Package: MSTest (via r-universe)

October 6, 2026

Type Package

Title Hypothesis Testing for Markov Switching Models

Date 2026-07-13

Version 0.1.9

Maintainer Gabriel Rodriguez-Rondon <gabrodriguezrondon@gmail.com>

Description Implementation of hypothesis testing procedures described in Hansen (1992) <doi:10.1002/jae.3950070506>, Carrasco, Hu, & Ploberger (2014) <doi:10.3982/ECTA8609>, Dufour & Luger (2017) <doi:10.1080/07474938.2017.1307548>, and Rodriguez-Rondon & Dufour (2026a) <doi:10.34989/swp-2026-23> that can be used to identify the number of regimes in Markov switching models. For a detailed description of the package, see Rodriguez-Rondon & Dufour (2026b) <doi:10.34989/swp-2026-7>.

License GPL (>= 2)

LazyData TRUE

RoxygenNote 7.3.3

Encoding UTF-8

URL https://github.com/roga11/MSTest

BugReports https://github.com/roga11/MSTest/issues

Imports stats, rlang, nloptr, Rcpp (>= 1.0.1), numDeriv, pracma, GenSA, pso, GA, graphics

LinkingTo Rcpp, RcppArmadillo

Depends R (>= 4.0.0)

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

SystemRequirements C++17

NeedsCompilation yes

Author Gabriel Rodriguez-Rondon [cre, aut] (ORCID:
<https://orcid.org/0009-0005-3769-9921>), Jean-Marie Dufour
[aut]

Repository https://cran.r-universe.dev

Date/Publication 2026-07-13 22:20:14 UTC

RemoteUrl https://github.com/cran/MSTest

RemoteRef HEAD

RemoteSha 51ac673cbc59ec622578b5cf1f91f233fd182d9b

Contents

MSTest-package . . . 3
ARmdl . . . 4
ARXmdl . . . 6
chp10GNP . . . 8
CHPbootCV . . . 8
CHPTest . . . 9
DLMCTest . . . 10
DLMMCTest . . . 12
hamilton84GNP . . . 15
HLRTest . . . 15
HMmdl . . . 18
LMCLRTest . . . 22
MCpval . . . 24
MMCLRTest . . . 25
MSARmdl . . . 28
MSARXmdl . . . 32
MSVARmdl . . . 36
MSVARXmdl . . . 40
Nmdl . . . 44
simuAR . . . 46
simuARX . . . 47
simuHMM . . . 48
simuMSAR . . . 49
simuMSARX . . . 50
simuMSVAR . . . 51
simuMSVARX . . . 52
simuNorm . . . 53
simuVAR . . . 54
simuVARX . . . 55
USGNP . . . 56
USRGDP . . . 57
VARmdl . . . 58
VARXmdl . . . 60

Index **63**

MSTest-package *Testing Markov Switching Models*

Description

This package implements hypothesis testing procedures that can be used to identify the number of regimes in a Markov-Switching model.

Author(s)

Gabriel Rodriguez-Rondon, gabrodriguezrondon@gmail.com (Maintainer)

Jean-Marie Dufour, jean-marie.dufour@mcgill.ca

References

Carrasco, Marine, Liang Hu, and Werner Ploberger. 2014. "Optimal test for Markov switching parameters." *Econometrica* 82 (2): 765–784.

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society*. Series B 39 (1): 1–38..

Dufour, Jean-Marie, and Richard Luger. 2017. "Identification-robust moment-based tests for Markov switching in autoregressive models." *Econometric Reviews* 36 (6-9): 713–727.

Kasahara, Hiroyuk, and Katsum Shimotsu. 2018. "Testing the number of regimes in Markov regime switching models." arXiv preprint arXiv:1801.06862.

Krolzig, Hans-Martin. 1997. "The markov-switching vector autoregressive model.". *Springer*.

Hamilton, James D. 1989. "A new approach to the economic analysis of nonstationary time series and the business cycle." *Econometrica* 57 (2): 357–384.

Hamilton, James D. 1994. "Time series analysis". *Princeton university press*.

Hansen, Bruce E. 1992. "The likelihood ratio test under nonstandard conditions: testing the Markov switching model of GNP." *Journal of applied Econometrics* 7 (S1): S61–S82.

Rodriguez-Rondon, G., & Dufour, J.-M. 2026a. "Monte Carlo Likelihood-Ratio Tests for Markov Switching Models." *Bank of Canada Staff Working Paper*, No. 2026-23. doi: 10.34989/swp-2026-23.

Rodriguez-Rondon, G., & Dufour, J.-M. 2026b. "MSTest: An R-Package for Testing Markov Switching Models." *Bank of Canada Staff Working Paper*, No. 2026-7. doi: 10.34989/swp-2026-7.

Qu, Zhongjun, and Fan Zhuo. 2021. "Likelihood Ratio-Based Tests for Markov Regime Switching." *The Review of Economic Studies* 88 (2): 937–968.

ARmdl *Autoregressive Model*

Description

This function estimates an autoregresive model with p lags. This can be used for the null hypothesis of a linear model against an alternative hypothesis of a Markov switching autoregressive model with k regimes.

Usage

```
ARmdl(Y, p, control = list())
```

Arguments

Y	A (T x 1) matrix of observations.
p	Integer determining the number of autoregressive lags.
control	List with model options including: • const: Boolean determining whether to estimate model with constant if TRUE or not if FALSE. Default is TRUE. • getSE: Boolean determining whether to compute standard errors of parameters if TRUE or not if FALSE. Default is TRUE.

Value

List of class ARmdl (S3 object) with model attributes including:

- y: a (T-p x 1) matrix of observations.
- X: a (T-p x p + const) matrix of lagged observations with a leading column of 1s if const=TRUE or not if const=FALSE.
- x: a (T-p x p) matrix of lagged observations.
- fitted: a (T-p x 1) matrix of fitted values.
- resid: a (T-p x 1) matrix of residuals.
- mu: estimated mean of the process.
- beta: a ((1 + p) x 1) matrix of estimated coefficients.
- intercept: estimate of intercept.
- phi: estimates of autoregressive coefficients.
- stdev: estimated standard deviation of the process.
- sigma: estimated variance of the process.
- theta: vector containing: mu, sigma, and phi.
- theta_mu_ind: vector indicating location of mean with 1 and 0 otherwise.
- theta_sig_ind: vector indicating location of variance with 1 and 0 otherwise.

- theta_var_ind: vector indicating location of variance with 1 and 0 otherwise. This is the same as theta_sig_ind in ARmdl.
- theta_phi_ind: vector indicating location of autoregressive coefficients with 1 and 0 otherwise.
- stationary: Boolean indicating if process is stationary if TRUE or non-stationary if FALSE.
- n: number of observations after lag transformation (i.e., n = T-p).
- p: number of autoregressive lags.
- q: number of series. This is always 1 in ARmdl.
- k: number of regimes. This is always 1 in ARmdl.
- control: List with model options used.
- logLike: log-likelihood.
- AIC: Akaike information criterion.
- BIC: Bayesian (Schwarz) information criterion.
- Hess: Hessian matrix. Approximated using hessian and only returned if getSE=TRUE.
- info_mat: Information matrix. Computed as the inverse of -Hess. If matrix is not PD then nearest PD matrix is obtained using nearest_spd. Only returned if getSE=TRUE.
- nearPD_used: Boolean determining whether nearPD function was used on info_mat if TRUE or not if FALSE. Only returned if getSE=TRUE.
- theta_se: standard errors of parameters in theta. Only returned if getSE=TRUE.

See Also

MSARmdl

Examples

```
set.seed(1234)
# Define DGP of AR process
mdl_ar <- list(n     = 500,
               mu    = 5,
               sigma = 2,
               phi   = c(0.5,0.2))

# Simulate process using simuAR() function
y_simu <- simuAR(mdl_ar)

# Set options for model estimation
control <- list(const  = TRUE,
                getSE  = TRUE)

# Estimate model
y_ar_mdl <- ARmdl(y_simu$y, p = 2, control)
y_ar_mdl
```

ARXmdl *Autoregressive X Model*

Description

This function estimates an ARX model with p lags. This can be used for the null hypothesis of a linear model against an alternative hypothesis of a Markov switching autoregressive model with k regimes.

Usage

```
ARXmdl(Y, p, Z, control = list())
```

Arguments

Y	A (T x 1) matrix of observations.
p	Integer determining the number of autoregressive lags.
Z	A (T x qz) matrix of exogenous regressors.
control	List with model options including: • const: Boolean determining whether to estimate model with constant if TRUE or not if FALSE. Default is TRUE. • getSE: Boolean determining whether to compute standard errors of parameters if TRUE or not if FALSE. Default is TRUE.

Value

List of class ARmdl (S3 object) with model attributes including:

- y: a (T-p x 1) matrix of observations.
- X: a (T-p x p + const) matrix of lagged observations with a leading column of 1s if const=TRUE or not if const=FALSE.
- x: a (T-p x p) matrix of lagged observations.
- fitted: a (T-p x 1) matrix of fitted values.
- resid: a (T-p x 1) matrix of residuals.
- mu: estimated mean of the process.
- beta: a ((1 + p + qz) x 1) matrix of estimated coefficients.
- betaZ: a (qz x q) matrix of estimated exogenous regressor coefficients.
- intercept: estimate of intercept.
- phi: estimates of autoregressive coefficients.
- stdev: estimated standard deviation of the process.
- sigma: estimated variance of the process.
- theta: vector containing: mu, sigma, and phi.

- theta_mu_ind: vector indicating location of mean with 1 and 0 otherwise.
- theta_sig_ind: vector indicating location of variance with 1 and 0 otherwise.
- theta_var_ind: vector indicating location of variance with 1 and 0 otherwise. This is the same as theta_sig_ind in ARmdl.
- theta_phi_ind: vector indicating location of autoregressive coefficients with 1 and 0 otherwise.
- stationary: Boolean indicating if process is stationary if TRUE or non-stationary if FALSE.
- n: number of observations after lag transformation (i.e., n = T-p).
- p: number of autoregressive lags.
- q: number of series. This is always 1 in ARmdl.
- k: number of regimes. This is always 1 in ARmdl.
- control: List with model options used.
- logLike: log-likelihood.
- AIC: Akaike information criterion.
- BIC: Bayesian (Schwarz) information criterion.
- Hess: Hessian matrix. Approximated using hessian and only returned if getSE=TRUE.
- info_mat: Information matrix. Computed as the inverse of -Hess. If matrix is not PD then nearest PD matrix is obtained using nearest_spd. Only returned if getSE=TRUE.
- nearPD_used: Boolean determining whether nearPD function was used on info_mat if TRUE or not if FALSE. Only returned if getSE=TRUE.
- theta_se: standard errors of parameters in theta. Only returned if getSE=TRUE.

See Also

MSARmdl

Examples

```
set.seed(1234)
# Define DGP of AR process
mdl_ar <- list(n     = 500,
               mu    = 5,
               sigma = 2,
               phi   = c(0.5,0.2))

# Simulate process using simuAR() function
y_simu <- simuAR(mdl_ar)

# Set options for model estimation
control <- list(const  = TRUE,
                getSE  = TRUE)

# Estimate model
y_ar_mdl <- ARmdl(y_simu$y, p = 2, control)
y_ar_mdl
```

chp10GNP *Carrasco, Hu, & Ploberger 2010 GNP data*

Description

This data is the extension of the GNP series used in CHP (2014), Econometrica. This series ranges from 1951Q2 to 2010Q4.

Usage

```
chp10GNP
```

Format

Date Vector of dates

GNP US GNP series

GNP_gr log difference of US GNP series

Source

https://www.econometricsociety.org/content/supplement-optimal-test-markov-switching-parameters

References

Carrasco, Marine, Liang Hu, and Werner Ploberger. 2014. "Optimal test for Markov switching parameters." *Econometrica* 82 (2): 765–784.

CHPbootCV *Bootstrap critical values for CHP 2014 parameter stability test*

Description

This bootstrap procedure is described on pg. 771 of CHP 2014.

Usage

```
CHPbootCV(mdl, rho_b, N, msvar)
```

Arguments

mdl	List containing model attributes (see ARmdl).
rho_b	Number determining bounds for distribution of rh0 (i.e. rho ~ [-rho_b,rho_b]).
N	Number of bootstrap simulations.
msvar	Boolean indicator. If TRUE, there is a switch in variance. If FALSE only switch in mean is considered.

Value

Bootstrap critical values

References

Carrasco, Marine, Liang Hu, and Werner Ploberger. 2014. "Optimal test for Markov switching parameters." *Econometrica* 82 (2): 765–784.

`CHPTest` *Carrasco, Hu, and Ploberger (2014) parameter stability test*

Description

This function performs the CHP (2014) parameter stability test as outline in Carrasco, M., Hu, L. and Ploberger, W. (2014). Original source code can be found here.

Usage

```
CHPTest(Y, p, control = list())
```

Arguments

`Y`	A (`T x 1`) matrix of observations.
`p`	Integer determining the number of autoregressive lags.
`control`	List with test procedure options including:

- N: Integer determining the number of Bootstrap iterations. Default is set to `3000` as in paper.
- rho_b: Number determining bounds for distribution of `rh0` (i.e. `rho ~ [-rho_b,rho_b]`).
- msvar: Boolean indicator. If `TRUE`, there is a switch in variance. If `FALSE` only switch in mean is considered.
- getSE: Boolean indicator. If `TRUE`, standard errors for restricted model are estimated. If `FALSE` no standard errors are estimated. Default is `TRUE`.

Value

List of class `CHPTest` (`S3` object) with model attributes including:

- mdl_h0: List with restricted model attributes. This will be of class `ARmdl` (`S3` object). See `ARmdl`.
- supTS: supTS test statistic value.
- expTS: expTS test statistic value.
- supTS_N: A (`N x 1`) vector with simulated supTS test statistics under null hypothesis.
- expTS_N: A (`N x 1`) vector with simulated expTS test statistics under null hypothesis.

- pval_supTS: P-value for supTS version of parameter stability test.
- pval_expTS: P-value for expTS version of parameter stability test.
- supTS_cv: Vector with 90%, 95%, and 99% bootstrap critical values for supTS version of parameter stability test.
- expTS_cv: Vector with 90%, 95%, and 99% bootstrap critical values for expTS version of parameter stability test.
- control: List with test procedure options used.

References

Carrasco, Marine, Liang Hu, and Werner Ploberger. 2014. "Optimal test for Markov switching parameters." *Econometrica* 82 (2): 765–784.

Examples

```
# load data used in Hamilton 1989
y84 <- as.matrix(hamilton84GNP$GNP_gr)


# Set test procedure options
control = list(N = 1000,
               rho_b = 0.7,
               msvar = FALSE)

# perform test with switch in mean only on Hamilton 1989 data

  mdl_84_msmu <- CHPTest(y84, p = 4, control = control)
  summary(mdl_84_msmu)
```

`DLMCTest` *Monte Carlo moment-based test for Markov switching model*

Description

This function performs the Local Monte Carlo moment-based test for Markov switching autoregressive models proposed in Dufour & Luger (2017).

Usage

```
DLMCTest(Y, p, control = list())
```

Arguments

`Y`	Series to be tested
`p`	Number of autoregressive lags.
`control`	List with test procedure options including:

- N: Integer determining the number of Monte Carlo simulations. Default is set to `99` as in paper.
- simdist_N: Integer determining the number of simulations for CDF distribution approximation. Default is set to `10000`.
- getSE: Boolean indicator. If `TRUE`, standard errors for restricted model are estimated. If `FALSE` no standard errors are estimated. Default is `TRUE`.

Value

List of class `DLMCTest` (`S3` object) with attributes including:

- mdl_h0: List with restricted model attributes. This will be of class `ARmdl` if `p>0` or `Nmdl` otherwise (`S3` objects). See `ARmdl` or `Nmdl`.
- theta: Value of nuisance parameters. Specifically, these are the consistent estimates of nuisance parameters as discussed in Dufour & Luger (2017) LMC procedure.
- S0: A (`1 x 4`)) matrix containing the four moment-based test statistics defined in (11) - (14) in Dufour & Luger (2017).
- F0_min: Test statistic value for min version of Local Monte Carlo moment-based test.
- F0_prod: Test statistic value for prod version of Local Monte Carlo moment-based test.
- FN_min: A (`N x 1`) vector with simulated test statistics for min version of Local Monte Carlo moment-based test under null hypothesis.
- FN_prod: A (`N x 1`) vector with simulated test statistics for prod version of Local Monte Carlo moment-based test under null hypothesis.
- pval_min: P-value for min version of Local Monte Carlo moment-based test.
- pval_prod: P-value for prod version of Local Monte Carlo moment-based test.
- FN_min_cv: Vector with 90%, 95%, and 99% Monte Carlo critical values for min version of Local Monte Carlo moment-based test.
- FN_prod_cv: Vector with 90%, 95%, and 99% Monte Carlo critical values for prod version of Local Monte Carlo moment-based test.
- control: List with test procedure options used.

References

Dufour, J. M., & Luger, R. 2017. "Identification-robust moment-based tests for Markov switching in autoregressive models." *Econometric Reviews*, 36(6-9), 713-727.

Examples

```
set.seed(1234)
# load data used in Hamilton 1989 and extended data used in CHP 2014
y84 <- as.matrix(hamilton84GNP$GNP_gr)
y10 <- as.matrix(chp10GNP$GNP_gr)

# Set test procedure options
lmc_control = list(N = 99,
                   simdist_N = 10000,
                   getSE = TRUE)

# perform test on Hamilton 1989 data
lmc_gnp84 <- DLMCTest(y84, p = 4, control = lmc_control)
summary(lmc_gnp84)
```

`DLMMCTest` *Maximized Monte Carlo moment-based test for Markov switching model*

Description

This function performs the maximized Monte Carlo moment-based test for Markov switching autoregressive models proposed in Dufour & Luger (2017).

Usage

```
DLMMCTest(Y, p, control = list())
```

Arguments

`Y`	Series to be tested
`p`	Number of autoregressive lags.
`control`	List with test procedure options including: • N: Integer determining the number of Monte Carlo simulations. Default is set to `99` as in paper. • simdist_N: Integer determining the number of simulations for CDF distribution approximation. Default is set to `10000`. • getSE: Boolean indicator. If `TRUE`, standard errors for restricted model are estimated. If `FALSE` no standard errors are estimated. Default is `TRUE`. • eps: Fixed positive constant that does not depend on `T` used to determine lower and upper bounds on consistent set considered for nuisance parameter space. • CI_union: Boolean indicator determining if union between `eps` and confidence interval is used to determine lower and upper bound on consistent set considered for nuisance parameter space. If `TRUE` union is used and if `FALSE` only `eps` is used. Note that if standard errors obtained are not finite then only `eps` is used. Default is `FALSE`.

- lambda: Numeric value for penalty on stationary constraint not being met. Default is 100.
- stationary_ind: Boolean indicator determining if only stationary solutions should be considered if TRUE or any solution can be considered if FALSE. Default is TRUE.
- phi_low: Vector with lower bound for autoregressive parameters when optimizing. Default is NULL.
- phi_upp: Vector with upper bound for autoregressive parameters when optimizing. Default is NULL.
- optim_type: String determining type of numerical optimization algorithm to use. Available options are: pso, GenSA, GA. Default is pso.
- silence: Boolean indicator determining if optimization updates should be silenced if TRUE or not if FALSE. Default is FALSE.
- threshold_stop: Numeric value determining the maximum possible p-value attainable. Default is 1.
- maxit: Int determining the maximum number of function evaluations. Default is 1000.
- optim_control: List containing other optimization options specific to the numerical optimization algorithm used. For other options see documentation of numerical algorithm chosen.

Value

List of class DLMCTest (S3 object) with attributes including:

- mdl_h0: List with restricted model attributes. This will be of class ARmdl if p>0 or Nmdl otherwise (S3 objects). See ARmdl or Nmdl.
- theta_max_min: Value of nuisance parameters when min version of p-value is maximized as discussed in Dufour & Luger (2017) MMC procedure.
- theta_max_prod: Value of nuisance parameters when prod version of p-value is maximized as discussed in Dufour & Luger (2017) MMC procedure.
- theta_low: Lower bound on nuisance parameter values used when searching for maximum p-value.
- theta_upp: Upper bound on nuisance parameter values used when searching for maximum p-value.
- S0_min: A (1 x 4)) matrix containing the four moment-based test statistics defined in (11) - (14) in Dufour & Luger (2017) when theta_min is used.
- S0_prod: A (1 x 4)) matrix containing the four moment-based test statistics defined in (11) - (14) in Dufour & Luger (2017) when theta_prod is used.
- F0_min: Test statistic value for min version of Maximized Monte Carlo moment-based test.
- F0_prod: Test statistic value for prod version of Maximized Monte Carlo moment-based test.
- FN_min: A (N x 1) vector with simulated test statistics for min version of Maximized Monte Carlo moment-based test under null hypothesis.
- FN_prod: A (N x 1) vector with simulated test statistics for prod version of Maximized Monte Carlo moment-based test under null hypothesis.

- pval_min: Maximum p-value for min version of Maximized Monte Carlo moment-based test.
- pval_prod: Maximum p-value for prod version of Local Monte Carlo moment-based test.
- FN_min_cv: Vector with 90%, 95%, and 99% Monte Carlo critical values for min version of Local Monte Carlo moment-based test.
- FN_prod_cv: Vector with 90%, 95%, and 99% Monte Carlo critical values for prod version of Local Monte Carlo moment-based test.
- control: List with test procedure options used.
- optim_min_output: List with optimization output for min version of Maximized Monte Carlo moment-based test.
- optim_prod_output: List with optimization output for prod version of Maximized Monte Carlo moment-based test.

References

Dufour, J. M., & Luger, R. 2017. "Identification-robust moment-based tests for Markov switching in autoregressive models." *Econometric Reviews*, 36(6-9), 713-727.

Examples

```
set.seed(1234)
# load data used in Hamilton 1989 and extended data used in CHP 2014
y84 <- as.matrix(hamilton84GNP$GNP_gr)
y10 <- as.matrix(chp10GNP$GNP_gr)

# Set test procedure options
mmc_control <- list(N = 99,
                    simdist_N = 10000,
                    getSE = TRUE,
                    eps = 0,
                    CI_union = TRUE,
                    lambda = 100,
                    stationary_ind = TRUE,
                    optim_type = "GenSA",
                    silence = FALSE,
                    threshold_stop = 1,
                    maxit = 200)


# perform test on Hamilton 1989 data

  mmc_gnp84 <- DLMMCTest(y84, p = 4, control = mmc_control)
  summary(mmc_gnp84)
```

hamilton84GNP *Hamilton 1984 & Hansen 1992 GNP data*

Description

This data set is used in Hansen (1992) to test the US GNP model proposed by Hamilton (1989). This series ranges from 1951Q2 to 1984Q4.

Usage

```
hamilton84GNP
```

Format

Date Vector of dates

GNP US GNP series

GNP_gr US GNP log difference

Source

https://www.ssc.wisc.edu/~bhansen/progs/jae_92.html

References

Hansen, Bruce E. 1992. "The likelihood ratio test under nonstandard conditions: testing the Markov switching model of GNP." *Journal of applied Econometrics* 7 (S1): S61–S82.

Hamilton, James D. 1989. "A new approach to the economic analysis of nonstationary time series and the business cycle." *Econometrica* 57 (2): 357–384.

HLRTest *Hansen (1992) likelihood ratio test*

Description

This function performs Hansen's likelihood ratio test as described in Hansen (1992). Original source code can be found here.

Usage

```
HLRTest(Y, p, control = list())
```

Arguments

Y	A (T x 1) matrix of observations.
p	Integer determining the number of autoregressive lags.
control	List with test procedure options including:

- ix: List of Markov Switching parameters. 1 = just mean c(1,2) = mean and first param, (default: 1).
- msvar: Boolean indicator. If TRUE, there is a switch in variance. If FALSE only switch in mean is considered. Default is FALSE.
- qbound: Indicator that bounds q by 1-p (default: FALSE).
- gridsize: Integer determining the number of grid points for markov switching parameters. Default is 20.
- pgrid_from: Double determining the initial grid point for transition probabilities. Default is 0.1.
- pgrid_by: Double determining the step size for grid points of transition probabilities. This, along with p_gridsize will determine the bounds of search space. Default is 0.075.
- pgrid_to: Double determining the end grid point for transition probabilities. Default is 0.925.
- mugrid_from: Double determining the minimum value of mean in second regime. Default is 0.1.
- mugrid_by: Double determining the step size for grid points of mean in second regime. This, along with gridsize will determine the max value of mean in second regime. Default is 0.1.
- siggrid_from: Double determining the minimum value of sigma in second regime (if msvar = TRUE). Default is 0.1.
- siggrid_by: Double determining the step size for grid points of sigma in second regime. This, along with gridsize will determine the max value of sigma in second regime. Default is 0.1.
- N: Integer determining the number of replications. Default is 1000.
- nwband: Integer determining maximum bandwidth in Bartlett kernel. Critical values and p-values are returned for each bandwidth from 0:nwband as suggested in Hansen (1996). Default is 4.
- theta_null_low: Vector determining lower bound on parameters under the null hypothesis. Length of vector should be number of model coefficients + 1 for variance. Default is to only bound variance at 0.01.
- theta_null_upp: Vector determining upper bound on parameters under the null hypothesis. Length of vector should be number of model coefficients + 1 for variance. Default is to no bounds (i.e. Inf).
- optim_method: String determining the type of optimization procedure used. Allowed options are "gp-optim" for general purpose optimization using optim from or slsqp. Default is "gp-optim".

Value

List of class HLRTest (S3 object) with model attributes including:

- mdl_h0: List with restricted model attributes. This will be of class `ARmdl` (`S3` object). See `ARmdl`.
- LR0: Likelihood ratio test statistic value.
- LRN: A (`N x 1`) vector with simulated LRT statistics under null hypothesis.
- pval: P-value.
- LR_cv: A (`nwband x 3`) matrix with 90%, 95%, and 99% critical values in each column respectively.
- coef: Vector of coefficients from restricted model and grid search that maximized standardized LRT.
- control: List with test procedure options used.

References

Hansen, Bruce E. 1992. "The likelihood ratio test under nonstandard conditions: testing the Markov switching model of GNP." *Journal of applied Econometrics* 7 (S1): S61–S82.

Hansen, Bruce E. 1996. "Erratum: The likelihood ratio test under nonstandard conditions: testing the Markov switching model of GNP." *Journal of applied Econometrics* 7 (S1): S61–S82.

Examples

```
# --------------------------- Use simulated process ----------------------------
set.seed(1234)
# Define DGP of MS AR process
mdl_ms2 <- list(n     = 200,
                mu    = c(5,1),
                sigma = c(1,1),
                phi   = c(0.5),
                k     = 2,
                P     = rbind(c(0.90, 0.10),
                              c(0.10, 0.90)))

# Simulate process using simuMSAR() function
y_ms_simu <- simuMSAR(mdl_ms2)

hlrt_control  <- list(ix          = 1,
                      gridsize    = 7,
                      pgrid_from  = 0.05,
                      pgrid_by    = 0.05,
                      pgrid_to    = 0.95,
                      mugrid_from = 0,
                      mugrid_by   = 1)


  hlrt <- HLRTest(y_ms_simu$y, p = 1, control = hlrt_control)
  summary(hlrt)
```

HMmdl *Hidden Markov model*

Description

This function estimates a Hidden Markov model with k regimes.

Usage

```
HMmdl(Y, k, Z = NULL, control = list())
```

Arguments

Y	a (T x q) matrix of observations.
k	integer determining the number of regimes to use in estimation. Must be greater than or equal to 2.
Z	an otpional (T x qz) matrix of exogenous regressors. Default is NULL.
control	List with model options including:

- getSE: Boolean. If TRUE standard errors are computed and returned. If FALSE standard errors are not computed. Default is TRUE.
- se_method: String determining the standard-error method. Options are "hessian" (default) and "louis" (Louis 1982 expected-information); the Louis method is used automatically as a fallback when the Hessian is ill-conditioned.
- msmu: Boolean. If TRUE model is estimated with switch in mean. If FALSE model is estimated with constant mean. Default is TRUE.
- msvar: Boolean. If TRUE model is estimated with switch in variance. If FALSE model is estimated with constant variance. Default is TRUE.
- init_theta: vector of initial values. vector must contain (1 x q) vector mu, vech(sigma), and vec(P) where sigma is a (q x q) covariance matrix.This is optional. Default is NULL, in which case initVals_MSARmdl is used to generate initial values.
- method: string determining which method to use. Options are 'EM' for EM algorithm or 'MLE' for Maximum Likelihood Estimation. Default is 'EM'.
- maxit: integer determining the maximum number of EM iterations.
- thtol: double determining the convergence criterion for the relative change in the parameter estimates theta between iterations (used when conv is "theta", "both", or "both-A"). Default is 1e-6.
- ltol: double determining the convergence criterion for the relative change in the log-likelihood between iterations (used when conv is "loglik", "both", "loglik-A", or "both-A"). Default is 1e-7.
- conv: string determining the EM convergence criterion. "loglik" (the default) stops on the relative change in the log-likelihood; "theta" stops on the relative change in the parameters (the convention of Hamilton (1994)

and of earlier versions of this package); "both" requires both, following Krolzig (1997); "loglik-A" and "both-A" replace the log-likelihood test with the Aitken-accelerated criterion of Bohning et al. (1994) and McLachlan and Krishnan (2008). Default is "loglik".

- maxit_converge: integer determining the maximum number of initial values attempted until solution is finite. For example, if parameters in theta or logLike are NaN another set of initial values (up to maxit_converge) is attempted until finite values are returned. This does not occur frequently for most types of data but may be useful in some cases. Once finite values are obtained, this counts as one iteration towards use_diff_init. Default is 500.
- use_diff_init: integer determining how many different initial values to try (that do not return NaN; see maxit_converge). Default is 1.
- mle_variance_constraint: double used to determine the lower bound on the smallest eigenvalue for the covariance matrix of each regime. Default is 1e-3.
- mle_theta_low: Vector with lower bounds on parameters (Used only if method = "MLE"). Default is NULL.
- mle_theta_upp: Vector with upper bounds on parameters (Used only if method = "MLE"). Default is NULL.

Value

List of class HMmdl (S3 object) with model attributes including:

- y: a (T x q) matrix of observations.
- fitted: a (T x q) matrix of fitted values.
- resid: a (T x q) matrix of residuals.
- mu: a (k x q) matrix of estimated means of each process.
- beta: if q=1, this is a ((1 + qz) x k) matrix of estimated coefficients. If q>1, this is a list containing k separate ((1 + qz) x q) matrix of estimated coefficients for each regime.
- betaZ: a (qz x q) matrix of estimated exogenous regressor coefficients.
- intercept: a (k x q) matrix of estimated intercept of each process. If Z is Null, this is the same as mu.
- stdev: If q=1, this is a (k x 1) matrix with estimated standard. If q>1, this is a List with k (q x q) matrices with estimated standard deviation on the diagonal.
- sigma: If q=1, this is a (k x 1) matrix with variances. If q>1, this is a List with k (q x q) estimated covariance matrix.
- theta: vector containing: mu and vech(sigma).
- theta_mu_ind: vector indicating location of mean with 1 and 0 otherwise.
- theta_sig_ind: vector indicating location of variance and covariances with 1 and 0 otherwise.
- theta_var_ind: vector indicating location of variances with 1 and 0 otherwise.
- theta_P_ind: vector indicating location of transition matrix elements with 1 and 0 otherwise.
- n: number of observations (same as T).

- q: number of series.
- k: number of regimes in estimated model.
- P: a (k x k) transition matrix.
- pinf: a (k x 1) vector with limiting probabilities of each regime.
- St: a (T x k) vector with smoothed probabilities of each regime at each time t.
- deltath: double with the maximum relative change in vector theta on the last iteration.
- iterations: number of EM iterations performed to achieve convergence (if less than maxit).
- converged: Boolean. TRUE if the conv convergence criterion was met before maxit iterations, FALSE otherwise.
- theta_0: vector of initial values used.
- init_used: number of different initial values used to get a finite solution. See description of input maxit_converge.
- msmu: Boolean. If TRUE model was estimated with switch in mean. If FALSE model was estimated with constant mean.
- msvar: Boolean. If TRUE model was estimated with switch in variance. If FALSE model was estimated with constant variance.
- control: List with model options used.
- logLike: log-likelihood.
- AIC: Akaike information criterion.
- BIC: Bayesian (Schwarz) information criterion.
- Hess: Hessian matrix. Approximated using hessian and only returned if getSE=TRUE.
- info_mat: Information matrix. Computed as the inverse of -Hess. If matrix is not PD then nearest PD matrix is obtained using nearest_spd. Only returned if getSE=TRUE.
- nearPD_used: Boolean determining whether nearPD function was used on info_mat if TRUE or not if FALSE. Only returned if getSE=TRUE.
- theta_se: standard errors of parameters in theta. Only returned if getSE=TRUE.
- trace: List with Lists of estimation output for each initial value used due to use_diff_init > 1.

References

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society*. Series B 39 (1): 1–38..

Böhning, D., E. Dietz, R. Schaub, P. Schlattmann, and B. G. Lindsay. 1994. "The distribution of the likelihood ratio for mixtures of densities from the one-parameter exponential family." *Annals of the Institute of Statistical Mathematics* 46 (2): 373–388.

McLachlan, G. J., and T. Krishnan. 2008. *The EM Algorithm and Extensions*. 2nd ed. Hoboken, New Jersey: John Wiley & Sons.

Hamilton, James D. 1990. "Analysis of time series subject to changes in regime." *Journal of econometrics*, 45 (1-2): 39–70.

Krolzig, Hans-Martin. 1997. "The markov-switching vector autoregressive model.". Springer.

See Also

Nmdl

Examples

```
set.seed(1234)
# ----- Univariate ----- #
# Define DGP
mdl_hmm <- list(n     = 1000,
                q     = 1,
                mu    = as.matrix(c(5,-2)),
                sigma = list(as.matrix(5.0),
                             as.matrix(7.0)),
                k     = 2,
                P     = rbind(c(0.90, 0.10),
                              c(0.10, 0.90)))

# Simulate process using simuHMM() function
y_hmm_simu <- simuHMM(mdl_hmm)

# Set options for model estimation
control <- list(msmu   = TRUE,
                msvar  = TRUE,
                method = "EM",
                use_diff_init = 1)


# Estimate model
y_hmm_mdl <- HMmdl(y_hmm_simu$y, k = 2, control = control)
summary(y_hmm_mdl)



# ----- Multivariate normal process ----- #
# Define DGP
mdl_hmm <- list(n     = 500,
                q     = 2,
                mu    = rbind(c(5, -2),
                              c(10, 2)),
                sigma = list(rbind(c(5.0, 1.5),
                                   c(1.5, 1.0)),
                             rbind(c(7.0, 3.0),
                                   c(3.0, 2.0))),
                k     = 2,
                P     = rbind(c(0.90, 0.10),
                              c(0.10, 0.90)))

# Simulate process using simuHMM() function
y_hmm_simu <- simuHMM(mdl_hmm)

# Set options for model estimation
control <- list(msmu   = TRUE,
```

```
                msvar  = TRUE,
                method = "EM",
                use_diff_init = 1)

# Estimate model

  y_hmm_mdl <- HMmdl(y_hmm_simu$y, k = 2, control = control)
  summary(y_hmm_mdl)
```

LMCLRTest *Monte Carlo Likelihood Ratio Test*

Description

This function performs the Local Monte Carlo likelihood ratio test (LMC-LRT) proposed in Rodriguez-Rondon & Dufour (2026a). As discussed in their work, this test can be applied in very general settings and can be used to compare varioous regimes under the null and under the alternative.

Usage

```
LMCLRTest(Y, p, k0, k1, Z = NULL, control = list())
```

Arguments

Y	Series to be tested. Must be a (T x q) matrix where T is the number of time observations and q is the number of variables.
p	Number of autoregressive lags. Must be greater than or equal to 0.
k0	Number of regimes under null hypothesis. Must be greater than or equal to 1.
k1	Number of regimes under alternative hypothesis. Must be greater than k0.
Z	Exogenous regressors. Optional input and default is NULL. When used, it should be a (T x qz) matrix where T is the number of time observations and q is the number of exogenous variables.
control	List with test procedure options including: • N: Integer determining the number of Monte Carlo simulations. Default is set to 99 as in paper. • burnin: Number of simulated observations to remove from beginning. Default is 100. • converge_check: String or NULL determining if convergence of model(s) should be verified. Allowed inputs are: "null", "alt", "both", or NULL. If NULL (default) no model convergence is verified. • workers: Integer determining the number of workers to use for parallel computing. Default is 0 (sequential). If workers > N, the effective count is capped at N and an informational message is printed.

- mc_seed: Integer seed for reproducible Monte Carlo simulations. When set, seeds the RNG before observed-data estimation and pre-draws innovations once, ensuring fully reproducible results (including the observed LRT statistic). When `workers > 0`, also sets L'Ecuyer-CMRG parallel RNG streams. Default is `NULL` (non-reproducible, fresh draws).
- mdl_h0_control: List with restricted model options. See `Nmdl`, `ARmdl`, `VARmdl`, `HMmdl`, `MSARmdl`, or `MSVARmdl` documentation for available and default values.
- mdl_h1_control: List with unrestricted model options. See `HMmdl`, `MSARmdl`, or `MSVARmdl` documentation for available and default values.
- use_diff_init_sim: Value which determines the number of initial values to use when estimating models for null distribution. Default is set to use the same as specified in `mdl_h0_control` and `mdl_h1_control`.

Value

List of class `LMCLRTest` (`S3` object) with attributes including:

- mdl_h0: List with restricted model attributes.
- mdl_h1: List with unrestricted model attributes.
- LRT_0: Value of test statistic from observed data.
- LRN: A (`N x 1`) vector of test statistics from data simulated under the null hypothesis.
- pval: P-value of Local Monte Carlo Likelihood Ratio Test.
- LRN_cv: Vector with 90%, 95%, and 99% Monte Carlo simulated critical values (from vector `LRN`). These are not asymptotic critical values.
- control: List with test procedure options used.

References

Rodriguez-Rondon, G., & Dufour, J.-M. 2026a. "Monte Carlo Likelihood-Ratio Tests for Markov Switching Models." *Bank of Canada Staff Working Paper*, No. 2026-23. doi: 10.34989/swp-2026-23.

Examples

```
set.seed(1234)
# Define DGP of MS AR process
mdl_ms2 <- list(n     = 200,
                mu    = c(5,10),
                sigma = c(1,4),
                phi   = c(0.5),
                k     = 2,
                P     = rbind(c(0.90, 0.10),
                              c(0.10, 0.90)))

# Simulate process using simuMSAR() function
y_ms_simu <- simuMSAR(mdl_ms2)
```

```
# ------ MS-AR example ----- #
# Set test procedure options
lmc_control = list(N = 19,
                   burnin = 100,
                   converge_check = NULL,
                   mdl_h0_control = list(const  = TRUE,
                                         getSE  = FALSE),
                   mdl_h1_control = list(msmu   = TRUE,
                                         msvar  = TRUE,
                                         getSE  = FALSE,
                                         method = "EM",
                                         use_diff_init = 1))


# lmctest <- LMCLRTest(y_ms_simu$y, p = 1, k0 = 1 , k1 = 2, control = lmc_control)
# summary(lmctest)
```

`MCpval` *Monte Carlo P-value*

Description

This function computes the Monte Carlo P-value.

Usage

```
MCpval(test_stat, null_vec, type = "geq")
```

Arguments

`test_stat`	Test statistic under the alternative (e.g. `S_0`).
`null_vec`	A (`N x 1`) vector with test statistic under the null hypothesis.
`type`	String determining the type of test. Options are `"geq"` (right/upper-tail test; the default), `"leq"` (left/lower-tail test), `"two-tailed"` (or `"two-tail"`; two-sided test), and `"absolute"` (or `"abs"`; the upper-tail rule, intended for non-negative statistics). An unrecognized value raises an error.

Value

MC p-value of test

References

Dufour, Jean-Marie 2006. "Monte Carlo tests with nuisance parameters: A general approach to finite-sample inference and nonstandard asymptotics". *Journal of Econometrics*, 133(2), 443-477.

Dufour, Jean-Marie, and Richard Luger. 2017. "Identification-robust moment-based tests for Markov switching in autoregressive models". *Econometric Reviews*, 36(6-9), 713-727.

`MMCLRTest` *Maximized Monte Carlo Likelihood Ratio Test*

Description

This function performs the Maximized Monte Carlo likelihood ratio test (MMC-LRT) proposed in Rodriguez-Rondon & Dufour (2026a).

Usage

```
MMCLRTest(Y, p, k0, k1, Z = NULL, control = list())
```

Arguments

`Y`	Series to be tested. Must be a (`T` x `q`) matrix where T is the number of time observations and q is the number of variables.
`p`	Number of autoregressive lags. Must be greater than or equal to 0.
`k0`	Number of regimes under null hypothesis. Must be greater than or equal to 1.
`k1`	Number of regimes under alternative hypothesis. Must be greater than `k0`.
`Z`	Exogenous regressors. Optional input and default is NULL. When used, it should be a (`T` x `qz`) matrix where T is the number of time observations and q is the number of exogenous variables.
`control`	List with test procedure options including: • N: Integer determining the number of Monte Carlo simulations. Default is set to `99` as in paper. • burnin: Number of simulated observations to remove from beginning. Default is `100`. • converge_check: String of NULL determining if convergence of model(s) should be verified. Allowed inputs are: "null", "alt", "both", or `NULL`. If `NULL` (default) no model convergence is verified. • workers: Integer determining the number of workers to use for parallel computing. Default is `0` (sequential). If `workers > N`, the effective count is capped at `N` and an informational message is printed (unless `silence = TRUE`). • mc_seed: Integer seed for reproducible Monte Carlo simulations and fixed-error MMC (Dufour 2006, Prop. 4.2). When set, seeds the RNG before observed-data estimation, pre-draws innovations once and holds them fixed across all optimizer evaluations, ensuring full reproducibility (including the observed LRT statistic) and theoretical size control. Default is `NULL` (non-reproducible). • type: String that determines the type of optimization algorithm used. Arguments allowed are: `"pso"`, `"GenSA"`, and `"GA"`. Default is `"pso"`. • eps: Double determining the constant value that defines a consistent set for search. Default is `0.1`.

- CI_union: Boolean determining if union of set determined by eps and confidence set should be used to define consistent set for search. Default is TRUE.
- lambda: Double determining penalty on nonlinear constraint. Default is 100.
- stationary_constraint: Boolean determining if only stationary solutions are considered (if TRUE) or not (if FALSE). Default is TRUE.
- phi_low: Vector with lower bound for autoregressive parameters when optimizing. Default is NULL.
- phi_upp: Vector with upper bound for autoregressive parameters when optimizing. Default is NULL.
- P_low: Value with lower bound for transition probabilities when optimizing. Default is 0.
- P_upp: Value with upper bound for transition probabilities when optimizing. Default is 1.
- variance_constraint: Double used to determine the lower bound for variance in parameter set for search. Value should be between 0 and 1 as it is multiplied by consistent point estimates of variances. Default is 0.01 (i.e., 1% of consistent point estimates.
- silence: Boolean determining if optimization steps should be silenced (if TRUE) or not (if FALSE). Default is FALSE.
- threshold_stop: Double determining the global optimum of function. Default is 1.
- mdl_h0_control: List with restricted model options. See Nmdl, ARmdl, VARmdl, HMmdl, MSARmdl, or MSVARmdl documentation for available and default values.
- mdl_h1_control: List with unrestricted model options. See HMmdl, MSARmdl, or MSVARmdl documentation for available and default values.
- use_diff_init_sim: Value which determines the number of initial values to use when estimating models for null distribution. Default is set to use the same as specified in mdl_h0_control and mdl_h1_control.
- optim_control: List with optimization algorithm options. See psoptim, GenSA, ga. Default is list() (an empty list); the maximum number of optimizer iterations is instead controlled by the separate maxit argument (default 50).

Value

List of class MMCLRTest (S3 object) with attributes including:

- mdl_h0: List with restricted model attributes.
- mdl_h1: List with unrestricted model attributes.
- mdl_h0_mmc: List with restricted model attributes at the maximizing nuisance parameter values.
- mdl_h1_mmc: List with unrestricted model attributes (same as mdl_h1).
- LRT_0: Value of test statistic from observed data.

- pval: Maximized p-value of the Maximized Monte Carlo Likelihood Ratio Test.
- theta_h0: Maximizing nuisance parameter vector for the restricted model.
- theta_h1: Parameter vector of the unrestricted model.
- control: List with test procedure options used.
- mmc_optimout: Optimization output object returned by the selected optimizer (`pso`, `GenSA`, or `GA`).

References

Rodriguez-Rondon, G., & Dufour, J.-M. 2026a. "Monte Carlo Likelihood-Ratio Tests for Markov Switching Models." *Bank of Canada Staff Working Paper*, No. 2026-23. doi: 10.34989/swp-2026-23.

Examples

```
set.seed(1234)
# Define DGP of MS AR process
mdl_ms2 <- list(n     = 200,
                mu    = c(5,10),
                sigma = c(1,4),
                phi   = c(0.5),
                k     = 2,
                P     = rbind(c(0.90, 0.10),
                              c(0.10, 0.90)))

# Simulate process using simuMSAR() function
y_ms_simu <- simuMSAR(mdl_ms2)


# Set test procedure options
mmc_control = list(N = 19,
                   burnin = 100,
                   converge_check = NULL,
                   eps = 0.1,
                   CI_union = TRUE,
                   silence = FALSE,
                   threshold_stop = 0.05 + 1e-6,
                   type = "pso",
                   maxit = 50,
                   mdl_h0_control = list(const  = TRUE,
                                         getSE  = TRUE),
                   mdl_h1_control = list(msmu   = TRUE,
                                         msvar  = TRUE,
                                         getSE  = TRUE,
                                         method = "EM",
                                         use_diff_init = 1))

# MMCtest <- MMCLRTest(y_ms_simu$y, p = 1 , k0 = 1, k1 = 2, control = mmc_control)
# summary(MMCtest)
```

MSARmdl *Markov-switching autoregressive model*

Description

This function estimates a Markov-switching autoregressive model

Usage

```
MSARmdl(Y, p, k, control = list())
```

Arguments

Y	(T x 1) vector with observational data.
p	integer for the number of lags to use in estimation. Must be greater than or equal to 1.
k	integer for the number of regimes to use in estimation. Must be greater than or equal to 2.
control	List with model options including: • getSE: Boolean. If TRUE standard errors are computed and returned. If FALSE standard errors are not computed. Default is TRUE. • se_method: String determining the standard-error method. Options are "hessian" (default) and "louis" (Louis 1982 expected-information); the Louis method is used automatically as a fallback when the Hessian is ill-conditioned. • msmu: Boolean. If TRUE model is estimated with switch in mean. If FALSE model is estimated with constant mean. Default is TRUE. • msvar: Boolean. If TRUE model is estimated with switch in variance. If FALSE model is estimated with constant variance. Default is TRUE. • init_theta: vector of initial values. vector must contain (1 x q) vector mu, vech(sigma), and vec(P) where sigma is a (q x q) covariance matrix.This is optional. Default is NULL, in which case initVals_MSARmdl is used to generate initial values. • method: string determining which method to use. Options are 'EM' for EM algorithm or 'MLE' for Maximum Likelihood Estimation. Default is 'EM'. • maxit: integer determining the maximum number of EM iterations. • thtol: double determining the convergence criterion for the relative change in the parameter estimates theta between iterations (used when conv is "theta", "both", or "both-A"). Default is 1e-6. • ltol: double determining the convergence criterion for the relative change in the log-likelihood between iterations (used when conv is "loglik", "both", "loglik-A", or "both-A"). Default is 1e-7.

- conv: string determining the EM convergence criterion. "loglik" (the default) stops on the relative change in the log-likelihood; "theta" stops on the relative change in the parameters (the convention of Hamilton (1994) and of earlier versions of this package); "both" requires both, following Krolzig (1997); "loglik-A" and "both-A" replace the log-likelihood test with the Aitken-accelerated criterion of Bohning et al. (1994) and McLachlan and Krishnan (2008). Default is "loglik".
- maxit_converge: integer determining the maximum number of initial values attempted until solution is finite. For example, if parameters in theta or logLike are NaN another set of initial values (up to maxit_converge) is attempted until finite values are returned. This does not occur frequently for most types of data but may be useful in some cases. Once finite values are obtained, this counts as one iteration towards use_diff_init. Default is 500.
- use_diff_init: integer determining how many different initial values to try (that do not return NaN; see maxit_converge). Default is 1.
- mle_stationary_constraint: Boolean determining if only stationary solutions are considered (if TRUE) or not (if FALSE). Default is TRUE.
- mle_variance_constraint: Double used to determine the lower bound for variance in each regime. Value should be between 0 and 1 as it is multiplied by single regime variance. Default is 0.01 (i.e., 1% of single regime variance.
- mle_theta_low: Vector with lower bounds on parameters (Used only if method = "MLE"). Default is NULL.
- mle_theta_upp: Vector with upper bounds on parameters (Used only if method = "MLE"). Default is NULL.

Value

List of class MSARmdl (S3 object) with model attributes including:

- y: a (T x 1) matrix of observations.
- X: a (T-p x p + const) matrix of lagged observations with a leading column of 1s.
- x: a (T-p x p) matrix of lagged observations.
- fitted: a (T x 1) matrix of fitted values.
- resid: a (T x 1) matrix of residuals.
- intercept: a (k x 1) vector of estimated intercepts of each process.
- mu: a (k x 1) vector of estimated means of each process.
- beta: a ((1 + p) x k) matrix of estimated coefficients.
- phi: estimates of autoregressive coefficients.
- stdev: a (k x 1) vector of estimated standard deviation of each process.
- sigma: a (k x 1) estimated covariance matrix.
- theta: vector containing: mu and vech(sigma).
- theta_mu_ind: vector indicating location of mean with 1 and 0 otherwise.

- theta_sig_ind: vector indicating location of variances with 1 and 0 otherwise.
- theta_var_ind: vector indicating location of variances with 1 and 0 otherwise. This is the same as theta_sig_ind in MSARmdl.
- theta_P_ind: vector indicating location of transition matrix elements with 1 and 0 otherwise.
- stationary: Boolean indicating if process is stationary if TRUE or non-stationary if FALSE.
- n: number of observations (same as T).
- p: number of autoregressive lags.
- q: number of series. This is always 1 in MSARmdl.
- k: number of regimes in estimated model.
- P: a (k x k) transition matrix.
- pinf: a (k x 1) vector with limiting probabilities of each regime.
- St: a (T x k) vector with smoothed probabilities of each regime at each time t.
- deltath: double with the maximum relative change in vector theta on the last iteration.
- iterations: number of EM iterations performed to achieve convergence (if less than maxit).
- converged: Boolean. TRUE if the conv convergence criterion was met before maxit iterations, FALSE otherwise.
- theta_0: vector of initial values used.
- init_used: number of different initial values used to get a finite solution. See description of input maxit_converge.
- msmu: Boolean. If TRUE model was estimated with switch in mean. If FALSE model was estimated with constant mean.
- msvar: Boolean. If TRUE model was estimated with switch in variance. If FALSE model was estimated with constant variance.
- control: List with model options used.
- logLike: log-likelihood.
- AIC: Akaike information criterion.
- BIC: Bayesian (Schwarz) information criterion.
- Hess: Hessian matrix. Approximated using hessian and only returned if getSE=TRUE.
- info_mat: Information matrix. Computed as the inverse of -Hess. If matrix is not PD then nearest PD matrix is obtained using nearest_spd. Only returned if getSE=TRUE.
- nearPD_used: Boolean determining whether nearPD function was used on info_mat if TRUE or not if FALSE. Only returned if getSE=TRUE.
- theta_se: standard errors of parameters in theta. Only returned if getSE=TRUE.
- trace: List with Lists of estimation output for each initial value used due to use_diff_init > 1.

References

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society*. Series B 39 (1): 1–38..

Böhning, D., E. Dietz, R. Schaub, P. Schlattmann, and B. G. Lindsay. 1994. "The distribution of the likelihood ratio for mixtures of densities from the one-parameter exponential family." *Annals of the Institute of Statistical Mathematics* 46 (2): 373–388.

McLachlan, G. J., and T. Krishnan. 2008. *The EM Algorithm and Extensions*. 2nd ed. Hoboken, New Jersey: John Wiley & Sons.

Hamilton, James D. 1990. "Analysis of time series subject to changes in regime." *Journal of econometrics*, 45 (1-2): 39–70.

See Also

ARmdl

Examples

```
# --------------------------- Use simulated process -----------------------------
set.seed(1234)
# Define DGP of MS AR process
mdl_ms2 <- list(n     = 500,
                mu    = c(5,10),
                sigma = c(1,4),
                phi   = c(0.5),
                k     = 2,
                P     = rbind(c(0.90, 0.10),
                              c(0.10, 0.90)))

# Simulate process using simuMSAR() function
y_ms_simu <- simuMSAR(mdl_ms2)

# Set options for model estimation
control <- list(msmu   = TRUE,
                msvar  = TRUE,
                method = "EM",
                use_diff_init = 1)

# Estimate model

  ms_mdl <- MSARmdl(y_ms_simu$y, p = 1, k = 2, control = control)
  summary(ms_mdl)
```

MSARXmdl *Markov-switching autoregressive model with exogenous regressors*

Description

This function estimates a Markov-switching autoregressive model with exogenous regressors.

Usage

```
MSARXmdl(Y, p, k, Z, control = list())
```

Arguments

Y	a (T x 1) vector with observational data.
p	integer for the number of lags to use in estimation. Must be greater than or equal to 1.
k	integer for the number of regimes to use in estimation. Must be greater than or equal to 2.
Z	a (T x qz) matrix of exogenous regressors.
control	List with model options including: • getSE: Boolean. If TRUE standard errors are computed and returned. If FALSE standard errors are not computed. Default is TRUE. • se_method: String determining the standard-error method. Options are "hessian" (default) and "louis" (Louis 1982 expected-information); the Louis method is used automatically as a fallback when the Hessian is ill-conditioned. • msmu: Boolean. If TRUE model is estimated with switch in mean. If FALSE model is estimated with constant mean. Default is TRUE. • msvar: Boolean. If TRUE model is estimated with switch in variance. If FALSE model is estimated with constant variance. Default is TRUE. • init_theta: vector of initial values. vector must contain (1 x q) vector mu, vech(sigma), and vec(P) where sigma is a (q x q) covariance matrix.This is optional. Default is NULL, in which case initVals_MSARmdl is used to generate initial values. • method: string determining which method to use. Options are 'EM' for EM algorithm or 'MLE' for Maximum Likelihood Estimation. Default is 'EM'. • maxit: integer determining the maximum number of EM iterations. • thtol: double determining the convergence criterion for the relative change in the parameter estimates theta between iterations (used when conv is "theta", "both", or "both-A"). Default is 1e-6. • ltol: double determining the convergence criterion for the relative change in the log-likelihood between iterations (used when conv is "loglik", "both", "loglik-A", or "both-A"). Default is 1e-7.

- conv: string determining the EM convergence criterion. "loglik" (the default) stops on the relative change in the log-likelihood; "theta" stops on the relative change in the parameters (the convention of Hamilton (1994) and of earlier versions of this package); "both" requires both, following Krolzig (1997); "loglik-A" and "both-A" replace the log-likelihood test with the Aitken-accelerated criterion of Bohning et al. (1994) and McLachlan and Krishnan (2008). Default is "loglik".
- maxit_converge: integer determining the maximum number of initial values attempted until solution is finite. For example, if parameters in theta or logLike are NaN another set of initial values (up to maxit_converge) is attempted until finite values are returned. This does not occur frequently for most types of data but may be useful in some cases. Once finite values are obtained, this counts as one iteration towards use_diff_init. Default is 500.
- use_diff_init: integer determining how many different initial values to try (that do not return NaN; see maxit_converge). Default is 1.
- mle_stationary_constraint: Boolean determining if only stationary solutions are considered (if TRUE) or not (if FALSE). Default is TRUE.
- mle_variance_constraint: Double used to determine the lower bound for variance in each regime. Value should be between 0 and 1 as it is multiplied by single regime variance. Default is 0.01 (i.e., 1% of single regime variance.
- mle_theta_low: Vector with lower bounds on parameters (Used only if method = "MLE"). Default is NULL.
- mle_theta_upp: Vector with upper bounds on parameters (Used only if method = "MLE"). Default is NULL.

Value

List of class MSARXmdl (S3 object) with model attributes including:

- y: a (T x 1) matrix of observations.
- X: a (T-p x p + const) matrix of lagged observations with a leading column of 1s.
- x: a (T-p x p) matrix of lagged observations.
- fitted: a (T x 1) matrix of fitted values.
- resid: a (T x 1) matrix of residuals.
- intercept: a (k x 1) vector of estimated intercepts of each process.
- mu: a (k x 1) vector of estimated means of each process.
- beta: a ((1 + p + qz) x k) matrix of estimated coefficients.
- betaZ: a (qz x q) matrix of estimated exogenous regressor coefficients.
- phi: estimates of autoregressive coefficients.
- stdev: a (k x 1) vector of estimated standard deviation of each process.
- sigma: a (k x 1) estimated covariance matrix.
- theta: vector containing: mu and vech(sigma).

- theta_mu_ind: vector indicating location of mean with 1 and 0 otherwise.
- theta_sig_ind: vector indicating location of variances with 1 and 0 otherwise.
- theta_var_ind: vector indicating location of variances with 1 and 0 otherwise. This is the same as theta_sig_ind in MSARmdl.
- theta_P_ind: vector indicating location of transition matrix elements with 1 and 0 otherwise.
- stationary: Boolean indicating if process is stationary if TRUE or non-stationary if FALSE.
- n: number of observations (same as T).
- p: number of autoregressive lags.
- q: number of series. This is always 1 in MSARmdl.
- k: number of regimes in estimated model.
- P: a (k x k) transition matrix.
- pinf: a (k x 1) vector with limiting probabilities of each regime.
- St: a (T x k) vector with smoothed probabilities of each regime at each time t.
- deltath: double with the maximum relative change in vector theta on the last iteration.
- iterations: number of EM iterations performed to achieve convergence (if less than maxit).
- converged: Boolean. TRUE if the conv convergence criterion was met before maxit iterations, FALSE otherwise.
- theta_0: vector of initial values used.
- init_used: number of different initial values used to get a finite solution. See description of input maxit_converge.
- msmu: Boolean. If TRUE model was estimated with switch in mean. If FALSE model was estimated with constant mean.
- msvar: Boolean. If TRUE model was estimated with switch in variance. If FALSE model was estimated with constant variance.
- control: List with model options used.
- logLike: log-likelihood.
- AIC: Akaike information criterion.
- BIC: Bayesian (Schwarz) information criterion.
- Hess: Hessian matrix. Approximated using hessian and only returned if getSE=TRUE.
- info_mat: Information matrix. Computed as the inverse of -Hess. If matrix is not PD then nearest PD matrix is obtained using nearest_spd. Only returned if getSE=TRUE.
- nearPD_used: Boolean determining whether nearPD function was used on info_mat if TRUE or not if FALSE. Only returned if getSE=TRUE.
- theta_se: standard errors of parameters in theta. Only returned if getSE=TRUE.
- trace: List with Lists of estimation output for each initial value used due to use_diff_init > 1.

References

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society*. Series B 39 (1): 1–38..

Böhning, D., E. Dietz, R. Schaub, P. Schlattmann, and B. G. Lindsay. 1994. "The distribution of the likelihood ratio for mixtures of densities from the one-parameter exponential family." *Annals of the Institute of Statistical Mathematics* 46 (2): 373–388.

McLachlan, G. J., and T. Krishnan. 2008. *The EM Algorithm and Extensions*. 2nd ed. Hoboken, New Jersey: John Wiley & Sons.

Hamilton, James D. 1990. "Analysis of time series subject to changes in regime." *Journal of econometrics*, 45 (1-2): 39–70.

See Also

ARmdl

Examples

```
# --------------------------- Use simulated process ----------------------------
set.seed(1234)
# Define DGP of MS AR process
mdl_ms2 <- list(n     = 500,
                mu    = c(5,10),
                sigma = c(1,4),
                phi   = c(0.5),
                k     = 2,
                P     = rbind(c(0.90, 0.10),
                              c(0.10, 0.90)))

# Simulate process using simuMSAR() function
y_ms_simu <- simuMSAR(mdl_ms2)

# Set options for model estimation
control <- list(msmu   = TRUE,
                msvar  = TRUE,
                method = "EM",
                use_diff_init = 1)

# Estimate model

  ms_mdl <- MSARmdl(y_ms_simu$y, p = 1, k = 2, control = control)
  summary(ms_mdl)
```

MSVARmdl *Markov-switching vector autoregressive model*

Description

This function estimates a Markov-switching vector autoregressive model

Usage

```
MSVARmdl(Y, p, k, control = list())
```

Arguments

Y	(T x q) vector with observational data.
p	integer for the number of lags to use in estimation. Must be greater than or equal to 0.
k	integer for the number of regimes to use in estimation. Must be greater than or equal to 2.
control	List with optimization options including:

- getSE: Boolean. If TRUE standard errors are computed and returned. If FALSE standard errors are not computed. Default is TRUE.
- se_method: String determining the standard-error method. Options are "hessian" (default) and "louis" (Louis 1982 expected-information); the Louis method is used automatically as a fallback when the Hessian is ill-conditioned.
- msmu: Boolean. If TRUE model is estimated with switch in mean. If FALSE model is estimated with constant mean. Default is TRUE.
- msvar: Boolean. If TRUE model is estimated with switch in variance. If FALSE model is estimated with constant variance. Default is TRUE.
- init_theta: vector of initial values. vector must contain (1 x q) vector mu, vech(sigma), and vec(P) where sigma is a (q x q) covariance matrix. This is optional. Default is NULL, in which case initVals_MSARmdl is used to generate initial values.
- method: string determining which method to use. Options are 'EM' for EM algorithm or 'MLE' for Maximum Likelihood Estimation. Default is 'EM'.
- maxit: integer determining the maximum number of EM iterations.
- thtol: double determining the convergence criterion for the relative change in the parameter estimates theta between iterations (used when conv is "theta", "both", or "both-A"). Default is 1e-6.
- ltol: double determining the convergence criterion for the relative change in the log-likelihood between iterations (used when conv is "loglik", "both", "loglik-A", or "both-A"). Default is 1e-7.

- conv: string determining the EM convergence criterion. "loglik" (the default) stops on the relative change in the log-likelihood; "theta" stops on the relative change in the parameters (the convention of Hamilton (1994) and of earlier versions of this package); "both" requires both, following Krolzig (1997); "loglik-A" and "both-A" replace the log-likelihood test with the Aitken-accelerated criterion of Bohning et al. (1994) and McLachlan and Krishnan (2008). Default is "loglik".
- maxit_converge: integer determining the maximum number of initial values attempted until solution is finite. For example, if parameters in theta or logLike are NaN another set of initial values (up to maxit_converge) is attempted until finite values are returned. This does not occur frequently for most types of data but may be useful in some cases. Once finite values are obtained, this counts as one iteration towards use_diff_init. Default is 500.
- use_diff_init: integer determining how many different initial values to try (that do not return NaN; see maxit_converge). Default is 1.
- mle_stationary_constraint: Boolean determining if only stationary solutions are considered (if TRUE) or not (if FALSE). Default is TRUE.
- mle_variance_constraint: double used to determine the lower bound on the smallest eigenvalue for the covariance matrix of each regime. Default is 1e-3.
- mle_theta_low: Vector with lower bounds on parameters (Used only if method = "MLE"). Default is NULL.
- mle_theta_upp: Vector with upper bounds on parameters (Used only if method = "MLE"). Default is NULL.

Value

List of class MSVARmdl (S3 object) with model attributes including:

- y: a (T-p x q) matrix of observations.
- X: a (T-p x p*q + const) matrix of lagged observations with a leading column of 1s.
- x: a (T-p x p*q) matrix of lagged observations.
- fitted: a (T x q) matrix of fitted values.
- resid: a (T-p x q) matrix of residuals.
- intercept: a (k x q) matrix of estimated intercepts of each process.
- mu: a (k x q) matrix of estimated means of each process.
- beta: a list containing k separate ((1 + p) x q) matrix of estimated coefficients for each regime.
- phi: estimates of autoregressive coefficients.
- Fmat: Companion matrix containing autoregressive coefficients.
- stdev: List with k (q x q) matrices with estimated standard deviation on the diagonal.
- sigma: List with k (q x q) matrices with estimated covariance matrix.
- theta: vector containing: mu and vech(sigma).

- theta_mu_ind: vector indicating location of mean with 1 and 0 otherwise.
- theta_sig_ind: vector indicating location of variance and covariances with 1 and 0 otherwise.
- theta_var_ind: vector indicating location of variances with 1 and 0 otherwise.
- theta_P_ind: vector indicating location of transition matrix elements with 1 and 0 otherwise.
- stationary: Boolean indicating if process is stationary if TRUE or non-stationary if FALSE.
- n: number of observations (same as T).
- p: number of autoregressive lags.
- q: number of series.
- k: number of regimes in estimated model.
- P: a (k x k) transition matrix.
- pinf: a (k x 1) vector with limiting probabilities of each regime.
- St: a (T x k) vector with smoothed probabilities of each regime at each time t.
- deltath: double with the maximum relative change in vector theta on the last iteration.
- iterations: number of EM iterations performed to achieve convergence (if less than maxit).
- converged: Boolean. TRUE if the conv convergence criterion was met before maxit iterations, FALSE otherwise.
- theta_0: vector of initial values used.
- init_used: number of different initial values used to get a finite solution. See description of input maxit_converge.
- msmu: Boolean. If TRUE model was estimated with switch in mean. If FALSE model was estimated with constant mean.
- msvar: Boolean. If TRUE model was estimated with switch in variance. If FALSE model was estimated with constant variance.
- control: List with model options used.
- logLike: log-likelihood.
- AIC: Akaike information criterion.
- BIC: Bayesian (Schwarz) information criterion.
- Hess: Hessian matrix. Approximated using hessian and only returned if getSE=TRUE.
- info_mat: Information matrix. Computed as the inverse of -Hess. If matrix is not PD then nearest PD matrix is obtained using nearest_spd. Only returned if getSE=TRUE.
- nearPD_used: Boolean determining whether nearPD function was used on info_mat if TRUE or not if FALSE. Only returned if getSE=TRUE.
- theta_se: standard errors of parameters in theta. Only returned if getSE=TRUE.
- trace: List with Lists of estimation output for each initial value used due to use_diff_init > 1.

References

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society*. Series B 39 (1): 1–38..

Böhning, D., E. Dietz, R. Schaub, P. Schlattmann, and B. G. Lindsay. 1994. "The distribution of the likelihood ratio for mixtures of densities from the one-parameter exponential family." *Annals of the Institute of Statistical Mathematics* 46 (2): 373–388.

McLachlan, G. J., and T. Krishnan. 2008. *The EM Algorithm and Extensions*. 2nd ed. Hoboken, New Jersey: John Wiley & Sons.

Krolzig, Hans-Martin. 1997. "The markov-switching vector autoregressive model.". Springer.

See Also

VARmdl

Examples

```
set.seed(123)
# Define DGP of MS VAR process
mdl_msvar2 <- list(n     = 200,
                   p     = 1,
                   q     = 2,
                   mu    = rbind(c(5, -2),
                                 c(10, 2)),
                   sigma = list(rbind(c(5.0, 1.5),
                                      c(1.5, 1.0)),
                                rbind(c(7.0, 3.0),
                                      c(3.0, 2.0))),
                   phi   = rbind(c(0.50, 0.30),
                                 c(0.20, 0.70)),
                   k     = 2,
                   P     = rbind(c(0.90, 0.10),
                                 c(0.10, 0.90)))

# Simulate process using simuMSVAR() function
y_msvar_simu <- simuMSVAR(mdl_msvar2)

# Set options for model estimation
control <- list(msmu   = TRUE,
                msvar  = TRUE,
                method = "EM",
                use_diff_init = 1)

# Estimate model

  y_msvar_mdl <- MSVARmdl(y_msvar_simu$y, p = 1, k = 2, control = control)
  summary(y_msvar_mdl)
```

MSVARXmdl *Markov-switching vector autoregressive model with exogenous regressors*

Description

This function estimates a Markov-switching vector autoregressive model with exogenous regressors.

Usage

```
MSVARXmdl(Y, p, k, Z, control = list())
```

Arguments

Y	(T x q) vector with observational data.
p	integer for the number of lags to use in estimation. Must be greater than or equal to 0.
k	integer for the number of regimes to use in estimation. Must be greater than or equal to 2.
Z	a (T x qz) matrix of exogenous regressors.
control	List with optimization options including: • getSE: Boolean. If TRUE standard errors are computed and returned. If FALSE standard errors are not computed. Default is TRUE. • se_method: String determining the standard-error method. Options are "hessian" (default) and "louis" (Louis 1982 expected-information); the Louis method is used automatically as a fallback when the Hessian is ill-conditioned. • msmu: Boolean. If TRUE model is estimated with switch in mean. If FALSE model is estimated with constant mean. Default is TRUE. • msvar: Boolean. If TRUE model is estimated with switch in variance. If FALSE model is estimated with constant variance. Default is TRUE. • init_theta: vector of initial values. vector must contain (1 x q) vector mu, vech(sigma), and vec(P) where sigma is a (q x q) covariance matrix. This is optional. Default is NULL, in which case initVals_MSARmdl is used to generate initial values. • method: string determining which method to use. Options are 'EM' for EM algorithm or 'MLE' for Maximum Likelihood Estimation. Default is 'EM'. • maxit: integer determining the maximum number of EM iterations. • thtol: double determining the convergence criterion for the relative change in the parameter estimates theta between iterations (used when conv is "theta", "both", or "both-A"). Default is 1e-6. • ltol: double determining the convergence criterion for the relative change in the log-likelihood between iterations (used when conv is "loglik", "both", "loglik-A", or "both-A"). Default is 1e-7.

- conv: string determining the EM convergence criterion. "loglik" (the default) stops on the relative change in the log-likelihood; "theta" stops on the relative change in the parameters (the convention of Hamilton (1994) and of earlier versions of this package); "both" requires both, following Krolzig (1997); "loglik-A" and "both-A" replace the log-likelihood test with the Aitken-accelerated criterion of Bohning et al. (1994) and McLachlan and Krishnan (2008). Default is "loglik".
- maxit_converge: integer determining the maximum number of initial values attempted until solution is finite. For example, if parameters in theta or logLike are NaN another set of initial values (up to maxit_converge) is attempted until finite values are returned. This does not occur frequently for most types of data but may be useful in some cases. Once finite values are obtained, this counts as one iteration towards use_diff_init. Default is 500.
- use_diff_init: integer determining how many different initial values to try (that do not return NaN; see maxit_converge). Default is 1.
- mle_stationary_constraint: Boolean determining if only stationary solutions are considered (if TRUE) or not (if FALSE). Default is TRUE.
- mle_variance_constraint: double used to determine the lower bound on the smallest eigenvalue for the covariance matrix of each regime. Default is 1e-3.
- mle_theta_low: Vector with lower bounds on parameters (Used only if method = "MLE"). Default is NULL.
- mle_theta_upp: Vector with upper bounds on parameters (Used only if method = "MLE"). Default is NULL.

Value

List of class MSVARXmdl (S3 object) with model attributes including:

- y: a (T-p x q) matrix of observations.
- X: a (T-p x p*q + const) matrix of lagged observations with a leading column of 1s.
- x: a (T-p x p*q) matrix of lagged observations.
- resid: a (T-p x q) matrix of residuals.
- fitted: a (T x q) matrix of fitted values.
- intercept: a (k x q) matrix of estimated intercepts of each process.
- mu: a (k x q) matrix of estimated means of each process.
- beta: a list containing k separate ((1 + p + qz) x q) matrix of estimated coefficients for each regime.
- betaZ: a (qz x q) matrix of estimated exogenous regressor coefficients.
- phi: estimates of autoregressive coefficients.
- Fmat: Companion matrix containing autoregressive coefficients.
- stdev: List with k (q x q) matrices with estimated standard deviation on the diagonal.
- sigma: List with k (q x q) matrices with estimated covariance matrix.

- theta: vector containing: mu and vech(sigma).
- theta_mu_ind: vector indicating location of mean with 1 and 0 otherwise.
- theta_sig_ind: vector indicating location of variance and covariances with 1 and 0 otherwise.
- theta_var_ind: vector indicating location of variances with 1 and 0 otherwise.
- theta_P_ind: vector indicating location of transition matrix elements with 1 and 0 otherwise.
- stationary: Boolean indicating if process is stationary if TRUE or non-stationary if FALSE.
- n: number of observations (same as T).
- p: number of autoregressive lags.
- q: number of series.
- k: number of regimes in estimated model.
- P: a (k x k) transition matrix.
- pinf: a (k x 1) vector with limiting probabilities of each regime.
- St: a (T x k) vector with smoothed probabilities of each regime at each time t.
- deltath: double with the maximum relative change in vector theta on the last iteration.
- iterations: number of EM iterations performed to achieve convergence (if less than maxit).
- converged: Boolean. TRUE if the conv convergence criterion was met before maxit iterations, FALSE otherwise.
- theta_0: vector of initial values used.
- init_used: number of different initial values used to get a finite solution. See description of input maxit_converge.
- msmu: Boolean. If TRUE model was estimated with switch in mean. If FALSE model was estimated with constant mean.
- msvar: Boolean. If TRUE model was estimated with switch in variance. If FALSE model was estimated with constant variance.
- control: List with model options used.
- logLike: log-likelihood.
- AIC: Akaike information criterion.
- BIC: Bayesian (Schwarz) information criterion.
- Hess: Hessian matrix. Approximated using hessian and only returned if getSE=TRUE.
- info_mat: Information matrix. Computed as the inverse of -Hess. If matrix is not PD then nearest PD matrix is obtained using nearest_spd. Only returned if getSE=TRUE.
- nearPD_used: Boolean determining whether nearPD function was used on info_mat if TRUE or not if FALSE. Only returned if getSE=TRUE.
- theta_se: standard errors of parameters in theta. Only returned if getSE=TRUE.
- trace: List with Lists of estimation output for each initial value used due to use_diff_init > 1.

References

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society*. Series B 39 (1): 1–38..

Böhning, D., E. Dietz, R. Schaub, P. Schlattmann, and B. G. Lindsay. 1994. "The distribution of the likelihood ratio for mixtures of densities from the one-parameter exponential family." *Annals of the Institute of Statistical Mathematics* 46 (2): 373–388.

McLachlan, G. J., and T. Krishnan. 2008. *The EM Algorithm and Extensions*. 2nd ed. Hoboken, New Jersey: John Wiley & Sons.

Krolzig, Hans-Martin. 1997. "The markov-switching vector autoregressive model.". Springer.

See Also

VARmdl

Examples

```
set.seed(123)
# Define DGP of MS VAR process
mdl_msvar2 <- list(n     = 200,
                   p     = 1,
                   q     = 2,
                   mu    = rbind(c(5, -2),
                                 c(10, 2)),
                   sigma = list(rbind(c(5.0, 1.5),
                                      c(1.5, 1.0)),
                                rbind(c(7.0, 3.0),
                                      c(3.0, 2.0))),
                   phi   = rbind(c(0.50, 0.30),
                                 c(0.20, 0.70)),
                   k     = 2,
                   P     = rbind(c(0.90, 0.10),
                                 c(0.10, 0.90)))

# Simulate process using simuMSVAR() function
y_msvar_simu <- simuMSVAR(mdl_msvar2)

# Set options for model estimation
control <- list(msmu   = TRUE,
                msvar  = TRUE,
                method = "EM",
                use_diff_init = 1)

# Estimate model

  y_msvar_mdl <- MSVARmdl(y_msvar_simu$y, p = 1, k = 2, control = control)
  summary(y_msvar_mdl)
```

Nmdl *Normal distribution model*

Description

This function estimates a univariate or multivariate normally distributed model. This can be used for the null hypothesis of a linear model against an alternative hypothesis of a HMM with k regimes.

Usage

```
Nmdl(Y, Z = NULL, control = list())
```

Arguments

Y	a (T x q) matrix of observations.
Z	an otpional (T x qz) matrix of exogenous regressors. Default is NULL.
control	List with model options including: • const: Boolean determining whether to estimate model with constant if TRUE or not if FALSE. Default is TRUE. • getSE: Boolean determining whether to compute standard errors of parameters if TRUE or not if FALSE. Default is TRUE.

Value

List of class Nmdl (S3 object) with model attributes including:

- y: a (T x q) matrix of observations.
- fitted: a (T x q) matrix of fitted values.
- resid: a (T x q) matrix of residuals.
- mu: a (1 x q) vector of estimated means of each process.
- intercept: a (1 x q) vector of estimated intercept of each process. If Z is NULL, it is the same as mu.
- beta: a ((1 + p + qz) x q) matrix of estimated coefficients.
- betaZ: a (qz x q) matrix of estimated exogenous regressor coefficients (If Z is provided).
- stdev: a (q x 1) vector of estimated standard deviation of each process.
- sigma: a (q x q) estimated covariance matrix.
- theta: vector containing: mu, betaZ (if matrix Z is provided), and vech(sigma).
- theta_mu_ind: vector indicating location of mean with 1 and 0 otherwise.
- theta_sig_ind: vector indicating location of variance and covariances with 1 and 0 otherwise.
- theta_var_ind: vector indicating location of variances with 1 and 0 otherwise.
- n: number of observations (same as T).
- q: number of series.

- k: number of regimes. This is always 1 in Nmdl.
- control: List with model options used.
- logLike: log-likelihood.
- AIC: Akaike information criterion.
- BIC: Bayesian (Schwarz) information criterion.
- Hess: Hessian matrix. Approximated using hessian and only returned if getSE=TRUE.
- info_mat: Information matrix. Computed as the inverse of -Hess. If matrix is not PD then nearest PD matrix is obtained using nearest_spd. Only returned if getSE=TRUE.
- nearPD_used: Boolean determining whether nearPD function was used on info_mat if TRUE or not if FALSE. Only returned if getSE=TRUE.
- theta_se: standard errors of parameters in theta. Only returned if getSE=TRUE.

Examples

```
set.seed(1234)

# ----- Univariate ----- #
# Define DGP
mdl_norm <- list(n     = 1000,
                 q     = 1,
                 mu    = as.matrix(5),
                 sigma = as.matrix(5.0))

# Simulate process using simuNorm() function
y_norm_simu <- simuNorm(mdl_norm)

# estimate parameters
y_norm_mdl <- Nmdl(y_norm_simu$y)
summary(y_norm_mdl)


# ----- Multivariate ----- #
# Define DGP
mdl_norm <- list(n     = 1000,
                 q     = 2,
                 mu    = c(5, -2),
                 sigma = rbind(c(5.0, 1.5),
                               c(1.5, 1.0)))

# Simulate process using simuNorm() function
y_norm_simu <- simuNorm(mdl_norm)

# estimate parameters
y_norm_mdl <- Nmdl(y_norm_simu$y)
summary(y_norm_mdl)
```

simuAR *Simulate autoregressive process*

Description

This function simulates an autoregresive process.

Usage

```
simuAR(mdl_h0, burnin = 100)
```

Arguments

mdl_h0	List containing the following DGP parameters • n: Length of series. • mu: Mean of process. • sigma: Standard deviation of process. • phi: Vector of autoregressive coefficients. • eps: An optional (T+burnin x q) matrix with standard normal errors to be used. Errors will be generated if not provided.
burnin	Number of simulated observations to remove from beginning. Default is 100.

Value

List with simulated autoregressive series and its DGP parameters.

Examples

```
set.seed(1234)
# Define DGP of AR process
mdl_ar <- list(n     = 500,
               mu    = 5,
               sigma = 2,
               phi   = c(0.5,0.2))

# Simulate process using simuAR() function
y_simu <- simuAR(mdl_ar)

plot(y_simu)
```

simuARX *Simulate autoregressive X process*

Description

This function simulates an autoregresive X process.

Usage

```
simuARX(mdl_h0, burnin = 100)
```

Arguments

mdl_h0	List containing the following DGP parameters • n: Length of series. • mu: Mean of process. • sigma: Standard deviation of process. • phi: Vector of autoregressive coefficients. • eps: An optional (T+burnin x q) matrix with standard normal errors to be used. Errors will be generated if not provided. • Z: A (T x qz) matrix with exogenous regressors (Optional) and where qz is the number of exogenous variables. • betaZ: A (qz x 1) matrix true coefficients on exogenous regressors (Optional) and where qz is the number of exogenous variables.
burnin	Number of simulated observations to remove from beginning. Default is 100.

Value

List with simulated autoregressive series and its DGP parameters.

Examples

```
set.seed(1234)
# Define DGP of AR process
mdl_ar <- list(n     = 500,
               mu    = 5,
               sigma = 2,
               phi   = c(0.5,0.2))

# Simulate process using simuAR() function
y_simu <- simuAR(mdl_ar)

plot(y_simu)
```

simuHMM *Simulate Hidden Markov model with normally distributed errors*

Description

This function simulates a Hidden Markov Model process.

Usage

```
simuHMM(mdl_h0, burnin = 100)
```

Arguments

mdl_h0	List containing the following DGP parameters • n: Length of series. • k: Number of regimes. • mu: A (k x q) vector of means. • sigma: A (q x q) covariance matrix. • q: Number of series. • P: A (k x k) transition matrix (columns must sum to one). • eps: An optional (T+burnin x q) matrix with standard normal errors to be used. Errors will be generated if not provided. • Z: A (T x qz) matrix with exogenous regressors (Optional) and where qz is the number of exogenous variables. • betaZ: A (qz x q) matrix true coefficients on exogenous regressors (Optional) and where qz is the number of exogenous variables.
burnin	Number of simulated observations to remove from beginning. Default is 100.

Value

List with simulated series and its DGP parameters.

Examples

```
set.seed(1234)

# ----- Univariate ----- #
# Define DGP
mdl_hmm <- list(n     = 1000,
                q     = 1,
                mu    = as.matrix(c(5,
                                    -2)),
                sigma = list(as.matrix(5.0),
                             as.matrix(7.0)),
                k     = 2,
                P     = rbind(c(0.90, 0.10),
```

```
                              c(0.10, 0.90)))

# Simulate process using simuHMM() function
y_hmm_simu <- simuHMM(mdl_hmm)

plot(y_hmm_simu)

# ----- Multivariate ----- #
# Define DGP
mdl_hmm <- list(n     = 1000,
                q     = 2,
                mu    = rbind(c(5, -2),
                              c(10, 2)),
                sigma = list(rbind(c(5.0, 1.5),
                                   c(1.5, 1.0)),
                             rbind(c(7.0, 3.0),
                                   c(3.0, 2.0))),
                k     = 2,
                P     = rbind(c(0.90, 0.10),
                              c(0.10, 0.90)))

# Simulate process using simuHMM() function
y_hmm_simu <- simuHMM(mdl_hmm)


plot(y_hmm_simu)
```

`simuMSAR` *Simulate Markov-switching autoregressive process*

Description

This function simulates a Markov-switching autoregressive process.

Usage

```
simuMSAR(mdl_h0, burnin = 100)
```

Arguments

`mdl_h0`	List containing the following DGP parameters • n: Length of series. • k: Number of regimes. • mu: A (k x 1) vector with mean of process in each regime. • sigma: A (k x 1) vector with standard deviation of process in each regime. • phi: Vector of autoregressive coefficients. • P: A (k x k) transition matrix (columns must sum to one). • eps: An optional (T+burnin x q) matrix with standard normal errors to be used. Errors will be generated if not provided.
`burnin`	Number of simulated observations to remove from beginning. Default is 100.

Value

List with simulated Markov-switching autoregressive process and its DGP properties.

Examples

```
set.seed(1234)
# Define DGP of MS AR process
mdl_ms2 <- list(n     = 500,
                mu    = c(5,10),
                sigma = c(1,2),
                phi   = c(0.5, 0.2),
                k     = 2,
                P     = rbind(c(0.90, 0.10),
                              c(0.10, 0.90)))

# Simulate process using simuMSAR() function
y_ms_simu <- simuMSAR(mdl_ms2)

plot(y_ms_simu)
```

---

simuMSARX *Simulate Markov-switching ARX process*

Description

This function simulates a Markov-switching autoregressive process.

Usage

```
simuMSARX(mdl_h0, burnin = 100)
```

Arguments

mdl_h0	List containing the following DGP parameters • n: Length of series. • k: Number of regimes. • mu: A (k x 1) vector with mean of process in each regime. • sigma: A (k x 1) vector with standard deviation of process in each regime. • phi: Vector of autoregressive coefficients. • P: A (k x k) transition matrix (columns must sum to one). • eps: An optional (T+burnin x q) matrix with standard normal errors to be used. Errors will be generated if not provided. • Z: A (T x qz) matrix with exogenous regressors (Optional) and where qz is the number of exogenous variables. • betaZ: A (qz x 1) matrix true coefficients on exogenous regressors (Optional) and where qz is the number of exogenous variables.
burnin	Number of simulated observations to remove from beginning. Default is 100.

Value

List with simulated Markov-switching autoregressive process and its DGP properties.

Examples

```
set.seed(1234)
# Define DGP of MS AR process
mdl_ms2 <- list(n     = 500,
                mu    = c(5,10),
                sigma = c(1,2),
                phi   = c(0.5, 0.2),
                k     = 2,
                P     = rbind(c(0.90, 0.10),
                              c(0.10, 0.90)))

# Simulate process using simuMSAR() function
y_ms_simu <- simuMSAR(mdl_ms2)

plot(y_ms_simu)
```

---

simuMSVAR *Simulate Markov-switching vector autoregressive process*

Description

This function simulates a Markov-switching vector autoregressive process.

Usage

```
simuMSVAR(mdl_h0, burnin = 100)
```

Arguments

mdl_h0	List containing the following DGP parameters • n: Length of series. • k: Number of regimes. • mu: A (k x q) matrix of means. • sigma: List with k (q x q) covariance matrices. • phi: A (q x qp) matrix of autoregressive coefficients. • p: Number of autoregressive lags. • q: Number of series. • P: A (k x k) transition matrix (columns must sum to one). • eps: An optional (T+burnin x q) matrix with standard normal errors to be used. Errors will be generated if not provided.
burnin	Number of simulated observations to remove from beginning. Default is 100.

Value

List with simulated vector autoregressive series and its DGP parameters.

Examples

```
set.seed(1234)
# Define DGP of MS VAR process
mdl_msvar2 <- list(n     = 1000,
                   p     = 1,
                   q     = 2,
                   mu    = rbind(c(5, -2),
                                 c(10, 2)),
                   sigma = list(rbind(c(5.0, 1.5),
                                      c(1.5, 1.0)),
                                rbind(c(7.0, 3.0),
                                      c(3.0, 2.0))),
                   phi   = rbind(c(0.50, 0.30),
                                 c(0.20, 0.70)),
                   k     = 2,
                   P     = rbind(c(0.90, 0.10),
                                 c(0.10, 0.90)))

# Simulate process using simuMSVAR() function
y_msvar_simu <- simuMSVAR(mdl_msvar2)

plot(y_msvar_simu)
```

---

simuMSVARX *Simulate Markov-switching VARX process*

Description

This function simulates a Markov-switching VARX process.

Usage

```
simuMSVARX(mdl_h0, burnin = 100)
```

Arguments

mdl_h0	List containing the following DGP parameters • n: Length of series. • k: Number of regimes. • mu: A (k x q) matrix of means. • sigma: List with k (q x q) covariance matrices. • phi: A (q x qp) matrix of autoregressive coefficients. • p: Number of autoregressive lags.

- q: Number of series.
- P: A (k x k) transition matrix (columns must sum to one).
- eps: An optional (T+burnin x q) matrix with standard normal errors to be used. Errors will be generated if not provided.
- Z: A (T x qz) matrix with exogenous regressors (Optional) and where qz is the number of exogenous variables.
- betaZ: A (qz x q) matrix true coefficients on exogenous regressors (Optional) and where qz is the number of exogenous variables.

burnin Number of simulated observations to remove from beginning. Default is 100.

Value

List with simulated vector autoregressive series and its DGP parameters.

Examples

```
set.seed(1234)
# Define DGP of MS VAR process
mdl_msvar2 <- list(n     = 1000,
                   p     = 1,
                   q     = 2,
                   mu    = rbind(c(5, -2),
                                 c(10, 2)),
                   sigma = list(rbind(c(5.0, 1.5),
                                      c(1.5, 1.0)),
                                rbind(c(7.0, 3.0),
                                      c(3.0, 2.0))),
                   phi   = rbind(c(0.50, 0.30),
                                 c(0.20, 0.70)),
                   k     = 2,
                   P     = rbind(c(0.90, 0.10),
                                 c(0.10, 0.90)))

# Simulate process using simuMSVAR() function
y_msvar_simu <- simuMSVAR(mdl_msvar2)

plot(y_msvar_simu)
```

---

simuNorm *Simulate normally distributed process*

Description

This function simulates a normally distributed process.

Usage

```
simuNorm(mdl_h0, burnin = 0)
```

Arguments

mdl_h0	List containing the following DGP parameters • n: Length of series. • mu: A (q x 1) vector of means. • sigma: A (q x q) covariance matrix. • q: Number of series. • eps: An optional (T+burnin x q) matrix with standard normal errors to be used. Errors will be generated if not provided. • Z: A (T x qz) matrix with exogenous regressors (Optional) and where qz is the number of exogenous variables. • betaZ: A (qz x q) matrix true coefficients on exogenous regressors (Optional) and where qz is the number of exogenous variables.
burnin	Number of simulated observations to remove from beginning. Default is 0.

Value

List with simulated series and its DGP parameters.

Examples

```
set.seed(1234)
# Define DGP
mdl_norm <- list(n     = 1000,
                 q     = 2,
                 mu    = c(5, -2),
                 sigma = rbind(c(5.0, 1.5),
                               c(1.5, 1.0)))

# Simulate process using simuNorm() function
y_norm_simu <- simuNorm(mdl_norm)

plot(y_norm_simu)
```

---

simuVAR *Simulate VAR process*

Description

This function simulates a vector autoregresive process.

Usage

```
simuVAR(mdl_h0, burnin = 100)
```

Arguments

mdl_h0	List containing the following DGP parameters • n: Length of series. • mu: A (q x 1) vector of means. • sigma: A (q x q) covariance matrix. • phi: A (q x qp) matrix of autoregressive coefficients. • p: Number of autoregressive lags. • q: Number of series. • eps: An optional (T+burnin x q) matrix with standard normal errors to be used. Errors will be generated if not provided.
burnin	Number of simulated observations to remove from beginning. Default is 100.

Value

List with simulated vector autoregressive series and its DGP parameters.

Examples

```
set.seed(1234)
# Define DGP of VAR process
mdl_3var2 <- list(n     = 1000,
                  p     = 2,
                  q     = 3,
                  mu    = c(5, -2, 1),
                  sigma = rbind(c(5.0, 1.5, 2.5),
                                c(1.5, 1.0, 1.5),
                                c(2.5, 1.5, 4.2)),
                  phi   = rbind(c(0.70, 0.30, 0.35,  -0.50, -0.20,   0.25),
                                c(0.20, 0.40, 0.35,  -0.30,  0.30,   0.25),
                                c(0.20, 0.30, 0.50,  -0.30, -0.20,  -0.40)))

# Simulate process using simuVAR() function
y3var2_simu <- simuVAR(mdl_3var2)

plot(y3var2_simu)
```

---

simuVARX *Simulate VARX process*

Description

This function simulates a vector autoregressive process with exogenous regressors.

Usage

```
simuVARX(mdl_h0, burnin = 100)
```

Arguments

mdl_h0	List containing the following DGP parameters • n: Length of series. • mu: A (q x 1) vector of means. • sigma: A (q x q) covariance matrix. • phi: A (q x (q x p)) matrix of autoregressive coefficients. • p: Number of autoregressive lags. • q: Number of series. • eps: An optional (T+burnin x q) matrix with standard normal errors to be used. Errors will be generated if not provided. • Z: A (T x qz) matrix with exogenous regressors (Optional) and where qz is the number of exogenous variables. • betaZ: A (qz x q) matrix true coefficients on exogenous regressors (Optional) and where qz is the number of exogenous variables.
burnin	Number of simulated observations to remove from beginning. Default is 100.

Value

List with simulated vector autoregressive series and its DGP parameters.

Examples

```
set.seed(1234)
# Define DGP of VAR process
mdl_3var2 <- list(n     = 1000,
                  p     = 2,
                  q     = 3,
                  mu    = c(5, -2, 1),
                  sigma = rbind(c(5.0, 1.5, 2.5),
                                c(1.5, 1.0, 1.5),
                                c(2.5, 1.5, 4.2)),
                  phi   = rbind(c(0.70, 0.30, 0.35,  -0.50, -0.20,   0.25),
                                c(0.20, 0.40, 0.35,  -0.30,  0.30,   0.25),
                                c(0.20, 0.30, 0.50,  -0.30, -0.20,  -0.40)))

# Simulate process using simuVAR() function
y3var2_simu <- simuVAR(mdl_3var2)

plot(y3var2_simu)
```

USGNP *US GNP data 1947Q2 - 2024Q2*

Description

This data is used in Rodriguez-Rondon & Dufour (2026a). The series ranges from 1947Q2 to 2024Q2.

Usage

```
USGNP
```

Format

Date Vector of dates

GNP US GNP series

GNP_gr log difference of US GNP series

Source

https://fred.stlouisfed.org/graph/?g=UKDQ

References

U.S. Bureau of Economic Analysis, Gross National Product [GNP], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/GNP, September, 2024.

Rodriguez-Rondon, G., & Dufour, J.-M. 2026a. "Monte Carlo Likelihood-Ratio Tests for Markov Switching Models." *Bank of Canada Staff Working Paper*, No. 2026-23. doi: 10.34989/swp-2026-23.

USRGDP *US Real GDP data 1947Q2 - 2024Q2*

Description

This data is used in Rodriguez-Rondon & Dufour (2026a). The series ranges from 1947Q2 to 2024Q2.

Usage

```
USRGDP
```

Format

Date Vector of dates

RGDP US Real GDP series

RGDP_gr log difference of US Real GDP series

Source

https://fred.stlouisfed.org/series/GDPC1

References

U.S. Bureau of Economic Analysis, Real Gross Domestic Product [GDPC1], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/GDPC1, September, 2024.

Rodriguez-Rondon, G., & Dufour, J.-M. 2026a. "Monte Carlo Likelihood-Ratio Tests for Markov Switching Models." *Bank of Canada Staff Working Paper*, No. 2026-23. doi: 10.34989/swp-2026-23.

`VARmdl` *Vector autoregressive model*

Description

This function estimates a vector autoregresive model with `p` lags. This can be used for the null hypothesis of a linear model against an alternative hypothesis of a Markov switching vector autoregressive model with `k` regimes.

Usage

```
VARmdl(Y, p, control = list())
```

Arguments

`Y`	a `(T x q)` matrix of observations.
`p`	integer determining the number of autoregressive lags.
`control`	List with model options including: • const: Boolean determining whether to estimate model with constant if `TRUE` or not if `FALSE`. Default is `TRUE`. • getSE: Boolean determining whether to compute standard errors of parameters if `TRUE` or not if `FALSE`. Default is `TRUE`.

Value

List of class `VARmdl` (`S3` object) with model attributes including:

- y: a `(T-p x q)` matrix of observations.
- X: a `(T-p x p*q + const)` matrix of lagged observations with a leading column of `1`s if `const=TRUE` or not if `const=FALSE`.
- x: a `(T-p x p*q)` matrix of lagged observations.
- fitted: a `(T-p x q)` matrix of fitted values.
- resid: a `(T-p x q)` matrix of residuals.
- mu: a `(1 x q)` vector of estimated means of each process.
- beta: a `((1 + p) x q)` matrix of estimated coefficients. .
- intercept: estimate of intercepts.

- phi: a (q x p*q) matrix of estimated autoregressive coefficients.
- Fmat: Companion matrix containing autoregressive coefficients.
- stdev: a (q x 1) vector of estimated standard deviation of each process.
- sigma: a (q x q) estimated covariance matrix.
- theta: vector containing: mu, vech(sigma), and vec(t(phi)).
- theta_mu_ind: vector indicating location of mean with 1 and 0 otherwise.
- theta_sig_ind: vector indicating location of variance and covariances with 1 and 0 otherwise.
- theta_var_ind: vector indicating location of variances with 1 and 0 otherwise.
- theta_phi_ind: vector indicating location of autoregressive coefficients with 1 and 0 otherwise.
- stationary: Boolean indicating if process is stationary if TRUE or non-stationary if FALSE.
- n: number of observations after lag transformation (i.e., n = T-p).
- p: number of autoregressive lags.
- q: number of series.
- k: number of regimes. This is always 1 in VARmdl.
- Fmat: matrix from companion form. Used to determine is process is stationary.
- control: List with model options used.
- logLike: log-likelihood.
- AIC: Akaike information criterion.
- BIC: Bayesian (Schwarz) information criterion.
- Hess: Hessian matrix. Approximated using hessian and only returned if getSE=TRUE.
- info_mat: Information matrix. Computed as the inverse of -Hess. If matrix is not PD then nearest PD matrix is obtained using nearest_spd. Only returned if getSE=TRUE.
- nearPD_used: Boolean determining whether nearPD function was used on info_mat if TRUE or not if FALSE. Only returned if getSE=TRUE.
- theta_se: standard errors of parameters in theta. Only returned if getSE=TRUE.

See Also

MSVARmdl

Examples

```
# ----- Bivariate VAR(1) process ----- #
set.seed(1234)
# Define DGP of VAR process
mdl_var <- list(n     = 1000,
                p     = 1,
                q     = 2,
                mu    = c(5,-2),
                sigma = rbind(c(5.0, 1.5),
                              c(1.5, 1.0)),
                phi   = rbind(c(0.50, 0.30),
                              c(0.20, 0.70)))
```

```
# Simulate process using simuVAR() function
y_simu <- simuVAR(mdl_var)

# Set options for model estimation
control <- list(const  = TRUE,
                getSE  = TRUE)

# Estimate model
y_var_mdl <- VARmdl(y_simu$y, p = 2, control = control)
summary(y_var_mdl)
```

VARXmdl *Vector X autoregressive model*

Description

This function estimates a vector autoregresive model with p lags. This can be used for the null hypothesis of a linear model against an alternative hypothesis of a Markov switching vector autoregressive model with k regimes.

Usage

```
VARXmdl(Y, p, Z, control = list())
```

Arguments

Y	a (T x q) matrix of observations.
p	integer determining the number of autoregressive lags.
Z	a (T x qz) matrix of exogenous regressors.
control	List with model options including: • const: Boolean determining whether to estimate model with constant if TRUE or not if FALSE. Default is TRUE. • getSE: Boolean determining whether to compute standard errors of parameters if TRUE or not if FALSE. Default is TRUE.

Value

List of class VARmdl (S3 object) with model attributes including:

- y: a (T-p x q) matrix of observations.
- X: a (T-p x p*q + const) matrix of lagged observations with a leading column of 1s if const=TRUE or not if const=FALSE.
- x: a (T-p x p*q) matrix of lagged observations.
- fitted: a (T-p x q) matrix of fitted values.
- resid: a (T-p x q) matrix of residuals.

- mu: a (1 x q) vector of estimated means of each process.
- beta: a ((1 + p + qz) x q) matrix of estimated coefficients.
- betaZ: a (qz x q) matrix of estimated exogenous regressor coefficients.
- intercept: estimate of intercepts.
- phi: a (q x p*q) matrix of estimated autoregressive coefficients.
- Fmat: Companion matrix containing autoregressive coefficients.
- stdev: a (q x 1) vector of estimated standard deviation of each process.
- sigma: a (q x q) estimated covariance matrix.
- theta: vector containing: mu, vech(sigma), and vec(t(phi)).
- theta_mu_ind: vector indicating location of mean with 1 and 0 otherwise.
- theta_sig_ind: vector indicating location of variance and covariances with 1 and 0 otherwise.
- theta_var_ind: vector indicating location of variances with 1 and 0 otherwise.
- theta_phi_ind: vector indicating location of autoregressive coefficients with 1 and 0 otherwise.
- stationary: Boolean indicating if process is stationary if TRUE or non-stationary if FALSE.
- n: number of observations after lag transformation (i.e., n = T-p).
- p: number of autoregressive lags.
- q: number of series.
- k: number of regimes. This is always 1 in VARmdl.
- Fmat: matrix from companion form. Used to determine is process is stationary.
- control: List with model options used.
- logLike: log-likelihood.
- AIC: Akaike information criterion.
- BIC: Bayesian (Schwarz) information criterion.
- Hess: Hessian matrix. Approximated using hessian and only returned if getSE=TRUE.
- info_mat: Information matrix. Computed as the inverse of -Hess. If matrix is not PD then nearest PD matrix is obtained using nearest_spd. Only returned if getSE=TRUE.
- nearPD_used: Boolean determining whether nearPD function was used on info_mat if TRUE or not if FALSE. Only returned if getSE=TRUE.
- theta_se: standard errors of parameters in theta. Only returned if getSE=TRUE.

See Also

MSVARmdl

Examples

```
# ----- Bivariate VAR(1) process ----- #
set.seed(1234)
# Define DGP of VAR process
mdl_var <- list(n     = 1000,
                p     = 1,
```

```
                q     = 2,
                mu    = c(5,-2),
                sigma = rbind(c(5.0, 1.5),
                              c(1.5, 1.0)),
                phi   = rbind(c(0.50, 0.30),
                              c(0.20, 0.70)))

# Simulate process using simuVAR() function
y_simu <- simuVAR(mdl_var)

# Set options for model estimation
control <- list(const  = TRUE,
                getSE  = TRUE)

# Estimate model
y_var_mdl <- VARmdl(y_simu$y, p = 2, control = control)
summary(y_var_mdl)
```

Index

∗ **datasets**
  chp10GNP, 8
  hamilton84GNP, 15
  USGNP, 56
  USRGDP, 57
∗ **package**
  MSTest-package, 3

ARmdl, 4, *8, 9*, *11*, *13*, *17*, *23*, *26*, *31*, *35*
ARXmdl, 6

chp10GNP, 8
CHPbootCV, 8
CHPTest, 9

DLMCTest, 10
DLMMCTest, 12

GA, *13*
ga, *26*
GenSA, *13*, *26*

hamilton84GNP, 15
hessian, *5*, *7*, *20*, *30*, *34*, *38*, *42*, *45*, *59*, *61*
HLRTest, 15
HMmdl, 18, *23*, *26*

initVals_MSARmdl, *18*, *28*, *32*, *36*, *40*

LMCLRTest, 22

MCpval, 24
MMCLRTest, 25
MSARmdl, *5*, *7*, *23*, *26*, 28
MSARXmdl, 32
MSTest *(*MSTest-package*)*, 3
MSTest-package, 3
MSVARmdl, *23*, *26*, 36, *59*, *61*
MSVARXmdl, 40

nearest_spd, *5*, *7*, *20*, *30*, *34*, *38*, *42*, *45*, *59*, *61*
Nmdl, *11*, *13*, *21*, *23*, *26*, 44

optim, *16*

pso, *13*
psoptim, *26*

simuAR, 46
simuARX, 47
simuHMM, 48
simuMSAR, 49
simuMSARX, 50
simuMSVAR, 51
simuMSVARX, 52
simuNorm, 53
simuVAR, 54
simuVARX, 55
slsqp, *16*

USGNP, 56
USRGDP, 57

VARmdl, *23*, *26*, *39*, *43*, 58
VARXmdl, 60