Package: mtarm (via r-universe)

November 23, 2026

Type Package

Title Bayesian Estimation of Multivariate Threshold Autoregressive Models

Version 0.1.10

Author Luis Hernando Vanegas [aut, cre], Sergio Alejandro Calderón [aut], Luz Marina Rondón [aut]

Maintainer Luis Hernando Vanegas <lhvanegasp@unal.edu.co>

Description Estimation, inference and forecasting using the Bayesian approach for multivariate threshold autoregressive (TAR) models in which the distribution used to describe the noise process belongs to the class of Gaussian variance mixtures.

Imports methods, stats, utils, graphics, Formula, grDevices, GIGrvg, coda, mvtnorm, future.apply, progressr, future

Suggests cli, knitr, rmarkdown, testthat (>= 3.0.0)

BugReports https://github.com/lhvanegasp/mtarm/issues

Encoding UTF-8

LazyData false

RoxygenNote 7.3.3

NeedsCompilation no

License GPL-2 | GPL-3

Config/testthat/edition 3

URL https://lhvanegasp.github.io/mtarm/

VignetteBuilder knitr

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

Date/Publication 2026-11-22 23:00:02 UTC

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

RemoteRef HEAD

RemoteSha 7d7fed1aadb977999fae6eb8b40cd8966a1da07e

Contents

ars . . . 3
as.mcmc.mtar . . . 3
coef.mtar . . . 5
DIC . . . 6
DIC.listmtar . . . 7
DIC.mtar . . . 7
effectiveSize_TAR . . . 9
fitted.mtar . . . 10
geweke_diagTAR . . . 11
geweke_plotTAR . . . 13
HPDinterval.mtar . . . 15
iceland.rf . . . 16
model.matrix.mtar . . . 17
mtar . . . 18
mtar_grid . . . 22
out_of_sample . . . 25
out_of_sample.listmtar . . . 26
out_of_sample.mtar . . . 28
plot.fittedmtar . . . 29
plot.mtarmcmc . . . 30
plot.predict_mtar . . . 31
plot.residuals_mtar . . . 32
predict.mtar . . . 32
print.essmtar . . . 35
print.gdmtar . . . 36
print.HPDmtar . . . 36
print.listmtar . . . 37
print.mtar . . . 37
print.residuals_mtar . . . 38
print.summary_mtar . . . 38
print.summtarmcmc . . . 39
priors . . . 40
residuals.mtar . . . 41
returns . . . 41
riverflows . . . 42
simtar . . . 43
summary.mtar . . . 46
summary.mtarmcmc . . . 47
US.returns . . . 48
vcov.mtar . . . 49
WAIC . . . 49
WAIC.listmtar . . . 50
WAIC.mtar . . . 51

Index **53**

ars *Auxiliary Function for Specifying Regimes and Lag Orders*

Description

Defines the regime structure of a multivariate Threshold Autoregressive (TAR) model by specifying the number of regimes and the regime-specific lag orders for the output, exogenous, and threshold series.

Usage

```
ars(nregim = 1, p = 1, q = 0, d = 0)
```

Arguments

nregim	A positive integer specifying the total number of regimes.
p	A vector of positive integers specifying the autoregressive order of the output series in each regime.
q	A vector of non-negative integers specifying the maximum lag order of the exogenous series in each regime.
d	A vector of non-negative integers specifying the maximum lag order of the threshold series in each regime.

Value

A list containing the number of regimes and the regime-specific lag-order specifications for the output, exogenous, and threshold series.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

as.mcmc.mtar *Coercion of* mtar *objects to* mtarmcmc *objects*

Description

Converts an object of class mtar into an object of class mtarmcmc containing the posterior MCMC samples in a format compatible with the coda package. For models with multiple regimes, regime-specific regression coefficients and scale parameters are stored separately for each regime. When applicable, the object also contains MCMC samples for the threshold parameters, delay parameter, skewness parameters, additional distributional parameters, and SSVS indicators.

Usage

```
## S3 method for class 'mtar'
as.mcmc(x, ...)
```

Arguments

x	An object of class mtar obtained from a call to mtar().
...	Additional arguments passed to or from other methods.

Value

An object of class mtarmcmc containing the posterior simulation draws generated by mtar(), represented using mcmc objects from the **coda** package.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

See Also

as.mcmc

Examples

```
###### Example 1: Returns of the closing prices of three financial indexes
data(returns)
fit1 <- mtar(~ COLCAP + BOVESPA | SP500, data=returns, row.names=Date,
             subset={Date<="2015-12-07"}, dist="Student-t",
             ars=ars(nregim=3,p=c(1,1,2)), n.burnin=1000, n.sim=2000,
             n.thin=2, ssvs=TRUE)
fit1.mcmc <- coda::as.mcmc(fit1)
summary(fit1.mcmc)
#plot(fit1.mcmc)

###### Example 2: Rainfall and two river flows in Colombia
data(riverflows)
fit2 <- mtar(~ Bedon + LaPlata | Rainfall, data=riverflows, row.names=Date,
             subset={Date<="2009-02-13"}, dist="Laplace",
             ars=ars(nregim=3,p=5), n.burnin=1000, n.sim=2000, n.thin=2)
fit2.mcmc <- coda::as.mcmc(fit2)
summary(fit2.mcmc)
#plot(fit2.mcmc)

###### Example 3: Temperature, precipitation, and two river flows in Iceland
data(iceland.rf)
fit3 <- mtar(~ Jokulsa + Vatnsdalsa | Temperature | Precipitation,
             data=iceland.rf, subset={Date<="1974-11-06"}, row.names=Date,
```

```
            ars=ars(nregim=2,p=15,q=4,d=2), n.burnin=1000, n.sim=2000,
            n.thin=2, dist="Slash")
fit3.mcmc <- coda::as.mcmc(fit3)
summary(fit3.mcmc)
#plot(fit3.mcmc)

###### Example 4: U.S. stock returns
data(US.returns)
fit4 <- mtar(~ CCR | dVIX, data=US.returns, subset={Date<="2025-11-28"},
            row.names=Date, ars=ars(nregim=2,p=3,d=3), n.burnin=1000,
            n.sim=2000, n.thin=2, dist="Student-t")
fit4.mcmc <- coda::as.mcmc(fit4)
summary(fit4.mcmc)
#plot(fit4.mcmc)
```

`coef.mtar` *Extract Coefficient Estimates from* `mtar` *Objects*

Description

Extracts posterior summaries of the location, scale, threshold, skewness, and additional distributional parameters from a fitted multivariate Threshold Autoregressive (TAR) model. The summaries are obtained by applying a user-specified function to the corresponding MCMC samples. When Stochastic Search Variable Selection (SSVS) is used, posterior inclusion probabilities are summarized for the output, exogenous, and threshold lags. Coefficients with posterior inclusion probability not exceeding 0.5 are set to zero in the returned location parameters.

Usage

```
## S3 method for class 'mtar'
coef(object, ..., FUN = mean)
```

Arguments

`object`	An object of class `mtar` obtained from a call to `mtar()`.
`...`	Additional arguments passed to `FUN`.
`FUN`	A function applied to the MCMC samples of each parameter to obtain a posterior summary. By default, `FUN = mean`.

Value

A list containing posterior summaries of the parameters of the fitted model. Regime-specific location and scale parameters are stored in separate elements. For models with more than one regime, summaries of the delay and threshold parameters are also returned. For skewed distributions, the skewness parameters are included, and for distributions with additional parameters, the corresponding distribution-specific parameters are returned. When SSVS is used, posterior inclusion probabilities for the selected lag groups are also included.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

`DIC` *Deviance Information Criterion (DIC)*

Description

Computes the Deviance Information Criterion (DIC), an adjusted within-sample measure of predictive accuracy, for models estimated using Bayesian methods.

Usage

```
DIC(...)
```

Arguments

`...`	one or more fitted model objects of the same class.

Value

A numeric matrix containing the DIC values corresponding to each fitted object supplied in `...`.

References

Spiegelhalter D.J., Best N.G., Carlin B.P. and Van Der Linde A. (2002) Bayesian Measures of Model Complexity and Fit. Journal of the Royal Statistical Society Series B (Statistical Methodology), 64(4), 583–639.

Spiegelhalter D.J., Best N.G., Carlin B.P. and Van der Linde A. (2014). The deviance information criterion: 12 years on. Journal of the Royal Statistical Society Series B (Statistical Methodology), 76(3), 485–493.

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

DIC.listmtar *Deviance Information Criterion (DIC) for a list of* mtar *objects*

Description

Computes the Deviance Information Criterion (DIC) for each mtar object contained in a list.

Usage

```
## S3 method for class 'listmtar'
DIC(x, ...)
```

Arguments

x	A list containing fitted objects of class mtar.
...	Further arguments passed to or from other methods.

Value

A numeric matrix containing one DIC value for each mtar object in x. The row names are taken from the names of x.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

DIC.mtar *Deviance Information Criterion (DIC) for* mtar *objects*

Description

Computes the Deviance Information Criterion (DIC) for one or more fitted objects of class mtar. The criterion is computed from the posterior samples generated during Bayesian estimation and accounts for both model fit and effective model complexity.

Usage

```
## S3 method for class 'mtar'
DIC(...)
```

Arguments

...	One or more fitted objects of class mtar.

Details

For models with Skew-normal and Skew-Student-t error distributions, the computation of the DIC can be computationally demanding. This is because the likelihood evaluation requires the calculation of the probability of the latent variable being in the positive orthant. For the Skew-normal distribution, this probability is evaluated using a multivariate normal distribution, whereas for the Skew-Student-t distribution it is evaluated using a multivariate Student-t distribution. Consequently, the computation of the DIC for these distributions may require considerably more computational time than for the other error distributions implemented in `mtar`.

Value

A numeric matrix containing one DIC value for each `mtar` object supplied through ....

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

See Also

WAIC

Examples

```
###### Example 1: Returns of the closing prices of three financial indexes
data(returns)
fit1 <- mtar_grid(~ COLCAP + BOVESPA | SP500, data=returns, row.names=Date,
                  subset={Date<="2015-12-07"}, dist=c("Gaussian","Student-t",
                  "Slash","Laplace"), nregim.min=2, nregim.max=3, p.min=2,
                  p.max=2, n.burnin=1000, n.sim=2000, n.thin=2,
                  plan_strategy="multisession")
DIC(fit1)

###### Example 2: Rainfall and two river flows in Colombia
data(riverflows)
fit2 <- mtar_grid(~ Bedon + LaPlata | Rainfall, data=riverflows,
                  row.names=Date, subset={Date<="2009-02-13"},dist="Laplace",
                  nregim.min=2, nregim.max=3, p.min=1, p.max=3,n.burnin=1000,
                  n.sim=2000, n.thin=2, plan_strategy="multisession")
DIC(fit2)

###### Example 3: Temperature, precipitation, and two river flows in Iceland
data(iceland.rf)
fit3 <- mtar_grid(~ Jokulsa + Vatnsdalsa | Temperature | Precipitation,
                  data=iceland.rf,subset={Date<="1974-11-06"},row.names=Date,
                  dist=c("Slash","Student-t"), nregim.min=1, nregim.max=2,
                  p.min=15, p.max=15, q.min=4, q.max=4, d.min=2, d.max=2,
                  n.burnin=1000, n.sim=2000, n.thin=2,
                  plan_strategy="multisession")
DIC(fit3)
```

```
###### Example 4: U.S. stock returns
data(US.returns)
fit4 <- mtar_grid(~ CCR | dVIX, data=US.returns, subset={Date<="2025-11-28"},
                  row.names=Date, dist=c("Laplace","Student-t","Slash"),
                  nregim.min=2, nregim.max=2, p.min=3, p.max=3, d.min=3,
                  d.max=3, n.burnin=1000, n.sim=2000, n.thin=2,
                  plan_strategy="multisession")
DIC(fit4)
```

`effectiveSize_TAR` *Effective Sample Size for* `mtar` *Objects*

Description

Computes the effective sample size of the Markov chain Monte Carlo (MCMC) samples obtained from a fitted multivariate Threshold Autoregressive (TAR) model. The effective sample size accounts for the autocorrelation present in the posterior MCMC chains. The function is a wrapper around `effectiveSize`.

Usage

```
effectiveSize_TAR(x)
```

Arguments

`x`	An object of class `mtar` returned by `mtar()`.

Value

An object of class `essmtar` containing the effective sample sizes for the parameters of the fitted `mtar` model. The results are provided separately for each regime and, when applicable, for the threshold, skewness, and additional distributional parameters.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

See Also

`effectiveSize`

Examples

```
###### Example 1: Returns of the closing prices of three financial indexes
data(returns)
fit1 <- mtar(~ COLCAP + BOVESPA | SP500, data=returns, row.names=Date,
             subset={Date<="2015-12-07"}, dist="Student-t",
             ars=ars(nregim=3,p=c(1,1,2)), n.burnin=1000, n.sim=2000,
             n.thin=2)
effectiveSize_TAR(fit1)

###### Example 2: Rainfall and two river flows in Colombia
data(riverflows)
fit2 <- mtar(~ Bedon + LaPlata | Rainfall, data=riverflows, row.names=Date,
             subset={Date<="2009-02-13"}, dist="Laplace",
             ars=ars(nregim=3,p=5), n.burnin=1000, n.sim=2000, n.thin=2)
effectiveSize_TAR(fit2)

###### Example 3: Temperature, precipitation, and two river flows in Iceland
data(iceland.rf)
fit3 <- mtar(~ Jokulsa + Vatnsdalsa | Temperature | Precipitation,
             data=iceland.rf, subset={Date<="1974-11-06"}, row.names=Date,
             ars=ars(nregim=2,p=15,q=4,d=2), n.burnin=1000, n.sim=2000,
             n.thin=2, dist="Slash")
effectiveSize_TAR(fit3)

###### Example 4: U.S. stock returns
data(US.returns)
fit4 <- mtar(~ CCR | dVIX, data=US.returns, subset={Date<="2025-11-28"},
             row.names=Date, ars=ars(nregim=2,p=3,d=3), n.burnin=1000,
             n.sim=2000, n.thin=2, dist="Student-t")
effectiveSize_TAR(fit4)
```

`fitted.mtar` *Fitted values from a Bayesian multivariate threshold autoregressive model*

Description

Computes fitted values from a Bayesian multivariate threshold autoregressive (TAR) model. The fitted values are obtained by evaluating the theoretical conditional mean of the model at the posterior mean estimates of the model parameters.

Usage

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

Arguments

`object`	An object of class `"mtar"` returned by `mtar`.
`...`	Additional arguments. Currently not used.

Details

In an TAR model, the conditional location parameter of the response vector depends on the regime associated with each observation. Therefore, when the model contains more than one regime, each regime has its own regression coefficients, and the conditional location parameter can differ across regimes. For models with more than one regime, the regime associated with each observation is determined using the posterior mean estimates of the delay parameter and the threshold parameters. The corresponding regime-specific regression coefficients are then used to compute the conditional location parameter. For symmetric error distributions, such as the Gaussian, Student-t, Hyperbolic, Laplace, Slash, and contaminated normal distributions, the conditional mean coincides with the conditional location parameter. Consequently, the fitted values are given by the regime-specific conditional location evaluated at the posterior mean estimates of the regression coefficients. For the Skew-normal distribution, the conditional mean differs from the conditional location parameter because of the skewness of the distribution. The fitted values therefore include the contribution of the posterior mean estimate of the skewness parameter. For the Skew-Student-t distribution, the conditional mean also depends on the degrees of freedom parameter and the skewness parameter. The fitted values are obtained by evaluating the theoretical conditional mean at their posterior mean estimates. The conditional mean exists only when the degrees of freedom parameter is greater than one. When stochastic search variable selection (SSVS) is used, only coefficients with posterior inclusion probability greater than 0.5 are included when computing the fitted values. The resulting fitted values are returned in the original temporal order of the observations.

Value

An object of class `"fittedmtar"` containing:

observed The observed response matrix.

fitted The matrix of fitted values, with one row for each observation and one column for each response variable.

rownames A logical value indicating whether row names were originally present in the model call.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

`geweke_diagTAR` *Geweke's Convergence Diagnostic for* `mtar` *Objects*

Description

Computes Geweke's convergence diagnostic for the Markov chain Monte Carlo (MCMC) samples obtained from a fitted multivariate Threshold Autoregressive (TAR) model. The function is a wrapper around `geweke.diag` that applies the diagnostic to the posterior samples stored in an object of class `mtar`.

Usage

```
geweke_diagTAR(x, frac1 = 0.1, frac2 = 0.5)
```

Arguments

`x`	An object of class `mtar` returned by `mtar()`.
`frac1`	A numeric value in $(0, 1)$ specifying the fraction of the beginning of each MCMC chain used in the diagnostic.
`frac2`	A numeric value in $(0, 1)$ specifying the fraction of the end of each MCMC chain used in the diagnostic.

Value

An object of class `gdmtar` containing the Geweke z-scores for the parameters of the fitted `mtar` model. The results are provided separately for each regime and, when applicable, for the threshold, skewness, and additional distributional parameters.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

See Also

`geweke.diag`

Examples

```
###### Example 1: Returns of the closing prices of three financial indexes
data(returns)
fit1 <- mtar(~ COLCAP + BOVESPA | SP500, data=returns, row.names=Date,
             subset={Date<="2015-12-07"}, dist="Student-t",
             ars=ars(nregim=3,p=c(1,1,2)), n.burnin=1000, n.sim=2000,
             n.thin=2)
geweke_diagTAR(fit1)

###### Example 2: Rainfall and two river flows in Colombia
data(riverflows)
fit2 <- mtar(~ Bedon + LaPlata | Rainfall, data=riverflows, row.names=Date,
             subset={Date<="2009-02-13"}, dist="Laplace",
             ars=ars(nregim=3,p=5), n.burnin=1000, n.sim=2000, n.thin=2)
geweke_diagTAR(fit2)
```

```
###### Example 3: Temperature, precipitation, and two river flows in Iceland
data(iceland.rf)
fit3 <- mtar(~ Jokulsa + Vatnsdalsa | Temperature | Precipitation,
             data=iceland.rf, subset={Date<="1974-11-06"}, row.names=Date,
             ars=ars(nregim=2,p=15,q=4,d=2), n.burnin=1000, n.sim=2000,
             n.thin=2, dist="Slash")
geweke_diagTAR(fit3)

###### Example 4: U.S. stock returns
data(US.returns)
fit4 <- mtar(~ CCR | dVIX, data=US.returns, subset={Date<="2025-11-28"},
             row.names=Date, ars=ars(nregim=2,p=3,d=3), n.burnin=1000,
             n.sim=2000, n.thin=2, dist="Student-t")
geweke_diagTAR(fit4)
```

`geweke_plotTAR` *Geweke-Brooks Plot for* `mtar` *Objects*

Description

Produces Geweke-Brooks convergence diagnostic plots for the MCMC samples obtained from a fitted multivariate Threshold Autoregressive (TAR) model. The function is a wrapper around `geweke.plot` and produces separate plots for the parameters associated with each regime and, when applicable, for the threshold, skewness, and additional distributional parameters.

Usage

```
geweke_plotTAR(
  x,
  frac1 = 0.1,
  frac2 = 0.5,
  nbins = 20,
  pvalue = 0.05,
  auto.layout = TRUE,
  ask = dev.interactive(),
  ...
)
```

Arguments

`x`	An object of class `mtar` returned by `mtar()`.
`frac1`	A numeric value in $(0, 1)$ specifying the fraction of the beginning of each MCMC chain used in the diagnostic.
`frac2`	A numeric value in $(0, 1)$ specifying the fraction of the end of each MCMC chain used in the diagnostic.

nbins	An integer specifying the number of segments used to divide the MCMC chain.
pvalue	A numeric value specifying the significance level used to construct the confidence limits for the null hypothesis.
auto.layout	A logical indicating whether the plotting layout should be determined automatically. By default, auto.layout is TRUE.
ask	A logical indicating whether the user should be prompted before displaying each page of plots. By default, ask = dev.interactive().
...	Additional graphical parameters passed to geweke.plot.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

See Also

geweke.plot

Examples

```
###### Example 1: Returns of the closing prices of three financial indexes
data(returns)
fit1 <- mtar(~ COLCAP + BOVESPA | SP500, data=returns, row.names=Date,
             subset={Date<="2015-12-07"}, dist="Student-t",
             ars=ars(nregim=3,p=c(1,1,2)), n.burnin=1000, n.sim=2000,
             n.thin=2)
geweke_plotTAR(fit1)

###### Example 2: Rainfall and two river flows in Colombia
data(riverflows)
fit2 <- mtar(~ Bedon + LaPlata | Rainfall, data=riverflows, row.names=Date,
             subset={Date<="2009-02-13"}, dist="Laplace",
             ars=ars(nregim=3,p=5), n.burnin=1000, n.sim=2000, n.thin=2)
geweke_plotTAR(fit2)

###### Example 3: Temperature, precipitation, and two river flows in Iceland
data(iceland.rf)
fit3 <- mtar(~ Jokulsa + Vatnsdalsa | Temperature | Precipitation,
             data=iceland.rf, subset={Date<="1974-11-06"}, row.names=Date,
             ars=ars(nregim=2,p=15,q=4,d=2), n.burnin=1000, n.sim=2000,
             n.thin=2, dist="Slash")
geweke_plotTAR(fit3)

###### Example 4: U.S. stock returns
data(US.returns)
fit4 <- mtar(~ CCR | dVIX, data=US.returns, subset={Date<="2025-11-28"},
             row.names=Date, ars=ars(nregim=2,p=3,d=3), n.burnin=1000,
             n.sim=2000, n.thin=2, dist="Student-t")
geweke_plotTAR(fit4)
```

`HPDinterval.mtar` *Highest Posterior Density Intervals for* `mtar` *Objects*

Description

Computes Highest Posterior Density (HPD) intervals for the posterior distributions of the parameters of an object of class `mtar`. The intervals are computed from the posterior MCMC samples using the corresponding method from the **coda** package.

Usage

```
## S3 method for class 'mtar'
HPDinterval(obj, prob = 0.95, ...)
```

Arguments

`obj`	An object of class `mtar` obtained from a call to `mtar()`.
`prob`	A numeric scalar in the interval $(0, 1)$ specifying the target posterior probability content of the intervals. By default, `prob = 0.95`.
`...`	Additional arguments passed to `HPDinterval`.

Value

An object of class `HPDmtar` containing the HPD intervals for the parameters of the fitted model.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

See Also

`HPDinterval`

Examples

```
###### Example 1: Returns of the closing prices of three financial indexes
data(returns)
fit1 <- mtar(~ COLCAP + BOVESPA | SP500, data=returns, row.names=Date,
             subset={Date<="2015-12-07"}, dist="Student-t",
             ars=ars(nregim=3,p=c(1,1,2)), n.burnin=1000, n.sim=2000,
             n.thin=2, ssvs=TRUE)
coda::HPDinterval(fit1)

###### Example 2: Rainfall and two river flows in Colombia
data(riverflows)
fit2 <- mtar(~ Bedon + LaPlata | Rainfall, data=riverflows, row.names=Date,
             subset={Date<="2009-02-13"}, dist="Laplace",
```

```
            ars=ars(nregim=3,p=5), n.burnin=1000, n.sim=2000, n.thin=2)
coda::HPDinterval(fit2)

###### Example 3: Temperature, precipitation, and two river flows in Iceland
data(iceland.rf)
fit3 <- mtar(~ Jokulsa + Vatnsdalsa | Temperature | Precipitation,
            data=iceland.rf, subset={Date<="1974-11-06"}, row.names=Date,
            ars=ars(nregim=2,p=15,q=4,d=2), n.burnin=1000, n.sim=2000,
            n.thin=2, dist="Slash")
coda::HPDinterval(fit3)

###### Example 4: U.S. stock returns
data(US.returns)
fit4 <- mtar(~ CCR | dVIX, data=US.returns, subset={Date<="2025-11-28"},
            row.names=Date, ars=ars(nregim=2,p=3,d=3), n.burnin=1000,
            n.sim=2000, n.thin=2, dist="Student-t")
coda::HPDinterval(fit4)
```

`iceland.rf` *Temperature, Precipitation, and Two River Flows in Iceland*

Description

This dataset contains two daily river-flow time series, measured in cubic meters per second, for rivers in Iceland. It also contains daily measurements of precipitation, in millimeters, and temperature, in degrees Celsius, recorded at the Hveravellir meteorological station. The precipitation values correspond to accumulated precipitation up to 9:00 A.M. relative to the same time on the previous day.

Usage

```
data(iceland.rf)
```

Format

A data frame with 1,096 rows and 5 variables:

Vatnsdalsa A numeric vector containing the daily flow of the Vatnsdalsá River, in cubic meters per second.

Jokulsa A numeric vector containing the daily flow of the Jökulsá Eystri River, in cubic meters per second.

Precipitation A numeric vector containing daily precipitation amounts, in millimeters.

Temperature A numeric vector containing daily temperature measurements, in degrees Celsius.

Date A vector containing the calendar date of each observation.

References

Tong, H. (1990). *Non-linear Time Series: A Dynamical System Approach*. Oxford University Press, Oxford, UK. Tsay, R. S. (1998). Testing and modeling multivariate threshold models. *Journal of the American Statistical Association*, 93, 1188–1202.

Examples

```
data(iceland.rf)
dev.new()
plot(ts(as.matrix(iceland.rf[, -5])), main = "Iceland")
```

`model.matrix.mtar` *Extract Model Matrices from* `mtar` *Objects*

Description

Extracts the design matrix associated with each regime of a fitted multivariate Threshold Autoregressive (TAR) model.

Usage

```
## S3 method for class 'mtar'
model.matrix(object, ...)
```

Arguments

`object`	An object of class `mtar` obtained from a call to `mtar()`.
`...`	Further arguments passed to or from other methods.

Value

A list containing the model matrix associated with each regime of the fitted `mtar` model.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

`mtar` *Bayesian estimation of a multivariate Threshold Autoregressive (TAR) model.*

Description

Fits a multivariate Threshold Autoregressive (TAR) model using Bayesian inference and a Gibbs sampling algorithm. The model allows the conditional dynamics of the multivariate output series to vary across regimes determined by a threshold series and a delay parameter. The model includes, as special cases, Self-Exciting Threshold Autoregressive (SETAR) models and Vector Autoregressive (VAR) models. It can incorporate autoregressive terms of the output series, lagged exogenous variables, lagged values of the threshold series, deterministic terms, and seasonal effects. The innovation process can follow Gaussian, Student-t, Slash, Hyperbolic, Laplace, contaminated normal, skew-normal, or skew-Student-t distributions.

Usage

```
mtar(
  formula,
  data,
  subset,
  Intercept = TRUE,
  trend = c("none", "linear", "quadratic"),
  nseason = NULL,
  ars = ars(),
  row.names,
  dist = c("Gaussian", "Student-t", "Hyperbolic", "Laplace", "Slash",
    "Contaminated normal", "Skew-Student-t", "Skew-normal"),
  prior = list(),
  n.sim = 500,
  n.burnin = 100,
  n.thin = 1,
  ssvs = FALSE,
  setar = NULL,
  progress = TRUE,
  ...
)
```

Arguments

`formula`	A three-part expression of class `Formula` specifying the multivariate TAR model to be fitted. The first part defines the multivariate output series, the second part defines the threshold series, and the third part defines the multivariate exogenous series. The three parts are separated by the `|` operator.
`data`	A data frame containing the variables used in the model. If a variable is not found in `data`, it is searched for in the environment associated with `formula`.

`subset` An optional vector specifying a subset of observations to be used in the model fitting. If supplied, only the corresponding observations are included in the analysis.

`Intercept` An optional logical indicating whether an intercept term is included in each regime. By default, `Intercept` is set to `TRUE`.

`trend` An optional character string specifying the deterministic time trend included in each regime. Available options are `"none"`, `"linear"`, and `"quadratic"`, corresponding to no trend, a linear trend, and a quadratic trend, respectively. By default, `trend` is set to `"none"`.

`nseason` An optional integer specifying the number of seasonal periods. It must be greater than or equal to 2. When `nseason` is provided, `nseason - 1` seasonal dummy variables are included among the regressors in each regime. By default, `nseason` is set to `NULL`, indicating that no seasonal effects are included in the model.

`ars` A list specifying the autoregressive structure of the model. It must contain four components: `nregim`, the number of regimes; `p`, the autoregressive orders for the output series; `q`, the maximum lag orders for the exogenous series; and `d`, the maximum lag orders for the threshold series. The components `p`, `q`, and `d` are non-negative integer vectors with one element for each regime. The object can be constructed and validated using the `ars()` function.

`row.names` An optional variable in `data` identifying the time points associated with the observations. It is used to label the rows of the output and to preserve the temporal ordering of the observations.

`dist` A character string specifying the multivariate distribution of the innovation process. Available options are `"Gaussian"`, `"Student-t"`, `"Slash"`, `"Hyperbolic"`, `"Contaminated normal"`, `"Laplace"`, `"Skew-normal"`, and `"Skew-Student-t"`. By default, `dist` is set to `"Gaussian"`.

`prior` An optional list specifying the hyperparameter values defining the prior distributions of the model parameters. The list can be specified and validated using the `priors()` function. By default, `prior` is set to an empty list, in which case the default hyperparameter values implemented by `priors()` are used.

`n.sim` An optional positive integer specifying the number of posterior samples retained after burn-in and thinning. By default, `n.sim` is set to `500`.

`n.burnin` An optional non-negative integer specifying the number of initial MCMC iterations discarded as burn-in. By default, `n.burnin` is set to `100`.

`n.thin` An optional positive integer specifying the thinning interval used to retain MCMC samples. A value of `1` retains every iteration, whereas larger values retain every `n.thin`-th iteration. By default, `n.thin` is set to `1`.

`ssvs` An optional logical indicating whether Stochastic Search Variable Selection (SSVS) is used to identify relevant lagged variables. When `ssvs = TRUE`, the procedure performs variable selection among the lagged output, exogenous, and threshold variables. By default, `ssvs` is set to `FALSE`.

`setar` An optional positive integer specifying the component of the multivariate output series to be used as the threshold variable. When `setar` is `NULL`, the threshold series is specified through the second part of `formula`. When `setar` is supplied, the model uses the corresponding component of the output series as the threshold variable, yielding a SETAR-type specification. By default, `setar` is set to `NULL`.

progress	An optional logical indicating whether a progress bar is displayed during the MCMC simulation. By default, progress is set to TRUE.
...	Further arguments passed to or from other methods.

Value

An object of class mtar containing the fitted model and the results of the MCMC simulation. The object is a list with components including:

chains	A list containing the MCMC samples of the model parameters for each regime. The stored chains inclu
n.sim	Number of MCMC samples retained after burn-in and thinning.
n.burnin	Number of initial MCMC iterations discarded as burn-in.
n.thin	Thinning interval used to retain MCMC samples.
ars	A list containing the autoregressive structure of the fitted model: nregim, p, q, and d.
dist	Character string identifying the multivariate distribution used for the innovation process.
threshold.series	Vector containing the values of the threshold series used to define the regimes.
output.series	Matrix containing the values of the multivariate output series used in the model.
exogenous.series	Matrix containing the values of the multivariate exogenous series used in the model.
Intercept	Logical value indicating whether an intercept term was included in each regime.
trend	Character string indicating the deterministic time trend included in the model, if any.
nseason	Number of seasonal periods included in the model, if applicable.
formula	The Formula object used to fit the model.
call	The original function call.

References

Nieto, F.H. (2005) Modeling Bivariate Threshold Autoregressive Processes in the Presence of Missing Data. Communications in Statistics - Theory and Methods, 34, 905-930.

Romero, L.V. and Calderon, S.A. (2021) Bayesian estimation of a multivariate TAR model when the noise process follows a Student-t distribution. Communications in Statistics - Theory and Methods, 50, 2508-2530.

Calderon, S.A. and Nieto, F.H. (2017) Bayesian analysis of multivariate threshold autoregressive models with missing data. Communications in Statistics - Theory and Methods, 46, 296-318.

Vanegas, L.H. and Calderón, S.A. and Rondón, L.M. (2025) Bayesian estimation of a multivariate tar model when the noise process distribution belongs to the class of gaussian variance mixtures. International Journal of Forecasting.

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

See Also

DIC, WAIC

Examples

```
###### Example 1: Returns of the closing prices of three financial indexes
data(returns)
fit1 <- mtar(~ COLCAP + BOVESPA | SP500, data=returns, row.names=Date,
             subset={Date<="2015-12-07"}, dist="Student-t",
             ars=ars(nregim=3,p=c(1,1,2)), n.burnin=1000, n.sim=2000,
             n.thin=2)
summary(fit1)
DIC(fit1)
WAIC(fit1)

###### Example 2: Rainfall and two river flows in Colombia
data(riverflows)
fit2 <- mtar(~ Bedon + LaPlata | Rainfall, data=riverflows, row.names=Date,
             subset={Date<="2009-02-13"}, dist="Laplace",
             ars=ars(nregim=3,p=5), n.burnin=1000, n.sim=2000, n.thin=2)
summary(fit2)
DIC(fit2)
WAIC(fit2)

###### Example 3: Temperature, precipitation, and two river flows in Iceland
data(iceland.rf)
fit3 <- mtar(~ Jokulsa + Vatnsdalsa | Temperature | Precipitation,
             data=iceland.rf, subset={Date<="1974-11-06"}, row.names=Date,
             ars=ars(nregim=2,p=15,q=4,d=2), n.burnin=1000, n.sim=2000,
             n.thin=2, dist="Slash")
summary(fit3)
DIC(fit3)
WAIC(fit3)

###### Example 4: U.S. stock returns
data(US.returns)
fit4 <- mtar(~ CCR | dVIX, data=US.returns, subset={Date<="2025-11-28"},
             row.names=Date, ars=ars(nregim=2,p=3,d=3), n.burnin=1000,
             n.sim=2000, n.thin=2, dist="Student-t")
summary(fit4)
DIC(fit4)
WAIC(fit4)
```

mtar_grid *Bayesian Estimation of Multivariate TAR Models*

Description

This function is a wrapper that applies `mtar()` over a grid of model specifications defined by all combinations of the noise distribution (`dist`), the number of regimes (from `nregim.min` to `nregim.max`), the autoregressive order within each regime (from `p.min` to `p.max`), the maximum lag of the exogenous series within each regime (from `q.min` to `q.max`), and the maximum lag of the threshold series within each regime (from `d.min` to `d.max`). In all calls to `mtar()`, the same set of time points is used for model fitting. This is achieved by appropriately adjusting the `subset` argument of `mtar()` for each model specification, thereby ensuring comparability across models.

Usage

```
mtar_grid(
  formula,
  data,
  subset,
  Intercept = TRUE,
  trend = c("none", "linear", "quadratic"),
  nseason = NULL,
  nregim.min = 1,
  nregim.max = NULL,
  p.min = 1,
  p.max = NULL,
  q.min = 0,
  q.max = 0,
  d.min = 0,
  d.max = 0,
  row.names,
  dist = "Gaussian",
  prior = list(),
  n.sim = 500,
  n.burnin = 100,
  n.thin = 1,
  ssvs = FALSE,
  setar = NULL,
  plan_strategy = c("multisession", "sequential"),
  progress = TRUE
)
```

Arguments

formula	A three-part expression of class Formula describing the TAR model to be fitted. The first part specifies the variables in the multivariate output series, the second part defines the threshold series, and the third part specifies the variables in the multivariate exogenous series.
data	A data frame containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which mtar_grid() is called.
subset	An optional vector specifying a subset of observations to be used in the fitting process.
Intercept	An optional logical indicating whether an intercept should be included within each regime.
trend	An optional character string specifying the degree of deterministic time trend to be included in each regime. Available options are "linear", "quadratic", and "none". By default, trend is set to "none".
nseason	An optional integer, greater than or equal to 2, specifying the number of seasonal periods. When provided, nseason - 1 seasonal dummy variables are added to the regressors within each regime. By default, nseason is set to NULL, thereby indicating that the TAR model has no seasonal effects.
nregim.min	An optional integer specifying the minimum number of regimes. By default, nregim.min is set to 1.
nregim.max	An integer specifying the maximum number of regimes.
p.min	An optional integer specifying the minimum autoregressive order within each regime. By default, p.min is set to 1.
p.max	An integer specifying the maximum autoregressive order within each regime.
q.min	An optional integer specifying the minimum value of the maximum lag of the exogenous series within each regime. By default, q.min is set to 0.
q.max	An optional integer specifying the maximum value of the maximum lag of the exogenous series within each regime. By default, q.max is set to 0.
d.min	An optional integer specifying the minimum value of the maximum lag of the threshold series within each regime. By default, d.min is set to 0.
d.max	An optional integer specifying the maximum value of the maximum lag of the threshold series within each regime. By default, d.max is set to 0.
row.names	An optional variable in data labelling the time points corresponding to each row of the data set.
dist	A character vector specifying the multivariate distributions used to model the noise process. Available options are "Gaussian", "Student-t", "Slash", "Hyperbolic", "Laplace", "Contaminated normal", "Skew-normal", and "Skew-Student-t". By default, dist is set to "Gaussian".
prior	An optional list specifying the hyperparameter values that define the prior distribution. This list can be validated using the priors() function. By default, prior is set to an empty list, thereby indicating that the hyperparameter values should be set so that a non-informative prior distribution is obtained.

n.sim	An optional positive integer specifying the number of simulation iterations after the burn-in period. By default, n.sim is set to 500.
n.burnin	An optional positive integer specifying the number of burn-in iterations. By default, n.burnin is set to 100.
n.thin	An optional positive integer specifying the thinning interval. By default, n.thin is set to 1.
ssvs	An optional logical indicating whether the Stochastic Search Variable Selection (SSVS) procedure should be applied to identify relevant lags of the output, exogenous, and threshold series. By default, ssvs is set to FALSE.
setar	An optional positive integer indicating the component of the output series used as the threshold variable. By default, setar is set to NULL, indicating that the fitted model is not a SETAR model.
plan_strategy	An optional character string specifying the execution strategy for parallel computation. Available options are "sequential" and "multisession". By default, plan_strategy is set to "multisession".
progress	An optional logical indicating whether a progress bar should be displayed during execution. By default, progress is set to TRUE.

Value

A list whose elements are objects of class mtar, each corresponding to a distinct model specification considered in the grid.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

See Also

mtar

Examples

```
###### Example 1: Returns of the closing prices of three financial indexes
data(returns)
fit1 <- mtar_grid(~ COLCAP + BOVESPA | SP500, data=returns, row.names=Date,
                  subset={Date<="2015-12-07"}, dist=c("Gaussian","Student-t",
                  "Slash","Laplace"), nregim.min=2, nregim.max=3, p.min=2,
                  p.max=2, n.burnin=1000, n.sim=2000, n.thin=2,
                  plan_strategy="multisession")
summary(fit1)

###### Example 2: Rainfall and two river flows in Colombia
data(riverflows)
fit2 <- mtar_grid(~ Bedon + LaPlata | Rainfall, data=riverflows,
                  row.names=Date, subset={Date<="2009-02-13"},dist="Laplace",
                  nregim.min=2, nregim.max=3, p.min=1, p.max=3,n.burnin=1000,
                  n.sim=2000, n.thin=2, plan_strategy="multisession")
```

```
fit2

###### Example 3: Temperature, precipitation, and two river flows in Iceland
data(iceland.rf)
fit3 <- mtar_grid(~ Jokulsa + Vatnsdalsa | Temperature | Precipitation,
                  data=iceland.rf,subset={Date<="1974-11-06"},row.names=Date,
                  dist=c("Slash","Student-t"), nregim.min=1, nregim.max=2,
                  p.min=15, p.max=15, q.min=4, q.max=4, d.min=2, d.max=2,
                  n.burnin=1000, n.sim=2000, n.thin=2,
                  plan_strategy="multisession")
fit3

###### Example 4: U.S. stock returns
data(US.returns)
fit4 <- mtar_grid(~ CCR | dVIX, data=US.returns, subset={Date<="2025-11-28"},
                  row.names=Date, dist=c("Laplace","Student-t","Slash"),
                  nregim.min=2, nregim.max=2, p.min=3, p.max=3, d.min=3,
                  d.max=3, n.burnin=1000, n.sim=2000, n.thin=2,
                  plan_strategy="multisession")
fit4
```

`out_of_sample` *Out-of-sample predictive accuracy measures*

Description

Computes out-of-sample predictive accuracy measures for one or more fitted models of the same class using their predictive distributions and a set of future observations.

Usage

```
out_of_sample(..., newdata, n.ahead)
```

Arguments

`...`	One or more fitted model objects of the same class.
`newdata`	A data frame containing the future observations of the output series used to evaluate predictive performance.
`n.ahead`	A positive integer specifying the forecast horizon, that is, the number of steps ahead for which predictive accuracy is evaluated.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

`out_of_sample.listmtar` *Out-of-sample Predictive Accuracy for* `listmtar` *Objects*

Description

Computes out-of-sample predictive accuracy measures for the fitted models contained in an object of class `listmtar`. The measures are obtained by comparing the predictive distributions with the observed future values of the output series.

Usage

```
## S3 method for class 'listmtar'
out_of_sample(
  x,
  newdata,
  n.ahead = NULL,
  credible = 0.95,
  FUN = mean,
  rolling = NULL,
  point = c("auto", "mean", "median"),
  ...
)
```

Arguments

`x`	An object of class `listmtar` obtained from `mtar_grid()`.
`newdata`	A data frame containing the future observations required to evaluate predictive performance.
`n.ahead`	A positive integer specifying the number of forecast steps ahead used in the predictive performance evaluation.
`credible`	A numeric value in the interval $(0, 1)$ specifying the posterior probability content of the prediction intervals. By default, `credible = 0.95`.
`FUN`	A function used to summarize the predictive accuracy measures across observations. By default, `FUN = mean`.
`rolling`	An optional positive integer specifying the number of forecast horizons to include in the rolling-window evaluation. If `NULL`, the predictive accuracy measures are computed using the original estimation sample.
`point`	A character string specifying the point estimator used for the forecasts. Possible values are `"auto"`, `"mean"`, and `"median"`. When `"auto"` (the default), the mean is used for symmetric error distributions, whereas the median is used for asymmetric error distributions. When `"mean"` or `"median"` is specified, the corresponding estimator is used regardless of the error distribution.
`...`	Additional arguments passed to `FUN`.

Value

If `rolling = NULL`, a matrix containing the summarized predictive accuracy measures for the fitted models in `x`. If `rolling` is specified, a list of matrices is returned, with one matrix for each forecast horizon included in the rolling-window evaluation.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

Examples

```
###### Example 1: Returns of the closing prices of three financial indexes
data(returns)
fit1 <- mtar_grid(~ COLCAP + BOVESPA | SP500, data=returns, row.names=Date,
                  subset={Date<="2015-12-07"}, dist=c("Gaussian","Student-t",
                  "Slash","Laplace"), nregim.min=2, nregim.max=3, p.min=2,
                  p.max=2, n.burnin=1000, n.sim=2000, n.thin=2,
                  plan_strategy="multisession")
oos1 <- out_of_sample(fit1, newdata=subset(returns, Date>"2015-12-07"),
                      n.ahead=75, FUN=median)
oos1

###### Example 2: Rainfall and two river flows in Colombia
data(riverflows)
fit2 <- mtar_grid(~ Bedon + LaPlata | Rainfall, data=riverflows,
                  row.names=Date, subset={Date<="2009-02-13"},dist="Laplace",
                  nregim.min=2, nregim.max=3, p.min=1, p.max=3,n.burnin=1000,
                  n.sim=2000, n.thin=2, plan_strategy="multisession")
oos2 <- out_of_sample(fit2, newdata=subset(riverflows, Date>"2009-02-13"),
                      n.ahead=60, FUN=median)
oos2

###### Example 3: Temperature, precipitation, and two river flows in Iceland
data(iceland.rf)
fit3 <- mtar_grid(~ Jokulsa + Vatnsdalsa | Temperature | Precipitation,
                  data=iceland.rf,subset={Date<="1974-11-06"},row.names=Date,
                  dist=c("Slash","Student-t"), nregim.min=1, nregim.max=2,
                  p.min=15, p.max=15, q.min=4, q.max=4, d.min=2, d.max=2,
                  n.burnin=1000, n.sim=2000, n.thin=2,
                  plan_strategy="multisession")
oos3 <- out_of_sample(fit3, newdata=subset(iceland.rf, Date>"1974-11-06"),
                      n.ahead=55, FUN=median)
oos3

###### Example 4: U.S. stock returns
data(US.returns)
fit4 <- mtar_grid(~ CCR | dVIX, data=US.returns, subset={Date<="2025-11-28"},
                  row.names=Date, dist=c("Laplace","Student-t","Slash"),
                  nregim.min=2, nregim.max=2, p.min=3, p.max=3, d.min=3,
                  d.max=3, n.burnin=1000, n.sim=2000, n.thin=2,
                  plan_strategy="multisession")
```

```
oos4 <- out_of_sample(fit4, newdata=subset(US.returns, Date>"2025-11-28"),
                      n.ahead=100, FUN=median)
oos4
```

`out_of_sample.mtar` *Out-of-sample Predictive Accuracy for* `mtar` *Objects*

Description

Computes out-of-sample predictive accuracy measures for one or more fitted objects of class `mtar`. The measures are obtained by comparing the predictive distributions with the observed future values of the output series.

Usage

```
## S3 method for class 'mtar'
out_of_sample(
  ...,
  newdata,
  n.ahead = NULL,
  credible = 0.95,
  FUN = mean,
  rolling = NULL,
  point = c("auto", "mean", "median")
)
```

Arguments

`...`	One or more fitted objects of class `mtar`.
`newdata`	A data frame containing the future observations required to evaluate predictive performance.
`n.ahead`	A positive integer specifying the number of forecast steps ahead used in the predictive performance evaluation.
`credible`	A numeric value in the interval $(0, 1)$ specifying the posterior probability content of the prediction intervals. By default, `credible = 0.95`.
`FUN`	A function used to summarize the predictive accuracy measures across observations. By default, `FUN = mean`.
`rolling`	An optional positive integer specifying the number of forecast horizons to include in the rolling-window evaluation. If `NULL`, the predictive accuracy measures are computed using the original estimation sample.
`point`	A character string specifying the point estimator used for the forecasts. Possible values are `"auto"`, `"mean"`, and `"median"`. When `"auto"` (the default), the mean is used for symmetric error distributions, whereas the median is used for asymmetric error distributions. When `"mean"` or `"median"` is specified, the corresponding estimator is used regardless of the error distribution.

Value

If `rolling = NULL`, a matrix containing one row for each fitted `mtar` object and columns corresponding to the predictive accuracy measures. These include the Log Score, Energy Score, and, for each component of the output series, the Absolute Error, Absolute Percentage Error, Squared Error, prediction interval Width, and Coverage Rate. If `rolling` is specified, a list of matrices is returned, with one matrix for each forecast horizon included in the rolling-window evaluation.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

`plot.fittedmtar` *Plot Fitted Values from an* `fittedmtar` *Object*

Description

Produces plots comparing the observed and fitted values for each component of the multivariate output series of an `mtar` model.

Usage

```
## S3 method for class 'fittedmtar'
plot(x, ..., last = NULL, observed = list(), fitted = list())
```

Arguments

`x`	An object of class `fittedmtar` returned by `fitted.mtar()`.
`...`	Additional graphical arguments.
`last`	An optional positive integer specifying the number of most recent observations to display. By default, all available
`observed`	An optional list of graphical parameters used to display the observed series.
`fitted`	An optional list of graphical parameters used to display the fitted series.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

`plot.mtarmcmc` *Diagnostic plots for* `mtarmcmc` *objects*

Description

Produces trace plots and posterior density plots for the MCMC samples stored in an object of class `mtarmcmc`. Plots are produced separately for the parameters associated with each regime and, when applicable, for the skewness, additional distributional, threshold, and delay parameters.

Usage

```
## S3 method for class 'mtarmcmc'
plot(
  x,
  trace = TRUE,
  density = TRUE,
  smooth = FALSE,
  bwf = NULL,
  auto.layout = TRUE,
  ask = dev.interactive(),
  ...
)
```

Arguments

`x`	An object of class `mtarmcmc`.
`trace`	A logical indicating whether trace plots should be displayed. By default, `trace = TRUE`.
`density`	A logical indicating whether posterior density plots should be displayed. By default, `density = TRUE`.
`smooth`	A logical indicating whether a smoothed density estimate should be used. By default, `smooth = FALSE`.
`bwf`	An optional bandwidth adjustment passed to the underlying plotting method.
`auto.layout`	A logical indicating whether the graphical layout should be determined automatically. By default, `auto.layout = TRUE`.
`ask`	A logical indicating whether the user should be prompted before a new plot is displayed. By default, `ask = dev.interactive()`.
`...`	Further arguments passed to the underlying plotting methods.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

`plot.predict_mtar` *Plot Forecasts from a* `predict_mtar` *Object*

Description

Produces graphical displays of the observed output series and the forecasts obtained from a fitted multivariate TAR model. For each component of the output series, the plot displays the historical observations, the posterior forecast, and the corresponding credible interval.

Usage

```
## S3 method for class 'predict_mtar'
plot(
  x,
  ...,
  last,
  historical = list(),
  forecasts = list(),
  forecasts.PI = list()
)
```

Arguments

`x`	An object of class `predict_mtar` returned by `predict.mtar()`.
`...`	Additional graphical arguments.
`last`	An optional positive integer specifying the number of historical observations to display together with the forecasts. By default, all available historical observations are displayed.
`historical`	An optional list of graphical parameters passed to the `plot()` method for the historical output series.
`forecasts`	An optional list of graphical parameters passed to the `plot()` method for the forecasts.
`forecasts.PI`	An optional list of graphical parameters passed to `polygon()` for displaying the forecast credible intervals.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

plot.residuals_mtar *Plot Residual Diagnostics for* mtar *Models*

Description

Produces a normal Q–Q plot and a histogram of the full residuals contained in an object of class `residuals_mtar`. A standard normal density is superimposed on the histogram.

Usage

```
## S3 method for class 'residuals_mtar'
plot(x, ...)
```

Arguments

`x`	An object of class `residuals_mtar` returned by `residuals.mtar()`.
`...`	Additional graphical arguments.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

predict.mtar *Forecasting for Multivariate TAR Models*

Description

Computes forecasts and posterior credible intervals from a fitted multivariate Threshold Autoregressive (TAR) model. Forecasts are generated from the posterior predictive distribution obtained from the MCMC samples of the fitted model. The point forecast can be obtained using either the posterior predictive mean or median. By default, the mean is used for symmetric error distributions, whereas the median is used for asymmetric error distributions. When requested, the function also computes out-of-sample predictive accuracy measures by comparing the forecasts with the observed future values of the output series. Rolling-window forecasts can additionally be obtained using a fixed forecast window.

Usage

```
## S3 method for class 'mtar'
predict(
  object,
  ...,
  newdata,
  n.ahead = NULL,
```

```
  row.names,
  credible = 0.95,
  out.of.sample = FALSE,
  rolling = NULL,
  point = c("auto", "mean", "median")
)
```

Arguments

object	An object of class mtar obtained from a call to mtar().
...	Additional arguments that may affect the prediction method.
newdata	An optional data.frame containing the future values required to construct the predictors used for forecasting. Depending on the fitted model, these may include the threshold series and the exogenous series. When out.of.sample = TRUE, newdata must also contain the observed future values of the output series.
n.ahead	A positive integer specifying the number of steps ahead to forecast.
row.names	An optional variable in newdata specifying the labels associated with the forecast periods.
credible	A numeric value in the interval $(0, 1)$ specifying the posterior probability content of the credible intervals. By default, credible = 0.95.
out.of.sample	A logical indicating whether out-of-sample predictive accuracy measures should be computed. When out.of.sample = TRUE, newdata must contain the observed future values of the output series. The function then computes the Log Score (LS), Energy Score (ES), Absolute Error (AE), Absolute Percentage Error (APE), Squared Error (SE), prediction interval width (Width), and coverage rate (CR). By default, out.of.sample = FALSE.
rolling	An optional positive integer specifying the number of forecast horizons included in the rolling-window evaluation. When rolling = NULL, forecasts are evaluated from the original estimation sample without rolling-window forecasting. When supplied, rolling must be no greater than n.ahead and out.of.sample must be TRUE.
point	A character string specifying the point estimator used for the forecasts. Possible values are "auto", "mean", and "median". When "auto" (the default), the mean is used for symmetric error distributions, whereas the median is used for asymmetric error distributions. When "mean" or "median" is specified, the corresponding estimator is used regardless of the error distribution.

References

Nieto, F.H. (2005) Modeling Bivariate Threshold Autoregressive Processes in the Presence of Missing Data. Communications in Statistics - Theory and Methods, 34, 905-930.

Romero, L.V. and Calderon, S.A. (2021) Bayesian estimation of a multivariate TAR model when the noise process follows a Student-t distribution. Communications in Statistics - Theory and Methods, 50, 2508-2530.

Calderon, S.A. and Nieto, F.H. (2017) Bayesian analysis of multivariate threshold autoregressive models with missing data. Communications in Statistics - Theory and Methods, 46, 296-318.

Karlsson, S. (2013) Chapter 15-Forecasting with Bayesian Vector Autoregression. In Elliott, G. and Timmermann, A. Handbook of Economic Forecasting, Volume 2, 791–89, Elsevier.

Vanegas, L.H. and Calderón, S.A. and Rondón, L.M. (2025) Bayesian estimation of a multivariate tar model when the noise process distribution belongs to the class of gaussian variance mixtures. International Journal of Forecasting.

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

Examples

```
###### Example 1: Returns of the closing prices of three financial indexes
data(returns)
fit1 <- mtar(~ COLCAP + BOVESPA | SP500, data=returns, row.names=Date,
             subset={Date<="2015-12-07"}, dist="Student-t",
             ars=ars(nregim=3,p=c(1,1,2)), n.burnin=1000, n.sim=2000,
             n.thin=2)
p1 <- predict(fit1, newdata=subset(returns,Date>"2015-12-07"), n.ahead=75,
              credible=0.8, out.of.sample=TRUE)
with(p1,summary)
with(p1,cbind(LS,ES,APE,CR))
plot(p1,last=100)

###### Example 2: Rainfall and two river flows in Colombia
data(riverflows)
fit2 <- mtar(~ Bedon + LaPlata | Rainfall, data=riverflows, row.names=Date,
             subset={Date<="2009-02-13"}, dist="Laplace",
             ars=ars(nregim=3,p=5), n.burnin=1000, n.sim=2000, n.thin=2)
p2 <- predict(fit2, newdata=subset(riverflows,Date>"2009-02-13"), n.ahead=60,
              credible=0.8, out.of.sample=TRUE)
with(p2,summary)
with(p2,cbind(LS,ES,APE,CR))
plot(p2,last=100)

###### Example 3: Temperature, precipitation, and two river flows in Iceland
data(iceland.rf)
fit3 <- mtar(~ Jokulsa + Vatnsdalsa | Temperature | Precipitation,
             data=iceland.rf, subset={Date<="1974-11-06"}, row.names=Date,
             ars=ars(nregim=2,p=15,q=4,d=2), n.burnin=1000, n.sim=2000,
             n.thin=2, dist="Slash")
p3 <- predict(fit3, newdata=subset(iceland.rf,Date>"1974-11-06"), n.ahead=55,
              credible=0.8, out.of.sample=TRUE)
with(p3,summary)
with(p3,cbind(LS,ES,APE,CR))
plot(p3,last=100)

###### Example 4: U.S. stock returns
data(US.returns)
fit4 <- mtar(~ CCR | dVIX, data=US.returns, subset={Date<="2025-11-28"},
             row.names=Date, ars=ars(nregim=2,p=3,d=3), n.burnin=1000,
             n.sim=2000, n.thin=2, dist="Student-t")
p4 <- predict(fit4, newdata=subset(US.returns,Date>"2025-11-28"),n.ahead=100,
              credible=0.8, out.of.sample=TRUE)
```

```
with(p4,summary)
with(p4,cbind(LS,ES,APE,CR))
plot(p4,last=100)
```

`print.essmtar` *Print Effective Sample Size Results for* `mtar` *Objects*

Description

Prints the effective sample size (ESS) estimates obtained from the Markov chain Monte Carlo (MCMC) samples of a fitted multivariate Threshold Autoregressive (TAR) model. The effective sample size provides an estimate of the number of independent draws represented by the MCMC sample, accounting for autocorrelation in the simulated chains.

Usage

```
## S3 method for class 'essmtar'
print(x, digits = max(3, getOption("digits") - 2), ...)
```

Arguments

`x`	An object of class `essmtar` returned by `effectiveSize()`.
`digits`	An integer specifying the number of decimal places to be used when printing the results. The default is `4`.
`...`	Additional arguments passed to or from other methods.

Value

Invisibly returns `x`.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

print.gdmtar *Print Geweke Diagnostic Results for* mtar *Objects*

Description

Prints the results of the Geweke convergence diagnostic for the Markov chain Monte Carlo (MCMC) samples obtained from a fitted multivariate Threshold Autoregressive (TAR) model. The Geweke diagnostic compares the means of an early portion and a late portion of the MCMC chain. The output reports the corresponding diagnostic statistics and p-values for the monitored parameters.

Usage

```
## S3 method for class 'gdmtar'
print(x, digits = max(3, getOption("digits") - 2), ...)
```

Arguments

x	An object of class gdmtar returned by geweke_diagTAR().
digits	An integer specifying the number of significant digits to be printed. By default, digits is set to max(3, getOption("digits") - 2).
...	Further arguments passed to or from other methods.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

print.HPDmtar *Print Highest Posterior Density Intervals*

Description

Prints Highest Posterior Density (HPD) intervals stored in an object of class HPDmtar. The output includes the HPD intervals for the regime-specific location and scale parameters and, when applicable, the threshold, skewness, and additional distributional parameters.

Usage

```
## S3 method for class 'HPDmtar'
print(x, digits = max(3, getOption("digits") - 2), ...)
```

Arguments

x	An object of class HPDmtar obtained from HPDinterval.mtar().
digits	An integer specifying the number of significant digits to be printed. By default, digits is set to max(3, getOption("digits") - 2).
...	Further arguments passed to or from other methods.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

`print.listmtar` *Print a List of* `mtar` *Models*

Description

Prints a summary of the model specifications represented by an object of class `listmtar`. The output describes the common model components and the ranges of regimes and lag orders across the fitted models.

Usage

```
## S3 method for class 'listmtar'
print(x, ...)
```

Arguments

`x`	An object of class `listmtar`, typically returned by `mtar_grid()`.
`...`	Further arguments passed to or from other methods.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

`print.mtar` *Print an* `mtar` *Model*

Description

Prints the main specification of a fitted multivariate Threshold Autoregressive (TAR) model, including the output, threshold, and exogenous series, noise distribution, number of regimes, deterministic terms, and regime-specific lag orders.

Usage

```
## S3 method for class 'mtar'
print(x, ...)
```

Arguments

`x`	An object of class `mtar` obtained from a call to `mtar()`.
`...`	Further arguments passed to or from other methods.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

`print.residuals_mtar` *Print Quantiles of* `mtar` *Residuals*

Description

Prints selected empirical quantiles of the full and componentwise residuals contained in an object of class `residuals_mtar`.

Usage

```
## S3 method for class 'residuals_mtar'
print(x, digits = max(3, getOption("digits") - 2), ...)
```

Arguments

`x`	An object of class `residuals_mtar` returned by `residuals.mtar()`.
`digits`	An integer specifying the number of significant digits to be printed. By default, `digits` is set to `max(3, getOption("digits") - 2)`.
`...`	Further arguments passed to or from other methods.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

`print.summary_mtar` *Print Summaries of* `mtar` *Objects*

Description

Prints the posterior summaries contained in an object of class `summary_mtar`. The output includes information about the fitted model, threshold intervals when applicable, and posterior summaries of the regime-specific location and scale parameters.

Usage

```
## S3 method for class 'summary_mtar'
print(x, digits = max(3, getOption("digits") - 2), ...)
```

Arguments

x	An object of class summary_mtar returned by summary.mtar().
digits	An integer specifying the number of significant digits to be printed. By default, digits is set to max(3, getOption("digits") - 2).
...	Further arguments passed to or from other methods.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

---

print.summtarmcmc *Print method for* summtarmcmc *objects*

Description

Prints a summary of the posterior MCMC output stored in an object of class summtarmcmc, including the iteration range, thinning interval, sample size, and posterior summaries for the model parameters.

Usage

```
## S3 method for class 'summtarmcmc'
print(x, ..., digits = max(3, getOption("digits") - 2))
```

Arguments

x	An object of class summtarmcmc.
...	Further arguments passed to or from other methods.
digits	An integer specifying the number of significant digits to be printed. By default, digits is determined from the digits option.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

`priors` *Auxiliary Function for Setting Hyperparameter Values*

Description

Constructs and validates the hyperparameter values used to define the prior distributions of the model parameters. Hyperparameters not explicitly provided by the user are assigned default values.

Usage

```
priors(prior, regim, k, dist, setar, ssvs)
```

Arguments

`prior`	A list specifying user-defined values for the hyperparameters. Hyperparameters not included in this list are assigned their default values.
`regim`	A positive integer specifying the number of regimes in the model.
`k`	A positive integer specifying the dimension of the multivariate output series.
`dist`	A character string specifying the distribution used to model the noise process.
`setar`	A positive integer specifying the component of the output series used as the threshold variable in a SETAR specification. If `NULL`, the threshold variable is specified through the second part of the model formula.
`ssvs`	A logical indicating whether Stochastic Search Variable Selection (SSVS) is used for variable selection.

Value

A list containing the completed and validated hyperparameter values defining the prior distributions of the model parameters.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

residuals.mtar *Compute Residuals from an* mtar *Object*

Description

Computes diagnostic residuals from a fitted multivariate Threshold Autoregressive (TAR) model. The function returns both componentwise residuals and a full multivariate residual measure. For symmetric noise distributions, the residuals are standardized using the estimated scale matrix and transformed using a probability integral transform. For skewed noise distributions, the transformation is obtained by comparison with a simulated reference distribution.

Usage

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

Arguments

object	An object of class mtar obtained from a call to mtar().
...	Additional arguments.

Value

An object of class residuals_mtar containing two components: full, the full multivariate residual measure, and by.component, the residuals for each component of the output series.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

returns *Returns of the Closing Prices of Three Financial Indexes*

Description

This dataset contains daily returns computed from the closing prices of the COLCAP, BOVESPA, and S&P 500 stock market indexes over the period from February 10, 2010, to March 31, 2016, comprising 1,505 observations. The COLCAP index reflects the price dynamics of the most liquid stocks traded on the Colombian stock market. The BOVESPA index represents the Brazilian stock market. The Standard & Poor's 500 (S&P 500) index tracks the performance of 500 large-cap companies listed in the United States.

Usage

```
data(returns)
```

Format

A data frame with 1,505 rows and 4 variables:

Date A vector containing the date of each observation.

COLCAP A numeric vector containing the returns of the COLCAP index.

SP500 A numeric vector containing the returns of the S&P 500 index.

BOVESPA A numeric vector containing the returns of the BOVESPA index.

References

Romero, L. V. and Calderon, S. A. (2021). Bayesian estimation of a multivariate TAR model when the noise process follows a Student-t distribution. *Communications in Statistics - Theory and Methods*, 50, 2508–2530.

Examples

```
data(returns)
dev.new()
plot(ts(as.matrix(returns[, -1])), main = "Returns")#'
```

---

riverflows *Rainfall and Two River Flows in Colombia*

Description

This dataset contains daily observations of rainfall, in millimeters, and river flows, in m^3/s, for two rivers in southern Colombia. Rainfall was recorded at a meteorological station located at an altitude of 2,400 meters above sea level. River flows were measured at two hydrological stations: El Trébol station, which monitors the Bedón River at an altitude of 1,720 meters, and Villalosada station, which monitors the La Plata River at an altitude of 1,300 meters. The stations are located near the equator, a geographic feature that helps reduce seasonal distortions and facilitates the analysis of the dynamic relationship between rainfall and river flows. The sample period spans from January 1, 2006, to April 14, 2009.

Usage

```
data(riverflows)
```

Format

A data frame with 1,200 rows and 4 variables:

Date A vector containing the date of each observation.

Bedon A numeric vector containing the daily flow of the Bedón River, in cubic meters per second.

LaPlata A numeric vector containing the daily flow of the La Plata River, in cubic meters per second.

Rainfall A numeric vector containing daily rainfall amounts, in millimeters.

References

Calderon, S. A. and Nieto, F. H. (2017). Bayesian analysis of multivariate threshold autoregressive models with missing data. *Communications in Statistics - Theory and Methods*, 46, 296–318.

Examples

```
data(riverflows)
dev.new()
plot(ts(as.matrix(riverflows[, -1])), main = "Rainfall and River Flows")
```

`simtar` *Simulation of Multivariate Time Series from a TAR Model*

Description

Simulates a multivariate time series from a user-specified Threshold Autoregressive (TAR) model. The model may include multiple regimes, autoregressive terms, exogenous variables, threshold variables, deterministic terms, and seasonal effects. Several multivariate distributions are available for the noise process.

Usage

```
simtar(
  n,
  k = 2,
  ars = ars(),
  Intercept = TRUE,
  trend = c("none", "linear", "quadratic"),
  nseason = NULL,
  parms,
  delay = 0,
  thresholds = NULL,
  t.series = NULL,
  ex.series = NULL,
  dist = c("Gaussian", "Student-t", "Hyperbolic", "Laplace", "Slash",
    "Contaminated normal", "Skew-Student-t", "Skew-normal"),
  skewness = NULL,
  extra = NULL,
  setar = NULL,
  Verbose = TRUE
)
```

Arguments

`n`	A positive integer specifying the length of the simulated output series.
`k`	A positive integer specifying the dimension of the multivariate output series.

`ars` A list defining the TAR model structure, composed of `nregim`, the number of regimes, `p`, the autoregressive orders, `q`, the maximum lags of the exogenous series, and `d`, the maximum lags of the threshold series within each regime. This object can be constructed using `ars()`.

`Intercept` An optional logical indicating whether an intercept is included in each regime. By default, `Intercept = TRUE`.

`trend` An optional character string specifying the degree of deterministic time trend included in each regime. Available options are `"none"`, `"linear"`, and `"quadratic"`. By default, `trend = "none"`.

`nseason` An optional integer, greater than or equal to 2, specifying the number of seasonal periods. When provided, `nseason - 1` seasonal dummy variables are included in the regressors within each regime. By default, `nseason = NULL`.

`parms` A list containing the parameters defining the model, with one sublist for each regime. Each sublist contains a `location` matrix and a `scale` matrix. Alternatively, `parms` can be an object of class `mtar`, in which case the model parameters are extracted from the fitted object.

`delay` An optional non-negative integer specifying the delay parameter of the threshold series. By default, `delay = 0`. For SETAR models, the delay parameter must be positive when more than one regime is specified.

`thresholds` A numeric vector containing the threshold values separating the regimes. For a model with `nregim` regimes, `thresholds` must contain `nregim - 1` values. The values are sorted in ascending order.

`t.series` A vector containing the values of the threshold series used in a TAR model. It must contain the observations required to construct the specified lags.

`ex.series` A matrix containing the values of the multivariate exogenous series. It is required when at least one regime has a positive value of `q`.

`dist` An optional character string specifying the multivariate distribution used to model the noise process. Available options are `"Gaussian"`, `"Student-t"`, `"Hyperbolic"`, `"Laplace"`, `"Slash"`, `"Contaminated normal"`, `"Skew-normal"`, and `"Skew-Student-t"`. By default, `dist = "Gaussian"`.

`skewness` An optional numeric vector specifying the skewness parameters of the noise distribution. This argument is required for `"Skew-normal"` and `"Skew-Student-t"` distributions and must have length `k`.

`extra` An optional value specifying the additional distribution- specific parameter required by the selected noise distribution. For `"Contaminated normal"`, this is a vector of length two.

`setar` An optional positive integer specifying the component of the output series used as the threshold variable in a SETAR model. By default, `setar = NULL`, in which case an external threshold series is used for a TAR model.

`Verbose` An optional logical indicating whether information about the simulated TAR model should be printed. By default, `Verbose = TRUE`.

Value

A `data.frame` containing the simulated multivariate output series and a variable identifying the regime associated with each observation. For TAR models, the threshold series is also included when supplied. If exogenous variables are used, they are included as well.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

Vanegas, L. H., Calderón, S. A., and Rondón, L. M. (2025). Bayesian estimation of a multivariate TAR model when the noise process distribution belongs to the class of Gaussian variance mixtures. *International Journal of Forecasting*.

Examples

```
###### Simulation of a trivariate TAR model with two regimes
n <- 2000
k <- 3
myars <- ars(nregim=2,p=c(1,2))
Z <- as.matrix(arima.sim(n=n+max(myars$p),list(ar=c(0.5))))
probs <- sort((0.6 + runif(myars$nregim-1)*0.8)*c(1:(myars$nregim-1))/myars$nregim)
dist <- "Student-t"; extra <- 6
parms <- list()
for(j in 1:myars$nregim){
    np <- 1 + myars$p[j]*k
    parms[[j]] <- list()
   parms[[j]]$location <- c(ifelse(runif(np*k)<=0.5,1,-1)*rbeta(np*k,shape1=4,shape2=16))
    parms[[j]]$location <- matrix(parms[[j]]$location,np,k)
    parms[[j]]$scale <- rgamma(k,shape=1,scale=1)*diag(k)
}
thresholds <- quantile(Z,probs=probs)
out1 <- simtar(n=n, k=k, ars=myars, parms=parms, thresholds=thresholds,
               t.series=Z, dist=dist, extra=extra)
str(out1)

fit1 <- mtar(~ Y1 + Y2 + Y3 | Z, data=out1, ars=myars, dist=dist,
             n.burn=2000, n.sim=3000, n.thin=2)
summary(fit1)

###### Simulation of a trivariate VAR model
n <- 2000
k <- 3
myars <- ars(nregim=1,p=2)
dist <- "Slash"; extra <- 2
parms <- list()
for(j in 1:myars$nregim){
    np <- 1 + myars$p[j]*k
    parms[[j]] <- list()
   parms[[j]]$location <- c(ifelse(runif(np*k)<=0.5,1,-1)*rbeta(np*k,shape1=4,shape2=16))
    parms[[j]]$location <- matrix(parms[[j]]$location,np,k)
    parms[[j]]$scale <- rgamma(k,shape=1,scale=1)*diag(k)
```

```
}
out2 <- simtar(n=n, k=k, ars=myars, parms=parms, dist=dist, extra=extra)
str(out2)

fit2 <- mtar(~ Y1 + Y2 + Y3, data=out2, ars=myars, dist=dist,
             n.burn=2000, n.sim=3000, n.thin=2)
summary(fit2)

n <- 5000
k <- 3
myars <- ars(nregim=2,p=c(1,2))
dist <- "Laplace"
parms <- list()
for(j in 1:myars$nregim){
    np <- 1 + myars$p[j]*k
    parms[[j]] <- list()
   parms[[j]]$location <- c(ifelse(runif(np*k)<=0.5,1,-1)*rbeta(np*k,shape1=4,shape2=16))
    parms[[j]]$location <- matrix(parms[[j]]$location,np,k)
    parms[[j]]$scale <- rgamma(k,shape=1,scale=1)*diag(k)
}
out3 <- simtar(n=n, k=k, ars=myars, parms=parms, delay=2,
               thresholds=-1, dist=dist, setar=2)
str(out3)

fit3 <- mtar(~ Y1 + Y2 + Y3, data=out3, ars=myars, dist=dist,
             n.burn=2000, n.sim=3000, n.thin=2, setar=2)
summary(fit3)
```

`summary.mtar` *Summarize Fitted* `mtar` *Models*

Description

Summarizes the posterior MCMC samples obtained from a fitted multivariate Threshold Autoregressive (TAR) model. For each model parameter, the function reports the posterior mean, twice one minus the posterior probability of direction, and the bounds of the highest-density interval (HDI). When SSVS is used, the posterior inclusion probabilities of the output, exogenous, and threshold lags are also reported.

Usage

```
## S3 method for class 'mtar'
summary(object, credible = 0.95, ...)
```

Arguments

`object`	An object of class `mtar` obtained from a call to `mtar()`.
`credible`	A numeric value in the interval $(0, 1)$ specifying the posterior probability content of the HDI. By default, `credible = 0.95`.
`...`	Additional arguments.

Value

An object of class `summary_mtar` containing posterior summaries of the fitted model. The object includes summaries of the location and scale parameters for each regime and, when applicable, the delay, threshold, skewness, and additional distributional parameters. When SSVS is used, posterior inclusion probabilities are also included.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

`summary.mtarmcmc` *Posterior summary for* `mtarmcmc` *objects*

Description

Computes posterior summary statistics and selected posterior quantiles for an object of class `mtarmcmc`. For each model parameter, the summary includes its posterior mean, posterior standard deviation, Monte Carlo standard error, and the posterior quantiles specified by `quantiles`.

Usage

```
## S3 method for class 'mtarmcmc'
summary(object, ..., quantiles = c(0.025, 0.25, 0.5, 0.75, 0.975))
```

Arguments

`object`	An object of class `mtarmcmc`.
`...`	Further arguments passed to or from other methods.
`quantiles`	A numeric vector containing the posterior probabilities at which quantiles are to be computed. By default, `quantiles = c(0.025, 0.25, 0.5, 0.75, 0.975)`.

Value

An object of class `summtarmcmc` containing posterior summary statistics for the parameters of the fitted model.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

US.returns *U.S. Stock Returns*

Description

This dataset contains observations of the S&P 500 index, the Chicago Board Options Exchange Volatility Index (VIX), continuously compounded returns and simple returns of the S&P 500, and a differenced measure of the VIX. The sample period spans from January 5, 2005, to April 24, 2026.

Usage

```
data(US.returns)
```

Format

A data frame with 5,420 rows and 6 variables:

Date A vector containing the date of each observation.

SP500 A numeric vector containing the S&P 500 index.

VIX A numeric vector containing the Chicago Board Options Exchange Volatility Index (VIX).

CCR A numeric vector containing the continuously compounded returns of the S&P 500.

SR A numeric vector containing the simple returns of the S&P 500.

dVIX A numeric vector containing the difference $VIX_{t-1} - VIX_{t-2}$.

References

Massacci, D. (2014). A two-regime threshold model with conditional skewed Student-t distributions for stock returns. *Economic Modelling*, 43, 9–20.

Examples

```
data(US.returns)
dev.new()
plot(ts(as.matrix(US.returns[, -1])))
```

vcov.mtar *Extract Scale Matrix Estimates from* mtar *Objects*

Description

Extracts posterior summaries of the regime-specific scale matrices from a fitted multivariate Threshold Autoregressive (TAR) model. The summaries are obtained by applying a user-specified function to the MCMC samples of the scale parameters.

Usage

```
## S3 method for class 'mtar'
vcov(object, ..., FUN = mean)
```

Arguments

object	An object of class mtar obtained from a call to mtar().
...	Additional arguments passed to FUN.
FUN	A function applied to the MCMC samples of the scale parameters to obtain point estimates. By default, FUN = mean.

Value

A list containing one scale matrix for each regime. The elements of each matrix correspond to the summaries obtained by applying FUN to the posterior MCMC samples of the scale parameters.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

WAIC *Watanabe-Akaike or Widely Applicable Information Criterion (WAIC)*

Description

Computes the Watanabe-Akaike Information Criterion (WAIC) for models estimated using Bayesian methods. The WAIC is a Bayesian measure of predictive performance based on the pointwise posterior predictive distribution and includes a correction for model complexity.

Usage

```
WAIC(...)
```

Arguments

...	One or more fitted model objects of the same class.

Value

A numeric matrix containing the WAIC values corresponding to the fitted objects supplied in ....

References

Watanabe S. (2010). Asymptotic Equivalence of Bayes Cross Validation and Widely Applicable Information Criterion in Singular Learning Theory. The Journal of Machine Learning Research, 11, 3571–3594.

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

`WAIC.listmtar` *Watanabe-Akaike or Widely Applicable Information Criterion (WAIC) for a list of* `mtar` *objects*

Description

#' Computes the Watanabe-Akaike Information Criterion (WAIC) for each `mtar` object contained in a list.

Usage

```
## S3 method for class 'listmtar'
WAIC(x, ...)
```

Arguments

`x`	A list containing fitted objects of class `mtar`.
...	Further arguments passed to or from other methods.

Value

A numeric matrix containing one WAIC value for each `mtar` object in `x`. The row names are taken from the names of `x`.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

`WAIC.mtar` *Watanabe-Akaike or Widely Applicable Information Criterion (WAIC) for objects of class* `mtar`

Description

Computes the Watanabe-Akaike Information Criterion (WAIC) for one or more fitted objects of class `mtar`. The criterion is computed from the pointwise predictive distributions obtained from the posterior samples generated during Bayesian estimation.

Usage

```
## S3 method for class 'mtar'
WAIC(...)
```

Arguments

`...`	One or more fitted objects of class `mtar`.

Details

For models with Skew-normal and Skew-Student-t error distributions, the computation of the WAIC can be computationally demanding. This is because the likelihood evaluation requires the calculation of the probability of the latent variable being in the positive orthant. For the Skew-normal distribution, this probability is evaluated using a multivariate normal distribution, whereas for the Skew-Student-t distribution it is evaluated using a multivariate Student-t distribution. Consequently, the computation of the WAIC for these distributions may require considerably more computational time than for the other error distributions implemented in `mtar`.

Value

A numeric matrix containing one WAIC value for each `mtar` object supplied through `...`.

References

Vanegas L.H., Calderón S.A., Rondon L.M. (2026) mtarm: Bayesian Analysis of Multivariate Threshold Autoregressive Models in R.

See Also

DIC

Examples

```
###### Example 1: Returns of the closing prices of three financial indexes
data(returns)
fit1 <- mtar_grid(~ COLCAP + BOVESPA | SP500, data=returns, row.names=Date,
                  subset={Date<="2015-12-07"}, dist=c("Gaussian","Student-t",
```

```
                  "Slash","Laplace"), nregim.min=2, nregim.max=3, p.min=2,
                  p.max=2, n.burnin=1000, n.sim=2000, n.thin=2,
                  plan_strategy="multisession")
WAIC(fit1)

###### Example 2: Rainfall and two river flows in Colombia
data(riverflows)
fit2 <- mtar_grid(~ Bedon + LaPlata | Rainfall, data=riverflows,
                  row.names=Date, subset={Date<="2009-02-13"},dist="Laplace",
                  nregim.min=2, nregim.max=3, p.min=1, p.max=3,n.burnin=1000,
                  n.sim=2000, n.thin=2, plan_strategy="multisession")
WAIC(fit2)

###### Example 3: Temperature, precipitation, and two river flows in Iceland
data(iceland.rf)
fit3 <- mtar_grid(~ Jokulsa + Vatnsdalsa | Temperature | Precipitation,
                  data=iceland.rf,subset={Date<="1974-11-06"},row.names=Date,
                  dist=c("Slash","Student-t"), nregim.min=1, nregim.max=2,
                  p.min=15, p.max=15, q.min=4, q.max=4, d.min=2, d.max=2,
                  n.burnin=1000, n.sim=2000, n.thin=2,
                  plan_strategy="multisession")
WAIC(fit3)

###### Example 4: U.S. stock returns
data(US.returns)
fit4 <- mtar_grid(~ CCR | dVIX, data=US.returns, subset={Date<="2025-11-28"},
                  row.names=Date, dist=c("Laplace","Student-t","Slash"),
                  nregim.min=2, nregim.max=2, p.min=3, p.max=3, d.min=3,
                  d.max=3, n.burnin=1000, n.sim=2000, n.thin=2,
                  plan_strategy="multisession")
WAIC(fit4)
```

Index

∗ **datasets**
    iceland.rf, 16
    returns, 41
    riverflows, 42
    US.returns, 48

ars, 3
as.mcmc, *4*
as.mcmc.mtar, 3

coef.mtar, 5

DIC, 6, *21*, *51*
DIC.listmtar, 7
DIC.mtar, 7

effectiveSize, *9*
effectiveSize_TAR, 9

fitted.mtar, 10

geweke.diag, *12*
geweke.plot, *13, 14*
geweke_diagTAR, 11
geweke_plotTAR, 13

HPDinterval, *15*
HPDinterval.mtar, 15

iceland.rf, 16

model.matrix.mtar, 17
mtar, *11*, 18
mtar_grid, 22

out_of_sample, 25
out_of_sample.listmtar, 26
out_of_sample.mtar, 28

plot.fittedmtar, 29
plot.mtarmcmc, 30
plot.predict_mtar, 31
plot.residuals_mtar, 32
predict.mtar, 32
print.essmtar, 35
print.gdmtar, 36
print.HPDmtar, 36
print.listmtar, 37
print.mtar, 37
print.residuals_mtar, 38
print.summary_mtar, 38
print.summtarmcmc, 39
priors, 40

residuals.mtar, 41
returns, 41
riverflows, 42

simtar, 43
summary.mtar, 46
summary.mtarmcmc, 47

US.returns, 48

vcov.mtar, 49

WAIC, *8*, *21*, 49
WAIC.listmtar, 50
WAIC.mtar, 51