

# Package ‘weakARMA’

October 12, 2022

**Title** Tools for the Analysis of Weak ARMA Models

**Version** 1.0.3

**Date** 2022-04-04

**Description** Numerous time series admit autoregressive moving average (ARMA) representations, in which the errors are uncorrelated but not necessarily independent. These models are called weak ARMA by opposition to the standard ARMA models, also called strong ARMA models, in which the error terms are supposed to be independent and identically distributed (iid). This package allows the study of nonlinear time series models through weak ARMA representations. It determines identification, estimation and validation for ARMA models and for AR and MA models in particular. Functions can also be used in the strong case. This package also works on white noises by omitting arguments 'p', 'q', 'ar' and 'ma'. See Francq, C. and Zakoïan, J. (1998) <[doi:10.1016/S0378-3758\(97\)00139-0](https://doi.org/10.1016/S0378-3758(97)00139-0)> and Boubacar Maïnassara, Y. and Sausseureau, B. (2018) <[doi:10.1080/01621459.2017.1380030](https://doi.org/10.1080/01621459.2017.1380030)> for more details.

**Depends** R (>= 3.4.1)

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**Imports** CompQuadForm (>= 1.4.3), MASS (>= 7.3-54), matrixStats (>= 0.61), vars (>= 1.5-6)

**RoxygenNote** 7.1.2

**Suggests** timeSeries, testthat, knitr, rmarkdown, renv

**NeedsCompilation** no

**Maintainer** Julien Yves Rolland <[julien.rolland@univ-fcomte.fr](mailto:julien.rolland@univ-fcomte.fr)>

**URL** <https://plmlab.math.cnrs.fr/jrolland/weakARMA>

**BugReports** <https://plmlab.math.cnrs.fr/jrolland/weakARMA/-/issues>

**Author** Yacouba Boubacar Mainassara [aut]  
 (<<https://orcid.org/0000-0002-8604-5407>>),  
 Julien Yves Rolland [aut, cre]  
 (<<https://orcid.org/0000-0002-0960-6688>>),  
 Coraline Parguey [ctb],  
 Vincent Mouillot [ctb]

**Repository** CRAN

**Date/Publication** 2022-04-04 14:00:02 UTC

## R topics documented:

acf.gamma_m . . . . .	2
acf.univ . . . . .	3
ARMA.selec . . . . .	4
CAC40 . . . . .	6
CAC40return . . . . .	6
CAC40return.sq . . . . .	7
estimation . . . . .	7
gradient . . . . .	8
matXi . . . . .	9
meansq . . . . .	10
nl.acf . . . . .	11
omega . . . . .	12
portmanteauTest . . . . .	13
portmanteauTest.h . . . . .	14
signifparam . . . . .	16
sim.ARMA . . . . .	18
simGARCH . . . . .	19
VARest . . . . .	20
wnPT . . . . .	21
wnPT_SQ . . . . .	22
wnRT . . . . .	23
<b>Index</b>	<b>24</b>

---

acf.gamma_m	<i>Computation of autocovariance and autocorrelation for an ARMA residuals.</i>
-------------	---

---

## Description

Computes empirical autocovariances and autocorrelations function for an ARMA process for lag max given.

## Usage

```
acf.gamma_m(ar = NULL, ma = NULL, y, h, e = NULL)
```

**Arguments**

ar	Vector of AR coefficients. If NULL, it is a MA process.
ma	Vector of MA coefficients. If NULL, it is a AR process.
y	Univariate time series.
h	Computes autocovariances and autocorrelations from lag 1 to lag h with h an integer.
e	Vector of residuals. If NULL, the function will compute it.

**Value**

A list with :

gamma\_m Vector of the autocovariances.  
rho\_m Vector of the autocorrelations.

**See Also**

[acf.univ](#) for autocorrelation and autocovariance for only one given lag h.

**Examples**

```
param.estim <- estimation(p = 1, q = 1, y = CAC40return.sq)
acf.gamma_m(ar = param.estim$ar, ma = param.estim$ma, y = CAC40return.sq, h = 20)
```

---

acf.univ	<i>Computation of autocovariance and autocorrelation for an ARMA residuals.</i>
----------	---

---

**Description**

Computes empirical autocovariances and autocorrelations functions for an ARMA process for only one given lag.

**Usage**

```
acf.univ(ar = NULL, ma = NULL, y, h, e = NULL)
```

**Arguments**

ar	Vector of AR coefficients. If NULL, it is a MA process.
ma	Vector of MA coefficients. If NULL, it is a AR process.
y	Univariate time series.
h	Given lag to compute autocovariance and autocorrelation, with h an integer.
e	Vector of residuals of the time series. If NULL, the function will compute it.

**Value**

A list with :

autocov Value of the autocovariance.

autocor Value of the autocorrelation.

**See Also**

[acf.gamma\\_m](#) for autocorrelation and autocovariance for all h lag.

**Examples**

```
param.estim <- estimation(p = 1, q = 1, y = CAC40return.sq)
acf.univ(ar = param.estim$ar, ma = param.estim$ma, y = CAC40return.sq, h = 20)
```

---

 ARMA.selec

*Selection of ARMA models*


---

**Description**

Identifies the orders p and q of an ARMA model according to several information criteria.

**Usage**

```
ARMA.selec(data, P, Q, c = 2)
```

**Arguments**

data	Univariate time series.
P	Integer for the maximum lag order of autoregressive component.
Q	Integer for the maximum lag order of moving-average component.
c	Real number >1 needed to compute Hannan-Quinn information criterion.

**Details**

The fitted model which is favored is the one corresponding to the minimum value of the criterion. The most popular criterion is the Akaike information criterion (AIC). This was designed to be an approximately unbiased estimator of a fitted model. For small sample or when the number of fitted parameters is large, it is more appropriate to manipulate a corrected AIC version (AICc) which is more nearly unbiased. But these two criteria are inconsistent for model orders selection. If you want to use a consistent criterion, it is possible to take the Bayesian information criterion (BIC) or the Hannan-Quinn information criteria (HQ).

For the weak ARMA, i.e under the assumption that the errors are uncorrelated but not necessarily independant, modified criteria has been adapted : AICm, AICcm, BICm, HQm.

The criteria definitions are the following :

$$AIC = n \log(\sigma^2) + 2(p + q)$$

$$AICm = n \log(\sigma^2) + \frac{Tr(IJ^{-1})}{\sigma^2}$$

$$AICc = n \log(\sigma^2) + n + \frac{n}{(n - (p + q + 1))} 2(p + q)$$

$$AICcm = n \log(\sigma^2) + \frac{n^2}{(n - (p + q + 1))} + \frac{n}{(2(n - (p + q + 1)))} \frac{Tr(IJ^{-1})}{\sigma^2}$$

$$BIC = n \log(\sigma^2) + (p + q) \log(n)$$

$$BICm = n \log(\sigma^2) + \frac{1}{2} \frac{Tr(IJ^{-1})}{\sigma^2} \log(n)$$

$$HQ = n \log(\sigma^2) + 2c(p + q) \log(\log(n))$$

$$HQm = n \log(\sigma^2) + c \frac{Tr(IJ^{-1})}{\sigma^2} \log(\log(n))$$

### Value

A list of the different criteria, each item contains the matrix of the computed value for the different model and the selected order with this criterion (corresponding to the minimum value in the previous matrix).

### References

Boubacar Maïnassara, Y. 2012, Selection of weak VARMA models by modified Akaike's information criteria, *Journal of Time Series Analysis*, vol. 33, no. 1, pp. 121-130

Boubacar Maïnassara, Y. and Kokonendji, C. C. 2016, Modified Schwarz and Hannan-Quin information criteria for weak VARMA models, *Stat Inference Stoch Process*, vol. 19, no. 2, pp. 199-217

### Examples

ARMA.selec (CAC40return.sq, P = 3, Q = 3)

---

CAC40	<i>Paris stock exchange</i>
-------	-----------------------------

---

**Description**

This data set considers market index at the closure of the market from March 1, 1990 to June 14, 2021.

**Usage**

CAC40

**Format**

A vector with the variable Close.

There are 7936 observations. We removed every NULL values.

**Source**

Data pulled from Yahoo Finance: 'https://fr.finance.yahoo.com/quote/%5EFCHI/history?p=%5EFCHI'

**See Also**

[CAC40return](#) and [CAC40return.sq](#)

---

CAC40return	<i>Paris stock exchange return</i>
-------------	------------------------------------

---

**Description**

This data set considers CAC40 return at the closure of the market from March 2, 1990 to June 14, 2021.

**Usage**

CAC40return

**Format**

A numerical vector with 7935 observations.

We computed every value from the dataset [CAC40](#) with the following code:

```
cac<-CAC40;
n<-length(cac);
rend<-rep(0,n);
rend[2:n]<-((log(cac[2:n]/cac[1:(n-1)]))*100);
CAC40return<-rend[2:n]
```

**See Also**

[CAC40](#) and [CAC40return.sq](#)

---

CAC40return.sq	<i>Paris stock exchange square return</i>
----------------	---

---

**Description**

This data set considers CAC40 square return at the closure of the market from March 2, 1990 to June 14, 2021.

**Usage**

```
CAC40return.sq
```

**Format**

A numerical vector with 7935 observations.

We computed every value from the dataset [CAC40](#) with the following code:

```
cac<-CAC40;
n<-length(cac);
rend<-rep(0,n);
rend[2:n]<-(log(cac[2:n]/cac[1:(n-1)]))*100);
CAC40return.sq<-rend[2:n]^2
```

**See Also**

[CAC40](#) and [CAC40return](#)

---

estimation	<i>Parameters estimation of a time series.</i>
------------	--

---

**Description**

Estimates the parameters of a time series for given orders p and q

**Usage**

```
estimation(p = NULL, q = NULL, y, meanparam = FALSE)
```

**Arguments**

p	Order of AR, if NULL, MA is computed.
q	Order of MA, if NULL, AR is computed.
y	Univariate time series.
meanparam	Logical argument if the mean parameter has to be computed or not. If FALSE $\mu$ is not computed.

**Details**

This function uses the algorithm BFGS in the function `optim` to minimize our objective function `meansq`.

**Value**

List of estimate coefficients:

- mu Mean parameter.
- ar Vector of AR coefficients with length is equal to p.
- ma Vector of MA coefficients with length is equal to q.
- sigma.carre Mean square residuals.

**References**

Francq, C. and Zakoïan, J. 1998, Estimating linear representations of nonlinear processes *Journal of Statistical Planning and Inference*, vol. 68, no. 1, pp. 145-165.

**Examples**

```
y<-sim.ARMA(1000,ar = c(0.9,-0.3), ma = 0.2, method = "product")
estimation(p = 2, q = 1, y = y)

estimation(p = 1, q = 1, y = CAC40return.sq, meanparam = TRUE)
```

---

gradient

*Computation the gradient of the residuals of an ARMA model*

---

**Description**

Computes the gradient of the residuals of an ARMA model.

**Usage**

```
gradient(ar = NULL, ma = NULL, y)
```



**Arguments**

ar	Vector of ar coefficients.
ma	Vector of ma coefficients.
y	Univariate time series.

**Value**

A list containing:

der.eps Matrix of the gradient.

esp Vector of residuals.

**Examples**

```
est<-estimation(p = 1, q = 1, y = CAC40return.sq)
gradient(ar = est$ar, ma = est$ma, y = CAC40return.sq)
```

---

 matXi

*Estimation of Fisher information matrix I*


---

**Description**

Uses a consistent estimator of the matrix I based on an autoregressive spectral estimator.

**Usage**

```
matXi(data, p = 0, q = 0)
```

**Arguments**

data	Matrix of dimension (p+q,n).
p	Dimension of AR estimate coefficients.
q	Dimension of MA estimate coefficients.

**Value**

Estimate Fisher information matrix  $I = \sum_{h=-\infty}^{+\infty} cov(2e_t \nabla e_t, 2e_{t-h} \nabla e_{t-h})$  where  $\nabla e_t$  denotes the gradient of the residuals.

## References

Berk, Kenneth N. 1974, Consistent autoregressive spectral estimates, *The Annals of Statistics*, vol. 2, pp. 489-502.

Boubacar Mainassara, Y. and Francq, C. 2011, Estimating structural VARMA models with uncorrelated but non-independent error terms, *Journal of Multivariate Analysis*, vol. 102, no. 3, pp. 496-505.

Boubacar Mainassara, Y. and Carbon, M. and Francq, C. 2012, Computing and estimating information matrices of weak ARMA models *Computational Statistics & Data Analysis*, vol. 56, no. 2, pp. 345-361.

---

meansq

*Function optim will minimize*

---

## Description

Computes the mean square of the time series at the point  $x$ , will be minimize with the [optim](#) function in our function [estimation](#).

## Usage

```
meansq(x, dim.ar = NULL, dim.ma = NULL, y)
```

## Arguments

$x$	One point in $\mathbb{R}^{(p+q)}$ .
<code>dim.ar</code>	Length of AR vector.
<code>dim.ma</code>	Length of MA vector.
$y$	Vector of a time series.

## Value

`ms` Mean square at the point  $x$ .

---

nl.acf	<i>Autocorrelogram</i>
--------	------------------------

---

### Description

Plots autocorrelogram for non linear process.

### Usage

```
nl.acf(
  ar = NULL,
  ma = NULL,
  y,
  main = NULL,
  nlag = NULL,
  conplevel = 0.05,
  z = 1.2,
  aff = "both"
)
```

### Arguments

ar	Vector of AR coefficients. If NULL, we consider a MA process.
ma	Vector of MA coefficients. If NULL, we consider an AR process.
y	Univariate time series.
main	Character string representing the title for the plot.
nlag	Maximum lag at which to calculate the acf. If NULL, it is determinate by $nlag = \min(10\log(n))$ where n is the number of observation.
conplevel	Value of the confidence level, 5% by default.
z	Zoom on the graph.
aff	Specify the method between SN, M and both (see in Details).

### Details

For the argument `aff` you have the choice between: SN, M and both. SN prints the self-normalized method (see Boubacar Maïnassara and Sausseureau) in green, M prints the modified method introduced by Francq, Roy and Zakoïan (see also Boubacar Maïnassara) in red and both prints both of the methods.

### Value

An autocorrelogram with every autocorrelations from 1 to a lag max, and with methods you choose to print.

**Note**

The only value available for the argument `conflevel` are 0.1, 0.05, 0.025, 0.01 or 0.005.

**References**

Boubacar Maïnassara, Y. 2011, Multivariate portmanteau test for structural VARMA models with uncorrelated but non-independent error terms *Journal of Statistical Planning and Inference*, vol. 141, no. 8, pp. 2961-2975.

Boubacar Maïnassara, Y. and Saussereau, B. 2018, Diagnostic checking in multivariate ARMA models with dependent errors using normalized residual autocorrelations, *Journal of the American Statistical Association*, vol. 113, no. 524, pp. 1813-1827.

Franco, C., Roy, R. and Zakoïan, J.M. 2005, Diagnostic Checking in ARMA Models with Uncorrelated Errors, *Journal of the American Statistical Association*, vol. 100, no. 470, pp. 532-544.

Lobato, I.N. 2001, Testing that a dependant process is uncorrelated. *J. Amer. Statist. Assos.* 96, vol. 455, pp. 1066-1076.

**Examples**

```
est<-estimation(p = 1, q = 1, y = CAC40return.sq)
nl.acf(ar = est$ar, ma = est$ma, y = CAC40return.sq, main = "Autocorrelation of an ARMA(1,1)
residuals of the CAC40 return square", nlag = 20)
```

---

 omega

---

*Computation of Fisher information matrice*


---

**Description**

Computes matrices of Fisher information like  $I$ ,  $J$ .

**Usage**

```
omega(ar = NULL, ma = NULL, y)
```

**Arguments**

ar	Vector of AR coefficients. If NULL, the simulation is a MA process.
ma	Vector of MA coefficients. If NULL, the simulation is a AR process.
y	Univariate time series.

**Value**

A list of matrix containing:

I Matrix I computed in function `matXi`.

J Matrix J computed as  $\frac{2}{n} H(e)H(e)^t$  where  $e$  is the residuals vector.

J.inv Inverse of the matrix J.

matOmega Matrix variance-covariance in the weak case computed as  $J^{-1}IJ^{-1}$ .

matvar.strong Matrix variance-covariance in the strong case computed as  $2\sigma^2 J^{-1}$ .

standard.dev.Omega Standard deviation of the matrix matOmega.

standard.dev.strong Standard deviation of the matrix matvar.strong.

sig2 Innovation variance estimate.

**Examples**

```
y <- sim.ARMA(n = 1000, ar = c(0.95, -0.8), ma = -0.6)
est<-estimation(p = 2, q = 1, y = y)
omega(ar = est$ar, ma = est$ma, y = y)
```

```
estCAC<-estimation(p = 1, q = 1, y = CAC40return.sq, meanparam = TRUE)
omega(ar = estCAC$ar, ma = estCAC$ma, y = CAC40return.sq)
```

---

portmanteauTest	<i>Portmanteau tests</i>
-----------------	--------------------------

---

**Description**

Realizes portmanteau tests of the first  $m$  lags, this function uses `portmanteauTest.h` for  $h$  in 1: $m$ .

**Usage**

```
portmanteauTest(ar = NULL, ma = NULL, y, m = NULL)
```

**Arguments**

ar	Vector of AR coefficients. If NULL, it is a MA process.
ma	Vector of MA coefficients. If NULL, it is an AR process.
y	Univariate time series.
m	Integer for the lag.

**Value**

A list of vectors of length  $m$ , corresponding to statistics and p-value for each lag, for standard, modified and self-normalized Ljung-Box and Box-Pierce methods.

## References

Boubacar Maïnassara, Y. 2011, Multivariate portmanteau test for structural VARMA models with uncorrelated but non-independent error terms *Journal of Statistical Planning and Inference*, vol. 141, no. 8, pp. 2961-2975.

Boubacar Maïnassara, Y. and Sausseureau, B. 2018, Diagnostic checking in multivariate ARMA models with dependent errors using normalized residual autocorrelations , *Journal of the American Statistical Association*, vol. 113, no. 524, pp. 1813-1827.

Franco, C., Roy, R. and Zakoïan, J.M. 2005, Diagnostic Checking in ARMA Models with Uncorrelated Errors, *Journal of the American Statistical Association*, vol. 100, no. 470, pp. 532-544.

## See Also

[portmanteauTest.h](#) to obtain statistics for only one h lag.

## Examples

```
est<-estimation(p = 1, q = 1, y = CAC40return.sq)
portmanteauTest(ar = est$ar, ma = est$ma, y = CAC40return.sq, m = 20)
```

---

portmanteauTest.h      *Portmanteau tests for one lag.*

---

## Description

Computes Box-Pierce and Ljung-Box statistics for standard, modified and self-normalized test procedures.

## Usage

```
portmanteauTest.h(ar = NULL, ma = NULL, y, h, grad = NULL)
```

## Arguments

ar	Vector of AR coefficients. If NULL, it is a MA process.
ma	Vector of MA coefficients. If NULL, it is an AR process.
y	Univariate time series.
h	Integer for the chosen lag.
grad	Gradient of the series from the function <a href="#">gradient</a> . If NULL gradient will be computed.

## Details

Portmanteau statistics are generally used to test the null hypothesis.  $H_0 : X_t$  satisfies an ARMA(p,q) representation.

The Box-Pierce (BP) and Ljung-Box (LB) statistics, defined as follows, are based on the residual empirical autocorrelation.

$$Q_m^{BP} = n \sum_h^m \rho^2(h)$$

$$Q_m^{LB} = n(n+2) \sum_h^m \frac{\rho^2(h)}{(n-h)}$$

The standard test procedure consists in rejecting the null hypothesis of an ARMA(p,q) model if the statistic  $Q_m > \chi^2(1 - \alpha)$  where  $\chi^2(1 - \alpha)$  denotes the  $(1 - \alpha)$ -quantile of a chi-squared distribution with  $m - (p+q)$  (where  $m > p + q$ ) degrees of freedom. The two statistics have the same asymptotic distribution, but the LB statistic has the reputation of doing better for small or medium sized samples.

But the significance limits of the residual autocorrelation can be very different for an ARMA models with iid noise and ARMA models with only uncorrelated noise but dependant. The standard test is obtained under the stronger assumption that  $\epsilon_t$  is iid. So we give an another way to obtain the exact asymptotic distribution of the standard portmanteau statistics under the weak dependence assumptions.

Under  $H_0$ , the statistics  $Q_m^{BP}$  and  $Q_m^{LB}$  converge in distribution as  $n \rightarrow \infty$ , to

$$Z_m(\xi_m) := \sum_i^m \xi_{i,m} Z_i^2$$

where  $\xi_m = (\xi'_{1,m}, \dots, \xi'_{m,m})$  is the eigenvalues vector of the asymptotic covariance matrix of the residual autocorrelations vector and  $Z_1, \dots, Z_m$  are independent  $\mathcal{N}(0, 1)$  variables.

So when the error process is a weak white noise, the asymptotic distribution  $Q_m^{BP}$  and  $Q_m^{LB}$  statistics is a weighted sum of chi-squared. The distribution of the quadratic form  $Z_m(\xi_m)$  can be computed using the algorithm by Imhof available here : [imhof](#)

We propose an alternative method where we do not estimate an asymptotic covariance matrix. It is based on a self-normalization based approach to construct a new test-statistic which is asymptotically distribution-free under the null hypothesis.

The sample autocorrelation, at lag  $h$  take the form  $\hat{\rho}(h) = \frac{\hat{\Gamma}(h)}{\hat{\Gamma}(0)}$ . Where  $\hat{\Gamma}(h) = \frac{1}{n} \sum_{t=h+1}^n \hat{\epsilon}_t \hat{\epsilon}_{t-h}$ .

With  $\hat{\Gamma}_m = (\hat{\Gamma}(1), \dots, \hat{\Gamma}(m))$  The vector of the first  $m$  sample autocorrelations is written  $\hat{\rho}_m = (\hat{\rho}(1), \dots, \hat{\rho}(m))'$ .

The normalization matrix is defined by  $\hat{C}_m = \frac{1}{n^2} \sum_{t=1}^n \hat{S}_t \hat{S}_t'$  where  $\hat{S}_t = \sum_{j=1}^t (\hat{\Lambda} \hat{U}_j - \hat{\Gamma}_m)$ .

The sample autocorrelations satisfy  $Q_m^{SN} = n \hat{\sigma}^4 \hat{\rho}_m' \hat{C}_m^{-1} \hat{\rho}_m \rightarrow U_m$ .

$\tilde{Q}_m^{SN} = n \hat{\sigma}^4 \hat{\rho}_m' D_{n,m}^{1/2} \hat{C}_m^{-1} D_{n,m}^{1/2} \hat{\rho}_m \rightarrow U_m$  repressating respectively the version modified of Box-

Pierce (BP) and Ljung-Box (LB) statistics. Where  $D_{n,m} = \begin{pmatrix} \frac{n}{n-1} & & 0 \\ & \ddots & \\ 0 & & \frac{n}{n-m} \end{pmatrix}$ . The critical

values for  $U_m$  have been tabulated by Lobato.

**Value**

A list including statistics and p-value:

Pm.BP Standard portmanteau Box-Pierce statistics.

PvalBP p-value corresponding at standard test where the asymptotic distribution is approximated by a chi-squared

PvalBP.Imhof p-value corresponding at the exact asymptotic distribution of the standard portmanteau Box-Pierce statistics.

Pm.LB Standard portmanteau Box-Pierce statistics.

PvalLB p-value corresponding at standard test where the asymptotic distribution is approximated by a chi-squared.

PvalLB.Imhof p-value corresponding at the exact asymptotic distribution of the standard portmanteau Ljung-Box statistics.

LB.modSN Ljung-Box statistic with the self-normalization method.

BP.modSN Box-Pierce statistic with the self-normalization method.

**References**

Boubacar Maïnassara, Y. 2011, Multivariate portmanteau test for structural VARMA models with uncorrelated but non-independent error terms *Journal of Statistical Planning and Inference*, vol. 141, no. 8, pp. 2961-2975.

Boubacar Maïnassara, Y. and Saussereau, B. 2018, Diagnostic checking in multivariate ARMA models with dependent errors using normalized residual autocorrelations , *Journal of the American Statistical Association*, vol. 113, no. 524, pp. 1813-1827.

Franco, C., Roy, R. and Zakoïan, J.M. 2005, Diagnostic Checking in ARMA Models with Uncorrelated Errors, *Journal of the American Statistical Association*, vol. 100, no. 470 pp. 532-544

Lobato, I.N. 2001, Testing that a dependant process is uncorrelated. *J. Amer. Statist. Assos.* 96, vol. 455, pp. 1066-1076.

**See Also**

[portmanteauTest](#) to obtain the statistics of all m lags.

---

signifparam

*Computes the parameters significance*

---

**Description**

Computes a matrix with estimated coefficient and their significance.



**Usage**

```
signifparam(
  ar = NULL,
  ma = NULL,
  p = NULL,
  q = NULL,
  y,
  sd.strong = NULL,
  sd.weak = NULL,
  meanparam = TRUE,
  mu = NULL
)
```

**Arguments**

ar	Vector of AR coefficients, if NULL, MA process.
ma	Vector of MA coefficients, if NULL, AR process.
p	Order of AR, if NULL MA process.
q	Order of MA, if NULL AR process.
y	Univariate time series.
sd.strong	Standard error of time series in the strong case computed in <a href="#">omega</a> , if not provided the function will compute it.
sd.weak	Standard error of time series in the weak case computed in <a href="#">omega</a> , if not provided the function will compute it.
meanparam	If $\mu$ of the time series needs to be computed.
mu	Value of $\mu$ , if it is known and if the meanparam is TRUE. If not known the function will compute it.

**Details**

The function needs at least one pair between: ar and/or ma, or p and/or q to be executed. It will be faster with all the parameters provided.

**Value**

Matrix of the estimate coefficient with their significance.

coef Estimation of each coefficient.

sd Standard deviation in each case.

t-ratio T-ratio corresponding to each coefficient.

signif Significance of each parameter. Must be small, if not the parameter is not significant.

**Examples**

```
signifparam(p = 1, q = 2, y = CAC40return.sq) #The last parameter is not significant.
signifparam(p = 1, q = 1, y = CAC40return.sq) #All the parameters are significant.
```

---

sim.ARMA                      *Simulation of ARMA(p,q) model.*

---

### Description

Simulates an ARMA, AR or MA process according to the arguments given.

### Usage

```
sim.ARMA(
  n,
  ar = NULL,
  ma = NULL,
  sigma = 1,
  eta = NULL,
  method = "strong",
  k = 1,
  mu = 0,
  ...
)
```

### Arguments

n	Number of observations.
ar	Vector of AR coefficients. If NULL, the simulation is a MA process.
ma	Vector of MA coefficients. If NULL, the simulation is a AR process.
sigma	Standard deviation.
eta	Vector of white noise sequence. Allows the user to use his own white noise.
method	Defines the kind of noise used for the simulation. By default, the noise used is strong. See 'Details'.
k	Integer used in the creation of the noise. See 'Details'.
mu	Integer for the mean of the series.
...	Arguments needed to simulate GARCH noise. See 'Details'.

### Details

ARMA model is of the following form :

$$X_t - \mu = e_t + a_1(X_{t-1} - \mu) + a_2(X_{t-2} - \mu) + \dots + a_p(X_{t-p} - \mu) - b_1e_{t-1} - b_2e_{t-2} - \dots - b_qe_{t-q}$$

where  $e_t$  is a sequence of uncorrelated random variables with zero mean and common variance  $\sigma^2 > 0$ .  $ar = (a_1, a_2, \dots, a_p)$  are autoregressive coefficients and  $ma = (b_1, b_2, \dots, b_q)$  are moving average coefficients. Characteristic polynomials of ar and ma must constitute a stationary process.

Method "strong" realise a simulation with gaussian white noise.

Method "product", "ratio" and "product.square" realise a simulation with a weak white noise. These methods employ respectively the functions `wnPT`, `wnRT` and `wnPT_SQ` to simulate nonlinear ARMA model. So, the parameter `k` is an argument of these functions. See `wnPT`, `wnRT` or `wnPT_SQ`. Method "GARCH" gives an ARMA process with a GARCH noise. See `simGARCH`.

### Value

Returns a vector containing the `n` simulated observations of the time series.

### References

Francq, C. and Zakoïan, J.M. 1998, Estimating linear representations of nonlinear processes, *Journal of Statistical Planning and Inference*, vol. 68, no. 1, pp. 145-165

### See Also

`arima.sim`

### Examples

```
y <- sim.ARMA(n = 100, ar = 0.95, ma = -0.6, method = "strong" )
y2 <- sim.ARMA(n = 100, ar = 0.95, ma = -0.6, method = "ratio")
y3 <- sim.ARMA(n = 100, ar = 0.95, ma = -0.6, method = "GARCH", c = 1, A = 0.1, B = 0.88)
y4 <- sim.ARMA(n = 100, ar = 0.95, ma = -0.6, method = "product")
y5 <- sim.ARMA(n = 100, ar = 0.95, ma = -0.6, method = "product.square")
```

---

simGARCH

*GARCH process*

---

### Description

Simulates a GARCH process which is an example of a weak white noise.

### Usage

```
simGARCH(n, c, A, B = NULL, ninit = 100)
```

### Arguments

<code>n</code>	Number of observations.
<code>c</code>	Positive number.
<code>A</code>	Vector of ARCH coefficients $\geq 0$ .
<code>B</code>	Vector of GARCH coefficients $\geq 0$ . If NULL, the simulation is a ARCH process.
<code>ninit</code>	Length of 'burn-in' period.

**Value**

Vector of size n containing a nonlinear sequence  $\epsilon_t$  such as

$$\epsilon_t = H_t^{1/2} \eta_t$$

where

$$H_t = c + a_1 \epsilon_{t-1}^2 + \dots + a_q \epsilon_{t-q}^2 + b_1 H_{t-1} + \dots + b_p H_{t-p}$$

**References**

Francq C. and Zakořan J.M., 2010, *GARCH models: structure, statistical inference and financial applications*

**See Also**

[wnRT](#), [wnPT](#), [wnPT\\_SQ](#)

**Examples**

```
simGARCH(100, c = 1, A = 0.25)
simGARCH(100, c = 1, A = 0.1, B = 0.88)
```

---

 VARest

*Estimation of VAR(p) model*


---

**Description**

Estimates the coefficients of a VAR(p) model. Used in [matXi](#).

**Usage**

```
VARest(x, p)
```

**Arguments**

x	Matrix of dimension (n,p+q).
p	Integer for the lag order.

**Value**

A list containing:

- ac Coefficients data matrix.
- p Integer of the lag order.
- k Dimension of the VAR.
- res Matrix of residuals.

---

wnPT	<i>Weak white noise</i>
------	-------------------------

---

**Description**

Simulates an uncorrelated but dependant noise process.

**Usage**

```
wnPT(n, sigma = 1, k = 1, ninit = 100)
```

**Arguments**

n	Number of observations.
sigma	Standard deviation.
k	Integer corresponding to the number of past observation will be used.
ninit	Length of 'burn-in' period.

**Value**

Vector of size n containing a nonlinear sequence  $X_i$  such as  $X_i = Z_i Z_{i-1} \dots Z_{i-k}$ , where  $Z_i$  is a sequence of iid random variables mean-zero random variable with variance  $\sigma^2$ .

**References**

Romano, J. and Thombs, L. 1996, Inference for autocorrelation under weak assumptions, *Journal of the American Statistical Association*, vol. 91, no. 434, pp. 590-600

**See Also**

[wnRT](#), [wnPT\\_SQ](#), [simGARCH](#)

**Examples**

```
wnPT(100)
wnPT(100, sigma = 1, k = 1)
wnPT(100, k = 0) #strong noise
```

---

wnPT_SQ	<i>Weak white noise</i>
---------	-------------------------

---

**Description**

Simulates an uncorrelated but dependant noise process.

**Usage**

```
wnPT_SQ(n, sigma = 1, k = 1, ninit = 100)
```

**Arguments**

n	Number of observations.
sigma	Standard deviation.
k	Integer corresponding to the number of past observation will be used.
ninit	Length of 'burn-in' period.

**Value**

Vector of size n containing a nonlinear sequence  $X_i$  such as  $X_i = Z_i^2 Z_{i-1} \dots Z_{i-k}$ , where  $Z_i$  is a sequence of iid random variables mean-zero random variable with variance  $\sigma^2$ .

**References**

Romano, J. and Thombs, L. 1996, Inference for autocorrelation under weak assumptions, *Journal of the American Statistical Association*, vol. 91, no. 434, pp. 590-600

**See Also**

[wnRT](#), [wnPT](#), [simGARCH](#)

**Examples**

```
wnPT_SQ(100)
wnPT_SQ(100, sigma = 1, k = 1)
```

---

wnRT	<i>Weak white noise</i>
------	-------------------------

---

**Description**

Simulates an uncorrelated but dependant noise process.

**Usage**

```
wnRT(n, sigma = 1, k = 1, ninit = 100)
```

**Arguments**

n	Number of observations.
sigma	Standard deviation.
k	Integer $\neq 0$ to prevent a zero denominator.
ninit	Length of 'burn-in' period.

**Value**

Vector of size n containing a nonlinear sequence  $X_i$  such as  $X_i = \frac{Z_i}{|Z_{i+1}|+k}$ , where  $Z_i$  is a sequence of iid random variables mean-zero random variable with variance  $\sigma^2$ .

**References**

Romano, J. and Thombs, L. 1996, Inference for autocorrelation under weak assumptions, *Journal of the American Statistical Association*, vol. 91, no. 434, pp. 590-600

**See Also**

[wnPT](#), [wnPT\\_SQ](#), [simGARCH](#)

**Examples**

```
wnRT(100)
wnRT(100, sigma = 1)
```

# Index

## \* datasets

CAC40, 6

CAC40return, 6

CAC40return.sq, 7

acf.gamma\_m, 2, 4

acf.univ, 3, 3

arima.sim, 19

ARMA.selec, 4

CAC40, 6, 6, 7

CAC40return, 6, 6, 7

CAC40return.sq, 6, 7, 7

estimation, 7, 10

gradient, 8, 14

imhof, 15

matXi, 9, 13, 20

meansq, 8, 10

nl.acf, 11

omega, 12, 17

optim, 10

portmanteauTest, 13, 16

portmanteauTest.h, 13, 14, 14

signifparam, 16

sim.ARMA, 18

simGARCH, 19, 19, 21–23

VARest, 20

wnPT, 19, 20, 21, 22, 23

wnPT\_SQ, 19–21, 22, 23

wnRT, 19–22, 23