filtration*

Description

Procedure `parmafil` filters the vector `x` according to matrices `a, b` containing PARMA model parameters. The function returns series `y` such that $a(n,1) * y(n) = b(n,1) * x(n) + b(n,2) * x(n-1) + \ldots + b(n, nb+1) * x(n-nb) - a(n,2) * y(n-1) - \ldots - a(n, na+1) * y(n-na)$.

Usage

```
parmafil(b, a, x)
```

Arguments

`b`	matrix of size $T \times (nb+1)$, which elements satisfy $b(n,j) = b(n+T,j)$, usually in the literature `b` is called the periodic MA parameters and nb is denoted by q.
`a`	matrix of size $T \times na$, which elements satisfy $a(n,j) = a(n+T,j)$, usually in the literature `a` is called the periodic AR parameters and na is denoted p. If $a(n,1)$ is not equal to 1 for all n, the values of $a(n,j)$ are normalized by $a(n,j) = a(n,j)/a(n,1)$.
`x`	input time series.

Value

Filtered signal `y`.

Note

To filter using the convention $\phi(t,B)x(t) = \theta(t,B)\xi(t)$ with $\phi(t,B) = 1 - \phi(t,1)B - ... - \phi(t,p)B^p$, $\theta(t,B) = del(t,1) + \theta(t,1)B + ... + \theta(t,q)B^q$ set `a=[ones(T,1),-phi]`, `b=[theta]`, then `x=parmafil(b,a,xi)`.

Author(s)

Harry Hurd

See Also

`loglikec`, `loglikef`, `makeparma`

Examples

```
b=matrix(c(1,1,0,0,.5,.5),2,3)
a=matrix(c(1,1,.5,.5),2,2)
s=sample(1:100,50, replace=TRUE)
x=matrix(s,50,1)

parmafil_out<-parmafil(a,b,x)
y=parmafil_out$y
plot(y,type="l")
```

`parmaresid` *Computing residuals of PARMA series*

Description

Procedure `parmaresid`, given `phi` (of size $T \times p$), `del` (of size $T \times 1$), `theta` (of size $T \times q$), computes the residuals of PARMA series.

Usage

```
parmaresid(x, stype, del, phi,...)
```

Arguments

`x`	input time series.
`stype`	numeric parameter connected with covariance matrix computation, so far should be equal to 0 to use procedure `R_w_ma` (see `R_w_ma` description). In the future also other values of `stype` will be available for full covariance matrix computation.
`del`	vector of coefficients of length T.
`phi`	matrix of coefficients of size $T \times p$.
`...`	matrix of coefficients `theta` of size $T \times q$.

Details

This program uses parmafil to filter the series and computes the covariance matrix. This code does the Cholesky factorization and determines the residuals from the inverse of L (see the code: e=Linv*w0_r1). This allows the treatment of a deficient rank covariance and a reduction of rank. Procedure parmaresid is used in parmaf function.

Value

Series of residuals resids.

Author(s)

Harry Hurd

References

Box, G. E. P., Jenkins, G. M., Reinsel, G. (1994), Time Series Analysis, 3rd Ed., Prentice-Hall, Englewood Cliffs, NJ.

Brockwell, P. J., Davis, R. A. (1991), Time Series: Theory and Methods, 2nd Ed., Springer: New York.

Vecchia, A., (1985), Maximum Likelihood Estimation for Periodic Autoregressive Moving Average Models, Technometrics, v. 27, pp.375-384.

See Also

R_w_ma, loglikec, loglikef

Examples

```
## Do not run
## It could take a few seconds

data(volumes)
pmean<-permest(t(volumes),24, 0.05, NaN,'volumes', pp=0)
xd=pmean$xd
estimators<-perYW(volumes,24,2,NaN)

parmaresid(xd, 0, estimators$del, estimators$phi)
```

parma_ident — *Identification of PC-T structure*

Description

Procedure parma_ident utilizes a collection of procedures (functions) that together provide identification of PC structure in the series and saves results in the 'ident.txt' file, which is located in the working directory. This procedure could be applied to the original time series x or to the residuals of fitted PARMA models to characterize the goodness of fit.

Usage

```
parma_ident(x, T_t, missval, datastr, ...)
```

Arguments

x	input time series.
T_t	period of PC-T structure.
missval	notation for missing values.
datastr	string name of data for printing.
...	other arguments: outdir is string name of the directory in which file 'ident.txt' with results returned by parma_ident procedure will be saved, details should be equal to 1 to print all details. By default these parameters are fixed to outdir='IDENT_OUT', details=1.

Details

Procedure parma_ident provides a universal method for analyzing series or residuals. It calls following procedures: permest, persigest, peracf, Bcoeff, Bcoeffa, perpacf, ppfcoeffab, ppfplot, acfpacf.

Value

procedure returns list of values:

pmean	periodic mean values,
xd	series after removing periodic mean,
pstd	periodic standard deviations values,
xn	series obtained after removing periodic mean and divided by periodic standard deviations,

as well as a text file 'ident.txt' containing all the textual output generated in the running of parma_ident.

Author(s)

Harry Hurd

References

Hurd, H. L., Miamee, A. G., (2007), Periodically Correlated Random Sequences: Spectral Theory and Practice, Wiley InterScience.

Examples

```
############### PC-T series simulation
T=12
nlen=480
descriptor='PARMA(2,1) periodic phis all del =1'
p=2
 a=matrix(0,T,p)
q=1
 b=matrix(0,T,q)
a[1,1]=.8
a[2,1]=.3
a[1,2]=-.9

phia<-ab2phth(a)
phi0=phia$phi
phi0=as.matrix(phi0)

b[1,1]=-.7
b[2,1]=-.6
thetab<-ab2phth(b)
theta0=thetab$phi
theta0=as.matrix(theta0)
del0=matrix(1,T,1)

makeparma_out<-makeparma(nlen,phi0,theta0,del0)
y=makeparma_out$y

############### parma_ident use

parma_ident(t(y),T,NaN,descriptor,outdir=tempdir())
```

---

peracf *Periodic ACF function*

Description

Function `peracf`, given an input time series and a specified period `T`, computes the periodic correlation coefficients for which $\rho(t+\tau,t)=\rho(t,\tau)$, where $t=1,\ldots,T$ are seasons and τ is lag. For each possible pair of t and τ confidence limits for $\rho(t,\tau)$ are also computed using Fisher transformation. Procedure `peracf` provides also two important tests: $\rho(t+\tau,t)\equiv\rho(\tau)$ and $\rho(t+\tau,t)\equiv 0$.

Usage

```
peracf(x, T_t, tau, missval, datastr,...)
```

Arguments

x	input time series, at the begining missing values in x will be treat as zeros and periodic mean will be computed, then missing values will be replaced by periodic mean.
T_t	period of PC-T structure.
tau	vector of lag values for which estimation is made.
missval	notation for missing values (denoted as NaN).
datastr	string name of data for printing.
...	other arguments, that are connected with the plots: prttaus, plottaus, cialpha, typeci, typerho, pchci, pchrho, colci, colrho, where prttaus is a set of lags for which correlation coefficients are printed; it is a subset of tau, plottaus is a set of lags for plotting the correlation coefficients (one plot per lag); it is a subset of tau, cialpha threshold for confidence interval, typeci/ typerho, pchci/ pchrho, colci/colrho define the type, plot character and colors of confidence intervals/periodic correlation values. By default these parameters are fixed to prttaus = seq(1,T/2), plottaus = seq(1,T/2), cialpha = 0.05, typeci = "b", typerho = "b", pchci = 10, pchrho = 15, colci = "blue", colrho = "red".

Details

Function peracf uses three separate procedures:
rhoci() returns the upper and lower bands defining a $1-\alpha$ confidence interval for the true values of $\rho(t,\tau)$,
rho.zero.test() tests whether the estimated correlation coefficients are equal to zeros, $\rho(t+\tau,t)\equiv 0$.
rho.equal.test() tests whether the estimated correlation coefficients are equal to each other for all seasons in the period, $\rho(t+\tau,t)\equiv\rho(\tau)$.

In the test $\rho(t+\tau,t)\equiv\rho(\tau)$, rejection for some $\tau>0$ indicates that series is properly PC and is not just an amplitude modulated stationary sequence. In other words, there exists a nonzero lag τ for which $\rho(t+\tau,t)$ is properly periodic in t.
In the test $\rho(t+\tau,t)\equiv 0$, the rejection for some $\tau\neq 0$ indicates the sequence is not PC white noise.

Value

tables of values for each specified lag τ:

rho(t, tau)	estimated correlation coefficients.

lower	lower bands of confidence intervals.
upper	upper bands of confidence intervals.
nsamp	number of samples used in each estimation.

Above values are also returned as matrices.

Author(s)

Harry Hurd

References

Hurd, H. L., Miamee, A. G., (2007), Periodically Correlated Random Sequences: Spectral Theory and Practice, Wiley InterScience.

See Also

Bcoeff, perpacf

Examples

```
data(volumes)
dev.set(which=1)
peracf(t(volumes),24,seq(1,12),NaN,'volumes')
```

---

permest *Periodic Mean Estimation*

Description

Assuming that the period T is known, procedure permest plots and returns the estimated periodic mean as a function of season. Missing data are permitted. The confidence intervals for these values, based on the t-distribution, are also computed and plotted. The de-meaned x is also returned with missing values replaced by periodic mean values. If at time t there is a missing value, it is replaced with the periodic mean at (t mod T), provided the periodic mean exists (meaning there is at least one non-missing data for the season (t mod T)). Otherwise the periodic mean at (t mod T) will be set to "Missing" and in the output vectors xr and xd all the values whose times are congruent with (t mod T) will be set to "Missing".

Usage

```
permest(x, T_t, alpha, missval, datastr,...)
```

Arguments

x	input time series.
T_t	period of the computed mean.
alpha	1-alpha is confidence interval containment probability using the t-distribution.
missval	notation for missing values.
datastr	string name of data for printing.
...	other arguments used in the plot: typeci, typepmean, pchci, pchpmean, colci, colpmean, pp; typeci / typepmean, pchci / pchpmean, colci / colpmean set the type, plot character and colors of confidence intervals / periodic mean values on the plot, pp should be positive to print and plot permest values. By default these parameters are fixed to typeci = "o", typepmean = "b", pchci = 10, pchpmean = 15, colci = "red", colpmean = "blue", pp = 1.

Details

The series may contain missing values (we suggest using NaN) and the length of the series need not be an integer multiple of the period. The program returns and plots the periodic mean with 1-alpha confidence intervals based on all non-missing values present for each particular season. The p-value for a one-way ANOVA test for equality of seasonal means is also computed.

Value

procedure returns:

pmean	periodic mean values.
lower, upper	bounds of the confidence intervals.
xr	series with missing values replaced by periodic mean values.
xd	series after removing periodic mean.
pmpv	p-value for a one-way ANOVA test for equality of means.

Author(s)

Harry Hurd

References

Hurd, H. L., Miamee, A. G., (2007), Periodically Correlated Random Sequences: Spectral Theory and Practice, Wiley InterScience.

Westfall, P. H., Young, S. S. (1993), Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment, Wiley Series in Probability and Statistics.

See Also

persigest

Examples

```
data(arosa)
dev.set(which=1)
permest(t(arosa),12, 0.05, NaN,'arosa')
```

---

perpacf *Periodic PACF function*

Description

The function perpacf, given an input time series, a specified period T and a lag p, computes the periodic sample correlation coefficients $\pi(t,n)$ and returns their values as a matrix ppa of size $T \times (p+1)$.

The ppfcoeffab procedure transforms the output of perpacf into Fourier form, i.e. into Fourier coeficients, so we can represent $\pi(t,n)$ by its Fourier coefficients.

Function ppfplot plots perpacf coefficients returned by perpacf as function of n for each specified lag $t = 1, 2, \ldots, T$.

Usage

```
perpacf(x, T_t, p, missval)
ppfcoeffab(ppf, nsamp, printflg, datastr)
ppfplot(ppf, nsamp, alpha, datastr)
```

Arguments

x	input time series.
T_t	period of PC-T structure.
p	maximum lag used in computation.
missval	notation for missing values.
ppf	matrix of periodic PACF values (of size $T \times (p+1)$) returned by perpacf function.
nsamp	number of samples (periods) used to compute ppf.
printflg	parameter should be positive to return messages.
alpha	parameter for thresolds are displayed along with the Bonferroni corrected thresholds.
datastr	string name of data for printing.

Details

Procedure `perpacf` returns `ppa` matrix, where for each separation `n=0,1,...,p`, `ppa[,n]` is the value of $\hat{\pi}(t,n)$ for `t=1,2,...,T`. Further, since `T` is assumed to be the period of the underlying PC process, $\pi(t,n)$ is periodic in `t` with period `T`. So we can represent $\pi(t,n)$ by its Fourier coefficients. Further, if the variation in time of $\pi(t,n)$ is really smooth over the period, then looking at these Fourier coefficients (the output of `ppfcoeffab`) may be a more sensitive detector of linear dependence of x_{t+1} on the preceding `n` samples (think of `n` as fixed here) than looking at $\pi(t,n)$ for individual times. The `ppfcoeffab` procedure also needs the sample size `nsamp` used by `perpacf` in computing the $\pi(t,n)$ in order to compute p-values for the `pkab` coefficients. The p-values are computed assuming that for each `t`, $\pi(t,n)$ is `N(0,1/sqrt(nsamp))` under the null. The procedure `ppfcoeffab` is called in `parma_ident`.
Function `ppfplot` plots values of $\pi(t,n+1)$ and computes p-values for testing if $\pi(n_0+1,t)=0$ for all `t = 1, ..., T` and fix n_0 (p-values in column labelled $n_0 = n$) and if $\pi(n+1,t)=0$ for all `t = 1, ..., T` and $n_0 \le n \le nmax$ (p-values in column labelled $n_0 \le n \le nmax$). perpacf is plotted as function of n for each specified lag $t = 1, 2, \ldots, T$.

Value

The function `perpacf` returns two matrixes:

`ppa`	matrix of size $T \times (p+1)$ with perpacf coefficients.
`nsamp`	matrix of size $T \times (p+1)$ with numbers of samples used in estimation of sample correlation.

The function `ppfcoeffab` returns table of values:

`pihat_k`	Fourier coefficients `pkab` of `ppf` values.
`pv`	Bonferroni corrected p-values.

The function `ppfplot` returns plot of $\pi(t,n+1)$ coefficients and table of p-vaules for provided tests. Note that there are two plots; the first plot presents values of $\pi(t,n+1)$ for all considered t and n, whereas the second plot presents separate charts for particular t values.

Author(s)

Harry Hurd

References

Hurd, H. L., Miamee, A. G., (2007), Periodically Correlated Random Sequences: Spectral Theory and Practice, Wiley InterScience.

See Also

`peracf`

Examples

```
data(volumes)
perpacf_out<-perpacf(t(volumes),24,12,NaN)
ppa=perpacf_out$ppa
nsamp=perpacf_out$nsamp
ppfcoeffab(ppa,nsamp,1,'volumes')
ppfplot(ppa,41,.05,'volumes')
```

---

persigest *Periodic standard deviations*

Description

Assuming that the period T is known, procedure persigest plots and returns the estimated periodic standard deviation as a function of season. Missing data are permitted. The confidence intervals for these values, based on the chi-square distribution, are also computed and plotted. The de-meaned and normalized series xn is returned.
First, the periodic mean is computed using the method of permest. If at time t there is a missing value in the data, it is ignored in the computation of periodic standard deviation. For any season (t mod T) where all the data are missing, the periodic standard deviation is set to "Missing" and in the output vector xn all the values whose times are congruent with (t mod T) will be set to "Missing".

Usage

```
persigest(x, T_t, alpha, missval, datastr,...)
```

Arguments

x	input time series.
T_t	period of the computed standard deviation.
alpha	1-alpha is confidence interval containment probability using the chi-square distribution.
missval	notation for missing values.
datastr	string name of data for printing.
...	other arguments used in the plot: typeci, typepstd, pchci, pchpstd, colci, colpstd, pp; typeci / typepstd, pchci / pchpstd, colci / colpstd set the type, plot character and colors of confidence intervals / periodic mean values on the plot, pp should be positive to print and plot permest values. By default parameters are fixed to typeci = "o", typepstd = "b", pchci = 10, pchpstd = 15, colci = "red", colpstd = "blue", pp = 1.

Details

The series may contain missing values (we suggest using `NaN`) and the length of the series may not be an integer multiple of the period. The program returns and plots the periodic standard deviations with `1-alpha` confidence intervals based on all non-missing values present for each particular season. The p-value for Barttlet's test for homogenity of variance $\sigma(t) \equiv \sigma$ is also computed. Rejection of homogeneity (based on the `pspv` value) indicates a properly periodic variance, but leaves open whether or not series is simply the result of a stationary process subjected to amplitude-scale modulation. To resolve this$R(t+\tau, t)$ for some $\tau \neq 0$ need to be estimated.

Value

procedure returns:

`pstd`	periodic standard deviations values.
`lower, upper`	bounds of the confidence intervals.
`xn`	series after removing periodic mean and divided by standard deviations
`pspv`	p-value for Bartlett's test for the homogeneity of variance.

Author(s)

Harry Hurd

References

Hurd, H. L., Miamee, A. G., (2007), Periodically Correlated Random Sequences: Spectral Theory and Practice, Wiley InterScience.

See Also

`permest`

Examples

```
data(arosa)
dev.set(which=1)
persigest(t(arosa),12, 0.05, NaN,'arosa')
```

`perYW` *Yule-Walker estimators of PAR model*

Description

Assuming known `T`, procedure `perYW` implements Yule-Walker estimation method for a periodic autoregressive PAR(p) model. Order of autoregression `p`, which could be specified using sample periodic PACF, is constant for all seasons. For input time series `x`, matrix of parameters `phi` and vector of parameters `del` are computed.

Usage

```
perYW(x, T_t, p, missval)
```

Arguments

x	input time series.
T_t	period of PC-T structure (assumed constant over time).
p	order of the autoregression.
missval	notation for missing values.

Details

For fixed T, this procedure implements a periodic version of the Yule-Walker algorithm. The algorithm is based on solving for the best coefficients of LS prediction of $X(t)$ in terms of $X(t-1), ..., X(t-p+1)$. Sample autocorrelations are used in place of population autocorrelations in the expressions of the best coefficients.

Value

estimated parameters of PAR(p) model:

phi	matrix of coefficients for autoregressive part.
del	vector of noise weights (consider them variances of the shocks).

Author(s)

Harry Hurd

References

Brockwell, P. J., Davis, R. A. (1991), Time Series: Theory and Methods, 2nd Ed., Springer: New York.

Vecchia, A., (1985), Maximum Likelihood Estimation for Periodic Autoregressive Moving Average Models, Technometrics, v. 27, pp.375-384.

See Also

predictperYW, loglikef, parmaf

Examples

```
data(volumes)
perYW(volumes,24,2,NaN)
```

pgram *Plotting the periodogram of time series*

Description

The periodogram is a classical tool based on the sample Fourier transform for finding periodic components in a time series. The procedure `pgram` computes and plots an average of np periodograms where `np=floor(length(x)/fftlen)` where the input parameter `fftlen` is the length of the FFT; to get just 1 FFT of length `fftlen`, use `x(1:fftlen)` in place of `x`. To get a significance of high periodogram peaks, the procedure tests, at each frequency, the value of the averaged periodogram against the average of `2*halflen` neighboring cells (`halflen` on each side), and averaged over the np periodograms; the neighboring cell average is called the background. Significance of the ratio of center frequency average to the background average is computed from the F distribution.

Usage

```
pgram(x, fftlen,...)
```

Arguments

`x` input time series, missing values denoted by NaNs will be replaced in `pgram` by zeros.

`fftlen` length of FFT which will be used. In `pgram` we can specify the desired length of the FFT, then `x` is divided into pieces of this length. FFT is done on each of these pieces and the resulting magnitude squares values are added, so average of the periodograms for each frequency is obtained.

`...` other arguments that are connected with periodogram plot: `np1`, `np2`, `halflen`, `alpha`, `rejalpha`, `logsw`, `datastr`, `typeci`, `typepgram`, `colci`, `colpgram`, where
`np1` and `np2` are frequency indexes of the first and the last frequency in the periodogram plot; it is required that $np1 > halflen$ and usually $np2 = \lfloor length(x)/2 \rfloor$, because periodogram is symmetric;
`halflen` is a value on each side of the center for background estimation,
`alpha` is significance level for testing for periodic components,
`rejalpha` is significance level for rejecting outliers in the background estimation,
`logsw` if is equal to 1 plot of the periodogram is in $\log$ scale, else linear,
`datastr` string name of data for printing,
Parameters `typeci` / `typepgram`, `colci` / `colpgram` define the type and colors of confidence intervals / periodogram values on the plot.
By default they are fixed to `np1 = 5`, `np2 = fftlen/2`, `halflen = 4`, `alpha = .05`, `rejalpha = .01`, `logsw = 1`, `datastr = 'data'`, `typeci = "b"`, `typepgram = "b"`, `colci = "red"`, `colpgram = "blue"`.

Details

When we assume that period `T_t` of PC-T structure is unknown, function `pgram` enables us to find candidate for the period length assuming the period of the second order structure is the same as the period of the first order structure (IE, in the series itself).

Value

For any FFT index j (say where a strong peak occurs) j corresponds to the number of cycles in the FFT window, so the period can be easily computed as `T_t = fftlen/j`.

Author(s)

Harry Hurd

References

Box, G. E. P., Jenkins, G. M., Reinsel, G. (1994), Time Series Analysis, 3rd Ed., Prentice-Hall, Englewood Cliffs, NJ.

Hurd, H. L., Miamee, A. G., (2007), Periodically Correlated Random Sequences: Spectral Theory and Practice, Wiley InterScience.

See Also

`scoh`

Examples

```
data(volumes)
dev.set(which=1)
pgram(t(volumes),length(volumes),datastr='volumes')
```

`predictperYW` *Prediction for PAR model*

Description

Procedure `predictperYW` provideds the LMS forecast of a PAR(p) series. The Yule-Walker method is first use to estimate the LMS prediction coefficients using all the observed data in `x`.
Additionally, procedure `predseries` plots the predicted values of the series with real future values of the series (provided that such real data is available).

Usage

```
predictperYW(x, T_t, p, missval, start,...)
predseries(real, x, T_t, p, start,...)
```

Arguments

x	input time series.
T_t	period of PC-T structure.
p	order of autoregression, it is assumed constant over time.
missval	notation for missing values.
start	index of forecast value of the series; there are two possible scenarios: start<length(x) - procedure predictperYW enables to predict values of some piece of existing series (using Yule-Walker coefficients). In this case it is also necessary to define end value, as we want to predict values x[start:end] and compare them with known observations. start>length(x) - procedure predictperYW enables to predict future values of the series. In this scenario forecast of length start-length(x) is performed to find values xp[length(x)+1:start]. In this case one can use also predseries procedure to compare predicted future of the series with real data (if such data is available, see examples section).
real	the real future values of x series (historical data).
...	other arguments that will be connected with plot: realcol is a color of konwn values and predcol is a color of predicted values on the plot. By default parameters are fixed to realcol="blue", predcol="red".

Value

procedure predictperYW for start<length(x) plots values of x[start:end] and xp[start:end], where xp are predicted values; for start>length(x) function returns and plots two series:

x	input series together with predicted values added.
new	predicted part of the series only.

Procedure predseries plots predicted and real values of the series on the same plot.

Author(s)

Wioletta Wojtowicz

References

Box, G. E. P., Jenkins, G. M., Reinsel, G. (1994), Time Series Analysis, 3rd Ed., Prentice-Hall, Englewood Cliffs, NJ.

Brockwell, P. J., Davis, R. A. (1991), Time Series: Theory and Methods, 2nd Ed., Springer: New York.

Gladyshev, E. G., (1961), Periodically Correlated Random Sequences, Sov. Math., 2, 385-388.

Examples

```
data(volumes)
permest_out<-permest(t(volumes),24, 0.05, NaN,'volumes', pp=0)
xd=permest_out$xd
dev.set(which=1)
predictperYW(xd,24,2,NaN,956,end=980)

dev.set(which=1)
predictperYW(xd[1:980],24,2,NaN,1004)

data(volumes.sep)
dev.set(which=1)
realdata=c(volumes,volumes.sep)
predseries(realdata,t(volumes[1:980]),24,2,1004)
```

R_w_ma — *Covariance matrix for PARMA model (conditional)*

Description

Procedure `R_w_ma` computes the covariance matrix of the moving average part of a PARMA sequence. This is used in maximum likelihood estimation in conjunction with the Ansley transformation method of computing the likelihood of the sample conditioned on the firt `m = max(p; q)` samples being ignored (or set to null); see Ansley or Brockwell and Davis for background on the procedure. The method avoids the cumbersome calculation of full covariance matrix.

Usage

```
R_w_ma(theta, nstart, nlen)
```

Arguments

`theta`	matrix of size $T \times (q+1)$ contains vectorial parameters $[\theta_0, \theta_1, ..., \theta_q]$, where $\theta(0,t) = \sigma(t) = del(t)$, thus `theta = [del,theta_1,...,theta_q]`.
`nstart`	starting time, for conditional likelihood in PARMA set to `p+1`.
`nlen`	size of the covariance matrix.

Details

Procedure `R_w_ma` implements calculation of covariance matrix of size `nlen-p` from the parameters `theta` and `phi` of PARMA sequence. The result is returned as two vectors, first containing non-zero elements of covariance matrix and the second containing indexes of this parameters. Using these vectors covariance matrix can be easily reconstructed.

Value

procedure returns covariance matrix in sparse format as following:

`R`	vector of non-zero elements of covariance matrix.
`rindex`	vector of indexes of non-zero elements.

Author(s)

Harry Hurd

References

Ansley, (1979), An algorithm for the exact likelihood of a mixed autregressive moving average process, Biometrika, v.66, pp.59-65.

Brockwell, P. J., Davis, R. A. (1991), Time Series: Theory and Methods, 2nd Ed., Springer: New York.

See Also

`loglikec`, `loglikef`

Examples

```
T=12
nlen=480
p=2
 a=matrix(0,T,p)
q=1
 b=matrix(0,T,q)
a[1,1]=.8
a[2,1]=.3

phia<-ab2phth(a)
phi0=phia$phi
phi0=as.matrix(phi0)

b[1,1]=-.7
b[2,1]=-.6
thetab<-ab2phth(b)
theta0=thetab$phi
theta0=as.matrix(theta0)
del0=matrix(1,T,1)


R_w_ma(cbind(del0,theta0),p+1,T)
```

`scoh` *Plotting the squared coherence statistic of time series*

Description

The magnitude of squared coherence is computed in a specified square set of $(\lambda_p, \lambda_q) \in [0, 2\pi)$ and using a specified smoothing window. The perception of this empirical spectral coherence is aided by plotting the coherence values only at points where thereshold is exceeded. For identification/discovery of PC structure, the sample periodic mean should be first subtracted from the series because a periodic mean itself has PC structure that can dominate and confound the perception of the second order PC structure.

Usage

```
scoh(x, m, win,...)
```

Arguments

`x`	input time series.
`m`	length of the smoothing window.
`win`	vector of smoothing weights, they should be non-negative.
`...`	other arguments that will be connected with squared coherence statistic plot: `pfa`, `plflg`, `bfflg`, `ix`, `iy`, `nx`, `ny`, `datastr`, where `plflg` should be positive to plot values of statistic, `pfa` should be positive to plot threshold, `bfflg` is a Bonferroni correction parameter; it sholud be positive to correct `pfa` before thresholding, `ix` and `iy` are initial values at x and y axes (the lower left corner of plot), `nx, ny` are the incremental frequency indices above the starting point `(ix,iy)` (the ending values of frequency index are `ix+nx,iy+ny`), `datastr` string name of data for printing. By default they are fixed to `pfa = 1`, `plflg = 1`, `bfflg = 1`, `ix = 0`, `iy = 0`, `nx = length(x)/2`, `ny = length(x)/2`, `datastr = "data")`.

Details

To ensure that periodic structure seen in the spectral coherence image is not a consequence of an additive periodic mean, it is recommended that the `permest` function is first used to remove the periodic mean.

Value

The program returns plot of squared coherence statistic values, that exceed threshold.

Author(s)

Harry Hurd

References

Hurd, H. L., Gerr, N. L., (1991), Graphical Methods for Determining the Presence of Periodic Correlation in Time Series, J. Time Series Anal., (12), pp. 337-350(1991).

Hurd, H. L., Miamee, A. G., (2007), Periodically Correlated Random Sequences: Spectral Theory and Practice, Wiley InterScience.

See Also

pgram, permest

Examples

```
## Do not run
## It could take a few seconds

data(volumes)
m=16
win=matrix(1/m,1,m)
dev.set(which=1)
scoh(t(volumes),m,win,datastr='volumes')
```

---

volumes *Volumes of energy, Nord Pool Spot Exchange*

Description

One-dimensional discrete time series, which contains 984 real-valued observations of volumes of energy traded on the Nord Pool Spot Exchange from July 6th to August 31st 2010. Analysed series contains the hourly records only from weekdays from the considered period.

Usage

```
data(volumes)
```

Format

The format is: Time-Series [1:984] from 1 to 984: 24888 24316 23755 23354 23290 ...

Source

Data were found on http://www.npspot.com (Downloads -> Historical market data) selecting `Elspot volumes` and `hourly` resolution to download file `Elspot\_volumes\_2010\_hourly.xls`.

Examples

```
data(volumes)
message(volumes)
```

volumes.sep — *Volumes of energy, Nord Pool Spot Exchange, from 1st and 2nd September 2010.*

Description

One-dimenssional discrete time series, which conatins 48 real-valued observations of volumes of energy traded on the Nord Pool Spot Exchange. These are omitted before the last two days of `volumes` data and are used to compare the predicted values of the series `volumes` with real values in `volumes.sep`.

Usage

```
data(volumes.sep)
```

Format

The format is: Time-Series [1:48] from 1 to 48: 25281 24576 24306 24266 24515 ...

Source

Data were found on http://www.npspot.com (Downloads -> Historical market data) selecting `Elspot volumes` and `hourly` resolution to download file `Elspot\_volumes\_2010\_hourly.xls`.

Examples

```
data(volumes.sep)
message(volumes.sep)
```

Index