

# Package ‘uGMAR’

August 27, 2019

**Title** Estimate Univariate Gaussian or Student's t Mixture  
Autoregressive Model

**Version** 3.2.0

**Description** Maximum likelihood estimation of univariate Gaussian Mixture Autoregressive (GMAR), Student's t Mixture Autoregressive (StMAR) and Gaussian and Student's t Mixture Autoregressive (G-StMAR) models, quantile residual tests, graphical diagnostics, forecast and simulate from GMAR, StMAR and G-StMAR processes.

Leena Kalliovirta, Mika Meitz, Pentti Saikkonen (2015) <doi:10.1111/jtsa.12108>,  
Mika Meitz, Daniel Preve, Pentti Saikkonen (2018) <arXiv:1805.04010>.

**Depends** R (>= 3.4.0)

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Imports** Brodtingnag (>= 1.2-4), parallel, pbapply (>= 1.3-2), stats  
(>= 3.3.2)

**Suggests** gsl (>= 1.9-10.3), testthat, knitr, rmarkdown

**RoxygenNote** 6.1.1

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Savi Virolainen [aut, cre]

**Maintainer** Savi Virolainen <savi.virolainen@helsinki.fi>

**Repository** CRAN

**Date/Publication** 2019-08-27 10:10:02 UTC

## R topics documented:

add_data . . . . .	3
add_dfs . . . . .	4
all_pos_ints . . . . .	5
alt_gsmar . . . . .	6

calc_gradient . . . . .	7
changeRegime . . . . .	9
change_parametrization . . . . .	11
checkAndCorrectData . . . . .	12
checkConstraintMat . . . . .	13
checkPM . . . . .	14
check_data . . . . .	14
check_gsmar . . . . .	15
check_model . . . . .	15
check_params_length . . . . .	16
condMoments . . . . .	17
diagnosticPlot . . . . .	20
extractRegime . . . . .	22
fitGSMAR . . . . .	23
format_valuef . . . . .	28
GAfit . . . . .	28
getOmega . . . . .	32
get_ar_roots . . . . .	34
get_IC . . . . .	35
get_regime_autocovs . . . . .	35
get_regime_means . . . . .	36
get_regime_vars . . . . .	37
get_varying_h . . . . .	38
GSMAR . . . . .	40
IE . . . . .	44
isStationary . . . . .	44
isStationary_int . . . . .	46
iterate_more . . . . .	48
loglikelihood . . . . .	50
loglikelihood_int . . . . .	52
logVIX . . . . .	55
mixingWeights . . . . .	56
mixingWeights_int . . . . .	58
nParams . . . . .	60
parameterChecks . . . . .	61
pick_alphas . . . . .	62
pick_dfs . . . . .	63
pick_pars . . . . .	64
pick_phi0 . . . . .	66
plot.gsmarpred . . . . .	67
plot.qrtest . . . . .	68
predict.gsmar . . . . .	70
print.gsmarpred . . . . .	72
print.gsmarsum . . . . .	73
quantileResidualPlot . . . . .	74
quantileResiduals . . . . .	75
quantileResiduals_int . . . . .	78
randomIndividual . . . . .	80

randomIndividual_int . . . . .	83
random_arcoefs . . . . .	85
random_regime . . . . .	86
reformConstrainedPars . . . . .	87
reformParameters . . . . .	89
reformRestrictedPars . . . . .	90
regime_distance . . . . .	91
removeAllConstraints . . . . .	92
simulateGSMAR . . . . .	93
sortComponents . . . . .	95
standardErrors . . . . .	97
stmarpars_to_gstmar . . . . .	98
stmar_to_gstmar . . . . .	100
swap_parametrization . . . . .	102
uGSMAR . . . . .	103
uncondMoments . . . . .	103
uncondMoments_int . . . . .	104
VIX . . . . .	106
warn_dfs . . . . .	107

**Index****109**


---

add_data	<i>Add data to object of class 'gsmar' defining a GMAR, StMAR or G-StMAR model</i>
----------	--

---

**Description**

add\_data adds or updates data to object of class 'gsmar' that defines a GMAR, StMAR or G-StMAR model. Also calculates mixing weights, conditional moments and quantile residuals accordingly.

**Usage**

```
add_data(data, gsmar, calc_qresiduals = TRUE, calc_cond_moments = TRUE,
         calc_std_errors = FALSE, custom_h = NULL)
```

**Arguments**

data	a numeric vector class 'ts' object containing the data. NA values are not supported.
gsmar	object of class 'gsmar' created with the function fitGSMAR or GSMAR.
calc_qresiduals	should quantile residuals be calculated? Default is TRUE iff the model contains data.
calc_cond_moments	should conditional means and variances be calculated? Default is TRUE iff the model contains data.

`calc_std_errors` should approximate standard errors be calculated?

`custom_h` A numeric vector of with same length as the parameter vector of the estimated model:  $i$ :th element of `custom_h` is the difference used in central difference approximation for differentials of the log-likelihood function for the  $i$ :th parameter. If NULL (default), then the difference used for differentiating overly large degrees of freedom parameters is adjusted to avoid numerical problems, and the difference is  $6e-6$  for the other parameters.

### Value

Returns an object of class 'gsmar' defining the GMAR, StMAR or G-StMAR model with the data added to the model. If the object already contained data, the data will be updated. Does not modify the 'gsmar' object given as argument!

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for the G-StMAR model, but it's a straightforward generalization with theoretical properties similar to the GMAR and StMAR models.

### See Also

[fitGSMAR](#), [GSMAR](#), [iterate\\_more](#), [get\\_gradient](#), [get\\_regime\\_means](#), [swap\\_parametrization](#), [stmar\\_to\\_gstmar](#)

### Examples

```
# GMAR model without data
params12 <- c(0.18, 0.93, 0.01, 0.86, 0.68, 0.02, 0.88)
gmar12 <- GSMAR(p=1, M=2, params=params12, model="GMAR")
gmar12

# Add data to the model
gmar12 <- add_data(data=logVIX, gmar12)
gmar12
```

---

add\_dfs

*Add random dfs to a vector*

---

### Description

`add_dfs` adds random dfs to a vector

**Usage**

```
add_dfs(x, how_many)
```

**Arguments**

x                    a vector to add the dfs to  
how\_many            how many dfs?

**Details**

For details read the source code.

**Value**

Returns `c(ind,dfs)` with `how_many` dfs-elements.

---

`all_pos_ints`                    *Check whether all arguments are positive scalar whole numbers*

---

**Description**

`all_pos_ints` tells whether all the elements in a vector are strictly positive whole numbers.

**Usage**

```
all_pos_ints(x)
```

**Arguments**

x                    a vector containing the elements to be tested.

**Value**

Returns TRUE or FALSE accordingly.

---

alt_gsmar	<i>Construct a GSMAR model based on results from an arbitrary estimation round of fitGSMAR</i>
-----------	--

---

### Description

alt\_gsmar constructs a GSMAR model based on results from an arbitrary estimation round of fitGSMAR.

### Usage

```
alt_gsmar(gsmar, which_round = 1, calc_qresiduals = TRUE,
          calc_cond_moments = TRUE, calc_std_errors = TRUE, custom_h = NULL)
```

### Arguments

gsmar	object of class 'gsmar' created with the function fitGSMAR or GSMAR.
which_round	based on which estimation round should the model be constructed? An integer value in 1,...,ncalls.
calc_qresiduals	should quantile residuals be calculated? Default is TRUE iff the model contains data.
calc_cond_moments	should conditional means and variances be calculated? Default is TRUE iff the model contains data.
calc_std_errors	should approximate standard errors be calculated?
custom_h	A numeric vector of with same length as the parameter vector of the estimated model: i:th element of custom_h is the difference used in central difference approximation for differentials of the log-likelihood function for the i:th parameter. If NULL (default), then the difference used for differentiating overly large degrees of freedom parameters is adjusted to avoid numerical problems, and the difference is 6e-6 for the other parameters.

### Details

It's sometimes useful to examine other estimates than the one with the highest log-likelihood value. This function is just a simple wrapper to GSMAR that picks the correct estimates from an returned by fitGSMAR.

### Value

Returns an object of class 'gsmar' defining the specified GMAR, StMAR or G-StMAR model. If data is supplied, the returned object contains (by default) empirical mixing weights, conditional means and variances and quantile residuals. Note that the first p observations are taken as the initial values so mixing weights, conditional moments and qresiduals start from the p+1:th observation (interpreted as t=1).

## References

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for the G-StMAR model, but it's a straightforward generalization with theoretical properties similar to the GMAR and StMAR models.

## See Also

[fitGSMAR](#), [GSMAR](#), [iterate\\_more](#), [get\\_gradient](#), [get\\_regime\\_means](#), [swap\\_parametrization](#), [stmar\\_to\\_gstmar](#)

## Examples

```
# These are long running examples and use parallel computing
fit12t <- fitGSMAR(IE, 1, 2, model="StMAR", ncalls=2, seeds=1:2)
fit12t
fit12t2 <- alt_gsmar(fit12t, which_round=2)
fit12t2
```

---

calc\_gradient

*Calculate gradient or Hessian matrix*

---

## Description

calc\_gradient or calc\_hessian calculates the gradient or Hessian matrix of the given function at the given point using central difference numerical approximation. get\_gradient or get\_hessian calculates the gradient or Hessian matrix of the log-likelihood function at the parameter values of class 'gsmar' object. get\_soc returns eigenvalues of the Hessian matrix.

## Usage

```
calc_gradient(x, fn, h = 6e-06, varying_h = NULL, ...)
```

```
calc_hessian(x, fn, h = 6e-06, varying_h = NULL, ...)
```

```
get_gradient(gsmar, custom_h = NULL)
```

```
get_foc(gsmar, custom_h = NULL)
```

```
get_hessian(gsmar, custom_h = NULL)
```

```
get_soc(gsmar, custom_h = NULL)
```

**Arguments**

x	a numeric vector specifying the point where the gradient or Hessian should be calculated.
fn	a function that takes in argument x as the <b>first</b> argument.
h	difference used to approximate the derivatives.
varying_h	a numeric vector with the same length as x specifying the difference h for each dimension separately. If NULL (default), then the difference given as parameter h will be used for all dimensions.
...	other arguments passed to fn.
gsmar	object of class 'gsmar' created with the function fitGSMAR or GSMAR.
custom_h	same as varying_h but if NULL (default), then the difference h used for differentiating overly large degrees of freedom parameters is adjusted to avoid numerical problems, and the difference is 6e-6 for the other parameters.

**Details**

Especially the functions `get_foc` or `get_soc` can be used to check whether the found estimates denote a (local) maximum point, a saddle point or something else.

**Value**

Gradient functions return numerical approximation of the gradient, and Hessian functions return numerical approximation of the Hessian. `get_soc` returns eigenvalues of the Hessian matrix, `get_foc` is the same as `get_gradient` but named conveniently.

**Warning**

No argument checks!

**Examples**

```
# Simple function
foo <- function(x) x^2 + x
calc_gradient(x=1, fn=foo)
calc_gradient(x=-0.5, fn=foo)
calc_hessian(x=2, fn=foo)

# More complicated function
foo <- function(x, a, b) a*x[1]^2 - b*x[2]^2
calc_gradient(x=c(1, 2), fn=foo, a=0.3, b=0.1)
calc_hessian(x=c(1, 2), fn=foo, a=0.3, b=0.1)

# GMAR model:
params12 <- c(0.18281409, 0.92657275, 0.00214552,
  0.85725129, 0.68210294, 0.01900299, 0.88342018)
gmar12 <- GSMAR(logVIX, 1, 2, params12)
get_gradient(gmar12)
get_foc(gmar12)
```



```
get_hessian(gmar12)
get_soc(gmar12)
```

---

changeRegime	<i>Change the specified regime of parameter vector to the given regime-parameter vector</i>
--------------	---

---

### Description

changeRegime changes the specified regime of the parameter vector to correspond the given regime-parameter vector and returns the modified parameter vector. Does not affect mixing weight parameters.

### Usage

```
changeRegime(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL, regimeParams, regime)
```

### Arguments

- p** a positive integer specifying the order of AR coefficients.
- M** **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.  
**For G-StMAR model:** a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is M=M1+M2.
- params** a real valued parameter vector specifying the model.  
**For non-restricted models:** **For GMAR model:** Size  $(M(p+3) - 1x1)$  vector  $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .  
**For StMAR model:** Size  $(M(p+4) - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .  
**For G-StMAR model:** Size  $(M(p+3) + M2 - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .  
**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices **C** that satisfy  $\phi_m = R_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .  
**For restricted models:** **For GMAR model:** Size  $(3M + p - 1x1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .  
**For StMAR model:** Size  $(4M + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .  
**For G-StMAR model:** Size  $(3M + M2 + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .  
**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix **C** that satisfies  $\phi = R\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0}/(1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.

model is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.

restricted a logical argument stating whether the AR coefficients  $\phi_{m,1}, \dots, \phi_{m,p}$  are restricted to be the same for all regimes.

constraints specifies linear constraints applied to the autoregressive parameters.

**For non-restricted models:** a list of size  $(pxq_m)$  constraint matrices  $C_m$  of full column rank satisfying  $\phi_m = C_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$  and  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models:** a size  $(pxq)$  constraint matrix  $C$  of full column rank satisfying  $\phi = C\psi$ , where  $\phi = (\phi_1, \dots, \phi_p)$  and  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always  $p$  for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

regimeParams a numeric vector specifying the parameter values that should be inserted to the specified regime.

**For non-restricted models: For GMAR model:** Size  $(p+2x1)$  vector  $(\phi_{m,0}, \phi_{m,1}, \dots, \phi_{m,p}, \sigma_m^2)$ .

**For StMAR model:** Size  $(p+3x1)$  vector  $(\phi_{m,0}, \phi_{m,1}, \dots, \phi_{m,p}, \sigma_m^2, \nu_m)$ .

**For G-StMAR model:** Same as GMAR for GMAR-components and same as StMAR for StMAR-components.

**With linear constraints:** Parameter vector as described above, but vector  $\phi_m$  replaced with vector  $\psi_m$  that satisfies  $\phi_m = R_m \psi_m$ .

**For restricted models: For GMAR model:** Size  $(2x1)$  vector  $(\phi_{m,0}, \sigma_m^2)$ .

**For StMAR model:** Size  $(3x1)$  vector  $(\phi_{m,0}, \sigma_m^2, \nu_m)$ .

**For G-StMAR model:** Same as GMAR for GMAR-components and same as StMAR for StMAR-components.

**With linear constraints:** Parameter vector as described above.

regime a positive integer in the interval  $[1, M]$  defining which regime should be changed.

## Value

Returns modified parameter vector of the form described in params.

---

change\_parametrization

*Change parametrization of the parameter vector*


---

### Description

change\_parametrization changes the parametrization of the given parameter vector to change\_to.

### Usage

```
change_parametrization(p, M, params, model = c("GMAR", "StMAR",
"G-StMAR"), restricted = FALSE, constraints = NULL,
change_to = c("intercept", "mean"))
```

### Arguments

**p** a positive integer specifying the order of AR coefficients.

**M** **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.

**For G-StMAR model:** a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is M=M1+M2.

**params** a real valued parameter vector specifying the model.

**For non-restricted models: For GMAR model:** Size  $(M(p+3) - 1x1)$  vector  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .

**For StMAR model:** Size  $(M(p+4) - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(M(p+3) + M2 - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .

**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices **C** that satisfy  $\phi_m = R_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models: For GMAR model:** Size  $(3M + p - 1x1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .

**For StMAR model:** Size  $(4M + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(3M + M2 + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .

**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix **C** that satisfies  $\phi = R\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.

model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	specifies linear constraints applied to the autoregressive parameters.  <b>For non-restricted models:</b> a list of size $(pxq_m)$ constraint matrices $C_m$ of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$ , where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ . <b>For restricted models:</b> a size $(pxq)$ constraint matrix $C$ of full column rank satisfying $\phi = C\psi$ , where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$ . Symbol $\phi$ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always $p$ for all regimes. Ignore or set to NULL if applying linear constraints is <b>not</b> desired.
change_to	either "intercept" or "mean" specifying to which parametrization it should be switched to. If set to "intercept", it's assumed that params is mean-parametrized, and if set to "mean" it's assumed that params is intercept-parametrized.

### Value

Returns parameter vector described in params, but with parametrization changed from intercept to mean (when change\_to==mean) or from mean to intercept (when change\_to==intercept).

### Warning

No argument checks!

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for the G-StMAR model, but it's a straightforward generalization with theoretical properties similar to the GMAR and StMAR models.

---

checkAndCorrectData    *Check the data is set correctly and correct if not*

---

### Description

checkAndCorrectData checks that the data is set correctly and corrects it if not. Throws an error if it can't convert the data to the correct form.

**Usage**

```
checkAndCorrectData(data, p)
```

**Arguments**

`data` a numeric vector class 'ts' object containing the data. NA values are not supported.

`p` a positive integer specifying the order of AR coefficients.

**Value**

Returns a numeric column matrix containing the data.

---

checkConstraintMat      *Check the constraint matrices*

---

**Description**

checkConstraintMat checks for some parts that the constraint matrices are correctly set.

**Usage**

```
checkConstraintMat(p, M, restricted = FALSE, constraints = NULL)
```

**Arguments**

`p` a positive integer specifying the order of AR coefficients.

`M` **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.  
**For G-StMAR model:** a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is M=M1+M2.

`restricted` a logical argument stating whether the AR coefficients  $\phi_{m,1}, \dots, \phi_{m,p}$  are restricted to be the same for all regimes.

`constraints` specifies linear constraints applied to the autoregressive parameters.  
**For non-restricted models:** a list of size  $(pxq_m)$  constraint matrices  $C_m$  of full column rank satisfying  $\phi_m = C_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$  and  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .  
**For restricted models:** a size  $(pxq)$  constraint matrix  $C$  of full column rank satisfying  $\phi = C\psi$ , where  $\phi = (\phi_1, \dots, \phi_p)$  and  $\psi = (\psi_1, \dots, \psi_q)$ .  
 Symbol  $\phi$  denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always  $p$  for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

**Value**

Doesn't return anything, but throws an informative error if finds out that something is wrong.

---

checkPM	<i>Check p and M are correctly set</i>
---------	--

---

**Description**

checkPM checks that the arguments p and M are correctly set.

**Usage**

```
checkPM(p, M, model = c("GMAR", "StMAR", "G-StMAR"))
```

**Arguments**

p	a positive integer specifying the order of AR coefficients.
M	<b>For GMAR and StMAR models:</b> a positive integer specifying the number of mixture components. <b>For G-StMAR model:</b> a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is M=M1+M2.
model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .

**Value**

Doesn't return anything, but throws an informative error if something is wrong.

---

check_data	<i>Check that given object contains data</i>
------------	--

---

**Description**

check\_data checks that that given object contains data.

**Usage**

```
check_data(object)
```

**Arguments**

object	an object to be tested
--------	------------------------

**Value**

Doesn't return anything, but throws and error if something is wrong.

---

check_gsmar	<i>Check that given object has class attribute 'gsmar'</i>
-------------	--

---

**Description**

check\_gsmar checks that that given object has class attribute 'gsmar'.

**Usage**

```
check_gsmar(object)
```

**Arguments**

object            an object to be tested

**Value**

Doesn't return anything, but throws and error if something is wrong.

---

check_model	<i>Check that the argument model is correctly specified.</i>
-------------	--

---

**Description**

check\_model checks that the argument model is correctly specified.

**Usage**

```
check_model(model)
```

**Arguments**

model            is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.

**Value**

Doesn't return anything, but throws and error if something is wrong.

---

check\_params\_length    *Check that the parameter vector has the correct dimension*

---

### Description

check\_model checks that the parameter vector has the correct dimension

### Usage

```
check_params_length(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL)
```

### Arguments

- p** a positive integer specifying the order of AR coefficients.
- M** **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.  
**For G-StMAR model:** a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is M=M1+M2.
- params** a real valued parameter vector specifying the model.  
**For non-restricted models:** **For GMAR model:** Size  $(M(p+3) - 1x1)$  vector  $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .  
**For StMAR model:** Size  $(M(p+4) - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .  
**For G-StMAR model:** Size  $(M(p+3) + M2 - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .  
**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices **C** that satisfy  $\phi_m = R_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .
- For restricted models:** **For GMAR model:** Size  $(3M + p - 1x1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .  
**For StMAR model:** Size  $(4M + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .  
**For G-StMAR model:** Size  $(3M + M2 + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .  
**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix **C** that satisfies  $\phi = R\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.



model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	specifies linear constraints applied to the autoregressive parameters. <b>For non-restricted models:</b> a list of size $(pxq_m)$ constraint matrices $C_m$ of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$ , where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ . <b>For restricted models:</b> a size $(pxq)$ constraint matrix $C$ of full column rank satisfying $\phi = C\psi$ , where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$ . Symbol $\phi$ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always $p$ for all regimes. Ignore or set to NULL if applying linear constraints is <b>not</b> desired.

### Value

Doesn't return anything, but throws an error if something is wrong.

---

condMoments	<i>Calculate conditional moments of GMAR, StMAR or G-StMAR model</i>
-------------	--

---

### Description

condMoments calculates the regime specific conditional means and variances and total conditional means and variances of the specified GMAR, StMAR or G-StMAR model.

### Usage

```
condMoments(data, p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL,
  parametrization = c("intercept", "mean"),
  to_return = c("regime_cmeans", "regime_cvars", "total_cmeans",
    "total_cvars"))
```

### Arguments

data	a numeric vector class 'ts' object containing the data. NA values are not supported.
p	a positive integer specifying the order of AR coefficients.
M	<b>For GMAR and StMAR models:</b> a positive integer specifying the number of mixture components. <b>For G-StMAR model:</b> a size $(2 \times 1)$ vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is $M = M1 + M2$ .

params

a real valued parameter vector specifying the model.

**For non-restricted models: For GMAR model:** Size  $(M(p+3) - 1 \times 1)$  vector  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .

**For StMAR model:** Size  $(M(p+4) - 1 \times 1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(M(p+3) + M2 - 1 \times 1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .

**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices  $C$  that satisfy  $\phi_m = R_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models: For GMAR model:** Size  $(3M + p - 1 \times 1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .

**For StMAR model:** Size  $(4M + p - 1 \times 1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(3M + M2 + p - 1 \times 1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix  $C$  that satisfies  $\phi = R\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.

model

is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.

restricted

a logical argument stating whether the AR coefficients  $\phi_{m,1}, \dots, \phi_{m,p}$  are restricted to be the same for all regimes.

constraints

specifies linear constraints applied to the autoregressive parameters.

**For non-restricted models:** a list of size  $(pxq_m)$  constraint matrices  $C_m$  of full column rank satisfying  $\phi_m = C_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$  and  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models:** a size  $(pxq)$  constraint matrix  $C$  of full column rank satisfying  $\phi = C\psi$ , where  $\phi = (\phi_1, \dots, \phi_p)$  and  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

parametrization

is the model parametrized with the "intercepts"  $\phi_{m,0}$  or "means"  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ ?

to\_return

calculate regimewise conditional means (regime\_cmeans), regimewise conditional variances (regime\_cvars), total conditional means (total\_cmeans), or total conditional variances (total\_cvars)?

**Value**

Note that the first  $p$  observations are taken as the initial values so the conditional moments start from the  $p+1$ :th observation (interpreted as  $t=1$ ).

**if** `to_return=="regime_cmeans"`: a size  $((n_{\text{obs}}-p) \times M)$  matrix containing the regime specific conditional means.

**if** `to_return=="regime_cvars"`: a size  $((n_{\text{obs}}-p) \times M)$  matrix containing the regime specific conditional variances.

**if** `to_return=="total_cmeans"`: a size  $((n_{\text{obs}}-p) \times 1)$  vector containing the total conditional means.

**if** `to_return=="total_cvars"`: a size  $((n_{\text{obs}}-p) \times 1)$  vector containing the total conditional variances.

**References**

- Galbraith, R., Galbraith, J. 1974. On the inverses of some patterned matrices arising in the theory of stationary time series. *Journal of Applied Probability* **11**, 63-71.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for the G-StMAR model, but it's a straightforward generalization with theoretical properties similar to the GMAR and StMAR models.

**See Also**

Other moment functions: [get\\_regime\\_autocovs](#), [get\\_regime\\_means](#), [get\\_regime\\_vars](#), [uncondMoments](#)

**Examples**

```
# GMAR model
params12 <- c(0.18, 0.93, 0.01, 0.86, 0.68, 0.02, 0.88)
rcm12 <- condMoments(logVIX, 1, 2, params12, to_return="regime_cmeans")
rcv12 <- condMoments(logVIX, 1, 2, params12, to_return="regime_cvars")
tcm12 <- condMoments(logVIX, 1, 2, params12, to_return="total_cmeans")
tcv12 <- condMoments(logVIX, 1, 2, params12, to_return="total_cvars")

# StMAR model
params12t <- c(0.17, 0.93, 0.01, 4.87, -0.90, 0.01, 0.98, 4.22, 1000)
rcm12t <- condMoments(logVIX, 1, 2, params12t, model="StMAR",
  to_return="regime_cmeans")
rcv12t <- condMoments(logVIX, 1, 2, params12t, model="StMAR",
  to_return="regime_cvars")

# G-StMAR model
params12gs <- c(0.86, 0.68, 0.02, 0.18, 0.93, 0.01, 0.11, 44)
rcv12gs <- condMoments(logVIX, 1, c(1,1), params12gs, model="G-StMAR",
  to_return="regime_cvars")
tcv12gs <- condMoments(logVIX, 1, c(1,1), params12gs, model="G-StMAR",
  to_return="total_cvars")
```

---

diagnosticPlot	<i>Quantile residual based diagnostic plots for GMAR, StMAR and G-StMAR models</i>
----------------	--

---

### Description

diagnosticPlot plots quantile residual time series, normal QQ-plot, autocorrelation function and squared quantile residual autocorrelation function. There is an option to also plot the individual statistics associated with the quantile residual tests (for autocorrelation and conditional heteroskedasticity) divided by their approximate standard errors with their approximate 95% critical bounds.

### Usage

```
diagnosticPlot(gsmar, nlags = 20, nsimu = 2000,
               plot_indstats = FALSE)
```

### Arguments

gsmar	object of class 'gsmar' created with the function fitGSMAR or GSMAR.
nlags	a positive integer specifying how many lags should be calculated for the autocorrelation and conditional heteroscedasticity statistics.
nsimu	a positive integer specifying to how many simulated values from the process the covariance matrix "Omega" (used to compute the tests) should be based on. Larger number of simulations may result more reliable tests. If smaller than data size, then it will be based on the given data. Ignored if plot_indstats==FALSE.
plot_indstats	set TRUE if the individual statistics discussed in Kalliovirta (2012) should be plotted with their approximate 95% critical bounds (this may take some time).

### Details

Sometimes the individual statistics are not plotted because it's not (numerically) possible for to calculate all the necessary estimates required. This may suggest that the model is misspecified.

The dashed lines plotted with autocorrelation functions (for quantile residuals and their squares) are plus-minus  $1.96 * T^{-1/2}$ .

### Value

diagnosticPlot only plots to a graphical device and doesn't return anything. Use the function quantileResidualTests in order to obtain the individual statistics.

### Suggested packages

Install the suggested package "gsl" for faster evaluations in the cases of StMAR and G-StMAR models. For large StMAR and G-StMAR models with large data the calculations to obtain the individual statistics may take a significantly long time without the package "gsl".

## References

- Galbraith, R., Galbraith, J. 1974. On the inverses of some patterned matrices arising in the theory of stationary time series. *Journal of Applied Probability* **11**, 63-71.
- Kalliovirta L. (2012) Misspecification tests based on quantile residuals. *The Econometrics Journal*, **15**, 358-393.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for G-StMAR model, but it's a straight forward generalization with theoretical properties similar to GMAR and StMAR models.

## See Also

[fitGSMAR](#), [GSMAR](#), [quantileResidualTests](#), [quantileResidualPlot](#), [simulateGSMAR](#)

## Examples

```
# GMAR model
fit12 <- fitGSMAR(data=logVIX, p=1, M=2, model="GMAR")
diagnosticPlot(fit12)

# Restricted GMAR model: plot also the individual statistics with
# their approximate critical bounds using the given data
fit12r <- fitGSMAR(logVIX, 1, 2, model="GMAR", restricted=TRUE)
diagnosticPlot(fit12r, nlags=10, nsimu=1, plot_indstats=TRUE)

# Non-mixture version of StMAR model
fit11t <- fitGSMAR(logVIX, 1, 1, model="StMAR", ncores=1, ncalls=1)
diagnosticPlot(fit11t)

# G-StMAR model
fit12gs <- fitGSMAR(logVIX, 1, M=c(1, 1), model="G-StMAR")
diagnosticPlot(fit12gs)

# Restricted G-StMAR-model
fit12gsr <- fitGSMAR(logVIX, 1, M=c(1, 1), model="G-StMAR",
  restricted=TRUE)
diagnosticPlot(fit12gsr)

# GMAR model as a mixture of AR(2) and AR(1) models
constraints <- list(diag(1, ncol=2, nrow=2), as.matrix(c(1, 0)))
fit22c <- fitGSMAR(logVIX, 2, 2, constraints=constraints)
diagnosticPlot(fit22c)

# Such StMAR(3,2) that the AR coefficients are restricted to be
# the same for both regimes and that the second AR coefficients are
# constrained to zero.
fit32rc <- fitGSMAR(logVIX, 3, 2, model="StMAR", restricted=TRUE,
```

```
constraints=matrix(c(1, 0, 0, 0, 0, 1), ncol=2))
diagnosticPlot(fit32rc)
```

---

extractRegime	<i>Extract regime from a parameter vector</i>
---------------	---

---

### Description

extractRegime extracts the specified regime from the GMAR, StMAR or G-StMAR model's parameter vector. Doesn't extract mixing weight parameter alpha.

### Usage

```
extractRegime(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL, regime)
```

### Arguments

- p** a positive integer specifying the order of AR coefficients.
- M** **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.  
**For G-StMAR model:** a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is M=M1+M2.
- params** a real valued parameter vector specifying the model.  
**For non-restricted models: For GMAR model:** Size  $(M(p+3) - 1x1)$  vector  $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .  
**For StMAR model:** Size  $(M(p+4) - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .  
**For G-StMAR model:** Size  $(M(p+3) + M2 - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .  
**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices **C** that satisfy  $\phi_m = R_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .
- For restricted models: For GMAR model:** Size  $(3M + p - 1x1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .  
**For StMAR model:** Size  $(4M + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .  
**For G-StMAR model:** Size  $(3M + M2 + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .  
**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix **C** that satisfies  $\phi = R\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest

	M2 components are <i>StMAR-type</i> . Note that in the case <b>M=1</b> the parameter $\alpha$ is dropped, and in the case of <b>StMAR</b> or <b>G-StMAR</b> model the degrees of freedom parameters $\nu_m$ have to be larger than 2.
model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	specifies linear constraints applied to the autoregressive parameters. <b>For non-restricted models:</b> a list of size $(pxq_m)$ constraint matrices $C_m$ of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$ , where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ . <b>For restricted models:</b> a size $(pxq)$ constraint matrix $C$ of full column rank satisfying $\phi = C\psi$ , where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$ . Symbol $\phi$ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always $p$ for all regimes. Ignore or set to NULL if applying linear constraints is <b>not</b> desired.
regime	a positive integer in the interval $[1, M]$ defining which regime should be extracted.

### Value

Returns a numeric vector corresponding to the regime with...

**For non-restricted models: For GMAR model:** Size  $(p+2x1)$  vector  $(\phi_{m,0}, \phi_{m,1}, \dots, \phi_{m,p}, \sigma_m^2)$ .

**For StMAR model:** Size  $(p+3x1)$  vector  $(\phi_{m,0}, \phi_{m,1}, \dots, \phi_{m,p}, \sigma_m^2, \nu_m)$ .

**For G-StMAR model:** Same as GMAR for GMAR-components and same as StMAR for StMAR-components.

**With linear constraints:** Parameter vector as described above, but vector  $\phi_m$  replaced with vector  $\psi_m$  that satisfies  $\phi_m = R_m \psi_m$ .

**For restricted models: For GMAR model:** Size  $(2x1)$  vector  $(\phi_{m,0}, \sigma_m^2)$ .

**For StMAR model:** Size  $(3x1)$  vector  $(\phi_{m,0}, \sigma_m^2, \nu_m)$ .

**For G-StMAR model:** Same as GMAR for GMAR-components and same as StMAR for StMAR-components.

**With linear constraints:** Parameter vector as described above.

---

fitGSMAR

---

*Estimate Gaussian or Student's t Mixture Autoregressive model*


---

### Description

fitGSMAR estimates GMAR, StMAR or G-StMAR model in two phases: in the first phase, the genetic algorithm is employed to find starting values for the gradient based variable metric algorithm (also known as quasi-Newton method). In the second phase, the variable metric algorithm accurately converges to a nearby local maximum or saddle point. Parallel computing is used to perform multiple rounds of estimations in parallel.

**Usage**

```
fitGSMAR(data, p, M, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL, conditional = TRUE,
  parametrization = c("intercept", "mean"), ncalls = round(10 + 9 *
  log(sum(M))), ncores = min(2, ncalls, parallel::detectCores()),
  maxit = 300, seeds = NULL, printRes = TRUE, runTests = FALSE,
  ...)
```

**Arguments**

data	a numeric vector class 'ts' object containing the data. NA values are not supported.
p	a positive integer specifying the order of AR coefficients.
M	<b>For GMAR and StMAR models:</b> a positive integer specifying the number of mixture components. <b>For G-StMAR model:</b> a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is M=M1+M2.
model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	specifies linear constraints applied to the autoregressive parameters. <b>For non-restricted models:</b> a list of size $(pxq_m)$ constraint matrices $C_m$ of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$ , where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ . <b>For restricted models:</b> a size $(pxq)$ constraint matrix $C$ of full column rank satisfying $\phi = C\psi$ , where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$ . Symbol $\phi$ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is <b>not</b> desired.
conditional	a logical argument specifying whether the conditional or exact log-likelihood function should be used.
parametrization	is the model parametrized with the "intercepts" $\phi_{m,0}$ or "means" $\mu_m = \phi_{m,0}/(1 - \sum \phi_{i,m})$ ?
ncalls	a positive integer specifying how many rounds of estimation should be conducted. The estimation results may vary from round to round because of multimodality of the log-likelihood function and randomness associated with the genetic algorithm.
ncores	the number of cores to be used in the estimation process.
maxit	maximum number of iterations in the variable metric algorithm.



seeds	a length ncalls vector containing the random number generator seed for each call to the genetic algorithm, or NULL for not initializing the seed. Exists for creating reproducible results.
printRes	should the estimation results be printed?
runTests	should quantile residuals tests be performed after the estimation?
...	additional settings passed to the function GAFit employing the genetic algorithm.

## Details

Because of complexity and multimodality of the log-likelihood function, it's **not guaranteed** that the estimation algorithm will end up in the global maximum point. It's often expected that most of the estimation rounds will end up in some local maximum point instead, and therefore a number of estimation rounds is required for reliable results. Because of the nature of the models, the estimation may fail particularly in the cases where the number of mixture components is chosen too large.

If the iteration limit in the variable metric algorithm (maxit) is reached, one can continue the estimation by iterating more with the function `iterate_more`.

The genetic algorithm is mostly based on the description by *Dorsey and Mayer (1995)*. It uses (slightly modified) individually adaptive crossover and mutation rates described by *Patnaik and Srinivas (1994)* and employs (50%) fitness inheritance discussed by *Smith, Dike and Stegmann (1995)*. Large (in absolute value) but stationary AR parameter values are generated with the algorithm proposed by *Monahan (1984)*.

The variable metric algorithm (or quasi-Newton method, *Nash (1990, algorithm 21)*) used in the second phase is implemented with function `optim` from the package `stats`.

Some mixture components of the StMAR model may sometimes get very large estimates for degrees of freedom parameters. Such estimates may, for example, cause computing the quantile residual tests to fail. However, such mixture components are very much similar to the components of the GMAR model. It's hence advisable to further estimate a G-StMAR model by allowing the mixture components with large degrees of freedom parameter estimates to be GMAR type.

## Value

Returns an object of class 'gsmar' defining the estimated GMAR, StMAR or G-StMAR model. The returned object contains empirical mixing weights, conditional means and variances, quantile residuals, and quantile residual test results if the tests were performed. Note that the first  $p$  observations are taken as the initial values so mixing weights, conditional moments and qresiduals start from the  $p+1$ :th observation (interpreted as  $t=1$ ). In addition, the returned object contains the estimates and log-likelihood values from all the estimation rounds. The estimated parameter vector can be obtained at `gsmar$params` (and the corresponding approximate standard errors at `gsmar$std_errors`) and it's...

**For non-restricted models: For GMAR model:** Size  $(M(p+3) - 1 \times 1)$  vector  $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .

**For StMAR model:** Size  $(M(p+4) - 1 \times 1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(M(p+3) + M - 1 \times 1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M+1}, \dots, \nu_M)$ .

**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices  $C$  that satisfy  $\phi_m = C_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models: For GMAR model:** Size  $(3M+p-1 \times 1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .

**For StMAR model:** Size  $(4M+p-1 \times 1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(3M+M2+p-1 \times 1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_{M1+M2})$ .

**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix  $C$  that satisfies  $\phi = C\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first  $M1$  components are *GMAR-type* and the rest  $M2$  components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.

### S3 methods

The following S3 methods are supported for class 'gsmar' objects: print, summary, plot, logLik, residuals.

### Suggested packages

For faster evaluation of the quantile residuals of StMAR and G-StMAR models, install the suggested package "gsl". Note that for large StMAR and G-StMAR models with large data the evaluations of the quantile residual tests may take significantly long time without the package "gsl".

### References

- Dorsey R. E. and Mayer W. J. 1995. Genetic algorithms for estimation problems with multiple optima, nondifferentiability, and other irregular features. *Journal of Business & Economic Statistics*, **13**, 53-66.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [econ.EM].
- Monahan J.F. 1984. A Note on Enforcing Stationarity in Autoregressive-Moving Average Models. *Biometrika* **71**, 403-404.
- Nash J. 1990. Compact Numerical Methods for Computers. Linear algebra and Function Minimization. *Adam Hilger*.
- Patnaik L.M. and Srinivas M. 1994. Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms. *Transactions on Systems, Man and Cybernetics* **24**, 656-667.
- Smith R.E., Dike B.A., Stegmann S.A. 1995. Fitness inheritance in genetic algorithms. *Proceedings of the 1995 ACM Symposium on Applied Computing*, 345-350.
- There are currently no published references for the G-StMAR model, but it's a straightforward generalization with theoretical properties similar to the GMAR and StMAR models.

**See Also**

[GSMAR](#), [iterate\\_more](#), [stmar\\_to\\_gstmar](#), [add\\_data](#), [swap\\_parametrization](#), [get\\_gradient](#), [simulateGSMAR](#), [predict.gsmar](#), [diagnosticPlot](#), [quantileResidualTests](#), [condMoments](#), [uncondMoments](#)

**Examples**

```
# These are long running examples that use parallel computing

# GMAR model
fit12 <- fitGSMAR(data=logVIX, p=1, M=2, model="GMAR")
fit12
summary(fit12)
plot(fit12)

# Restricted GMAR model
fit12r <- fitGSMAR(logVIX, 1, 2, model="GMAR", restricted=TRUE,
  parametrization="mean", ncalls=10)
fit12r
summary(fit12r)

# Non-mixture version of StMAR model
fit11t <- fitGSMAR(logVIX, 1, 1, model="StMAR", ncores=1, ncalls=1)
fit11t

# StMAR model, seeds for reproducibility
fit12t <- fitGSMAR(logVIX, 1, 2, model="StMAR", ncalls=10,
  seeds=1:10)
fit12t

# G-StMAR model with one GMAR type and one StMAR type regime
fit12gs <- fitGSMAR(logVIX, 1, M=c(1, 1), model="G-StMAR")
fit12gs

# Restricted G-StMAR model
fit12gsr <- fitGSMAR(logVIX, 1, c(1, 1), model="G-StMAR", restricted=TRUE)
fit12gsr

# The following three examples demonstrate how to apply linear constraints
# to the AR parameters.

# GMAR(p=3, M=2) model that the second and third AR coefficients are constrained
# to zero in the first regime, and the second AR coefficient is constrained to
# zero in the second regime.
constraints <- list(matrix(c(1, 0, 0, 0, 0, 1), ncol=2), as.matrix(c(1, 0, 0)))
fit32c <- fitGSMAR(logVIX, 3, 2, constraints=constraints)
fit32c

# Such constrained StMAR(3, 1) model that the second order AR coefficient
# is constrained to zero.
constraints <- list(matrix(c(1, 0, 0, 0, 0, 1), ncol=2))
fit31tc <- fitGSMAR(logVIX, 3, 1, model="StMAR", constraints=constraints)
```

```

fit31tc

# Such StMAR(3, 2) model that the AR coefficients are restricted to be
# the same for both regimes and that the second AR coefficients are
# constrained to zero.
fit32trc <- fitGSMAR(logVIX, 3, 2, model="StMAR", restricted=TRUE,
                    constraints=matrix(c(1, 0, 0, 0, 0, 1), ncol=2))
fit32trc

```

---

format_valuef	<i>Function factory for value formatting</i>
---------------	--

---

### Description

format\_valuef generates functions that format values so that they print out with the desired number of digits.

### Usage

```
format_valuef(digits)
```

### Arguments

digits            number of digits to use

### Value

returns a function that takes an atomic vector as argument and returns it formatted to character with digits decimals.

---

GAfit	<i>Genetic algorithm for preliminary estimation of GMAR, StMAR or G-StMAR model</i>
-------	---

---

### Description

GAfit estimates specified GMAR, StMAR or G-StMAR model using genetic algorithm. It's designed to find starting values for gradient based methods.

### Usage

```

GAfit(data, p, M, model = c("GMAR", "StMAR", "G-StMAR"),
      restricted = FALSE, constraints = NULL,
      parametrization = c("intercept", "mean"), conditional = TRUE,
      ngen = 200, popsize, smartMu, meanscale, sigmascale, initpop = NULL,
      regime_force_scale = 1, red_criteria = c(0.05, 0.01),
      to_return = c("alt_ind", "best_ind"), minval, seed = NULL, ...)

```

**Arguments**

data	a numeric vector class 'ts' object containing the data. NA values are not supported.
p	a positive integer specifying the order of AR coefficients.
M	<b>For GMAR and StMAR models:</b> a positive integer specifying the number of mixture components. <b>For G-StMAR model:</b> a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is M=M1+M2.
model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	specifies linear constraints applied to the autoregressive parameters. <b>For non-restricted models:</b> a list of size $(pxq_m)$ constraint matrices $C_m$ of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$ , where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ . <b>For restricted models:</b> a size $(pxq)$ constraint matrix $C$ of full column rank satisfying $\phi = C\psi$ , where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$ . Symbol $\phi$ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is <b>not</b> desired.
parametrization	is the model parametrized with the "intercepts" $\phi_{m,0}$ or "means" $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ ?
conditional	a logical argument specifying whether the conditional or exact log-likelihood function should be used.
ngen	a positive integer specifying the number of generations to be ran through in the genetic algorithm.
popsize	a positive even integer specifying the population size in the genetic algorithm. Default is $10 \times d$ where d is the number of parameters.
smartMu	a positive integer specifying the generation after which the random mutations in the genetic algorithm are "smart". This means that mutating individuals will mostly mutate fairly close (or partially close) to the best fitting individual so far. Default is $\min(100, \text{round}(0.5 \times \text{ngen}))$ .
meanscale	a real valued vector of length two specifying the mean (the first element) and standard deviation (the second element) of the normal distribution from which the $\mu_m$ mean-parameters are generated in random mutations in the genetic algorithm. Default is $c(\text{mean}(\text{data}), \text{sd}(\text{data}))$ . Note that the genetic algorithm optimizes with mean-parametrization even when <code>parametrization="intercept"</code> , but input (in <code>initpop</code> ) and output (return value) parameter vectors may be intercept-parametrized.

sigmascale	<p>a positive real number specifying the standard deviation of the (zero mean, positive only) normal distribution from which the component variance parameters are generated in the random mutations in the genetic algorithm. Default is <code>var(stats::ar(data, order.max=10)\$resid, na.rm=TRUE)</code>.</p>
initpop	<p>a list of parameter vectors from which the initial population of the genetic algorithm will be generated from. The parameter vectors should be of form...</p> <p><b>For non-restricted models: For GMAR model:</b> Size <math>(M(p+3) - 1 \times 1)</math> vector <math>\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})</math>, where <math>v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)</math> and <math>\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})</math>, <math>m = 1, \dots, M</math>.</p> <p><b>For StMAR model:</b> Size <math>(M(p+4) - 1 \times 1)</math> vector <math>(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)</math>.</p> <p><b>For G-StMAR model:</b> Size <math>(M(p+3) + M2 - 1 \times 1)</math> vector <math>(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M+1}, \dots, \nu_M)</math>.</p> <p><b>With linear constraints:</b> Replace the vectors <math>\phi_m</math> with vectors <math>\psi_m</math> and provide a list of constraint matrices <math>\mathbf{C}</math> that satisfy <math>\phi_m = \mathbf{C}_m \psi_m</math> for all <math>m = 1, \dots, M</math>, where <math>\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})</math>.</p> <p><b>For restricted models: For GMAR model:</b> Size <math>(3M + p - 1 \times 1)</math> vector <math>\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})</math>, where <math>\phi = (\phi_1, \dots, \phi_M)</math>.</p> <p><b>For StMAR model:</b> Size <math>(4M + p - 1 \times 1)</math> vector <math>(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)</math>.</p> <p><b>For G-StMAR model:</b> Size <math>(3M + M2 + p - 1 \times 1)</math> vector <math>(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M+1}, \dots, \nu_M)</math>.</p> <p><b>With linear constraints:</b> Replace the vector <math>\phi</math> with vector <math>\psi</math> and provide a constraint matrix <math>\mathbf{C}</math> that satisfies <math>\phi = \mathbf{C}\psi</math>, where <math>\psi = (\psi_1, \dots, \psi_q)</math>.</p> <p>Symbol <math>\phi</math> denotes an AR coefficient, <math>\sigma^2</math> a variance, <math>\alpha</math> a mixing weight and <math>\nu</math> a degrees of freedom parameter. Note that in the case <math>\mathbf{M}=\mathbf{1}</math> the parameter <math>\alpha</math> is dropped, and in the case of <b>StMAR</b> or <b>G-StMAR</b> model the degrees of freedom parameters <math>\nu_m</math> have to be larger than 2. If not specified (or FALSE as is default), the initial population will be drawn randomly.</p>
regime_force_scale	<p>a non-negative real number specifying how much should natural selection favour individuals with less regimes that have almost all mixing weights (practically) at zero (see <code>red_criteria</code>), i.e. less "redundant regimes". Set to zero for no favouring or large number for heavy favouring. Without any favouring the genetic algorithm gets more often stuck in an area of the parameter space where some regimes are wasted, but with too much favouring the best genes might never mix into the population and the algorithm might converge poorly. Default is 1 and it gives <math>2x</math> larger surviving probabilities for individuals with no wasted regimes compared to individuals with one wasted regime. Number 2 would give <math>3x</math> larger probabilities etc.</p>
red_criteria	<p>a length 2 numeric vector specifying the criteria that is used to determine whether a regime is redundant or not. Any regime <math>m</math> which satisfies <code>sum(mixingWeights[,m] &gt; red_criteria[1]) &lt; red_criteria[2]*n_obs</code> will be considered "redundant". One should be careful when adjusting this argument.</p>
to_return	<p>should the genetic algorithm return the best fitting individual which has as much "non-redundant" regimes as possible ("<code>alt_ind</code>") or the individual which has the highest log-likelihood in general ("<code>best_ind</code>"), but might have more wasted regimes?</p>

minval	a real number defining the minimum value of the log-likelihood function that will be considered. Values smaller than this will be treated as they were minval and the corresponding individuals will never survive. The default is $-(10^{\lceil \log_{10}(\text{length}(\text{data}) + 1) \rceil} - 1)$ , and one should be very careful if adjusting this.
seed	a single value, interpreted as an integer, or NULL, that sets seed for the random number generator in the beginning of the function call. If calling GAfit from fitGSMAR, use the argument seeds instead of passing the argument seed.
...	Currently not in use. Exists for a technical reason only.

### Details

The genetic algorithm is mostly based on the description by *Dorsey and Mayer (1995)*. It uses (slightly modified) individually adaptive crossover and mutation rates described by *Patnaik and Srinivas (1994)* and employs (50%) fitness inheritance discussed by *Smith, Dike and Stegmann (1995)*. Large (in absolute value) but stationary AR parameter values are generated with the algorithm proposed by *Monahan (1984)*.

By "redundant" or "unidentified" regimes we mean regimes that have the time varying mixing weights basically at zero for all  $t$ . The model would have about the same log-likelihood value without redundant regimes and there is no purpose to have redundant regime in the model.

### Value

Returns estimated parameter vector described in `initpop`.

### References

- Dorsey R. E. and Mayer W. J. 1995. Genetic algorithms for estimation problems with multiple optima, nondifferentiability, and other irregular features. *Journal of Business & Economic Statistics*, **13**, 53-66.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [econ.EM].
- Monahan J.F. 1984. A Note on Enforcing Stationarity in Autoregressive-Moving Average Models. *Biometrika* **71**, 403-404.
- Patnaik L.M. and Srinivas M. 1994. Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms. *Transactions on Systems, Man and Cybernetics* **24**, 656-667.
- Smith R.E., Dike B.A., Stegmann S.A. 1995. Fitness inheritance in genetic algorithms. *Proceedings of the 1995 ACM Symposium on Applied Computing*, 345-350.
- There are currently no published references for G-StMAR model, but it's a straightforward generalization with theoretical properties similar to GMAR and StMAR models.

---

getOmega	Generate covariance matrix Omega for quantile residual tests
----------	--

---

### Description

getOmega generates the covariance matrix Omega used in the quantile residual tests.

### Usage

```
getOmega(data, p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL,
  parametrization = c("intercept", "mean"), g, dim_g)
```

### Arguments

data	a numeric vector class 'ts' object containing the data. NA values are not supported.
p	a positive integer specifying the order of AR coefficients.
M	<p><b>For GMAR and StMAR models:</b> a positive integer specifying the number of mixture components.</p> <p><b>For G-StMAR model:</b> a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is M=M1+M2.</p>
params	<p>a real valued parameter vector specifying the model.</p> <p><b>For non-restricted models:</b></p> <p><b>For GMAR model:</b> Size <math>(M(p+3) - 1x1)</math> vector <math>\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})</math>, where <math>v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)</math> and <math>\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})</math>, <math>m = 1, \dots, M</math>.</p> <p><b>For StMAR model:</b> Size <math>(M(p+4) - 1x1)</math> vector <math>(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)</math>.</p> <p><b>For G-StMAR model:</b> Size <math>(M(p+3) + M2 - 1x1)</math> vector <math>(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)</math>.</p> <p><b>With linear constraints:</b> Replace the vectors <math>\phi_m</math> with vectors <math>\psi_m</math> and provide a list of constraint matrices <b>C</b> that satisfy <math>\phi_m = R_m \psi_m</math> for all <math>m = 1, \dots, M</math>, where <math>\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})</math>.</p> <p><b>For restricted models:</b></p> <p><b>For GMAR model:</b> Size <math>(3M + p - 1x1)</math> vector <math>\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})</math>, where <math>\phi = (\phi_1, \dots, \phi_M)</math>.</p> <p><b>For StMAR model:</b> Size <math>(4M + p - 1x1)</math> vector <math>(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)</math>.</p> <p><b>For G-StMAR model:</b> Size <math>(3M + M2 + p - 1x1)</math> vector <math>(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)</math>.</p> <p><b>With linear constraints:</b> Replace the vector <math>\phi</math> with vector <math>\psi</math> and provide a constraint matrix <b>C</b> that satisfies <math>\phi = R\psi</math>, where <math>\psi = (\psi_1, \dots, \psi_q)</math>.</p>

Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.



model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	specifies linear constraints applied to the autoregressive parameters. <b>For non-restricted models:</b> a list of size $(pxq_m)$ constraint matrices $C_m$ of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$ , where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ . <b>For restricted models:</b> a size $(pxq)$ constraint matrix $C$ of full column rank satisfying $\phi = C\psi$ , where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$ . Symbol $\phi$ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always $p$ for all regimes. Ignore or set to NULL if applying linear constraints is <b>not</b> desired.
parametrization	is the model parametrized with the "intercepts" $\phi_{m,0}$ or "means" $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ ?
g	a function specifying the transformation.
dim_g	output dimension of the transformation g.

### Details

This function is used for quantile residuals tests in `quantileResidualTests`.

### Value

Returns size  $(\text{dim}_g \times \text{dim}_g)$  covariance matrix Omega.

### References

- Galbraith, R., Galbraith, J. 1974. On the inverses of some patterned matrices arising in the theory of stationary time series. *Journal of Applied Probability* **11**, 63-71.
- Kalliovirta L. (2012) Misspecification tests based on quantile residuals. *The Econometrics Journal*, **15**, 358-393.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [econ.EM].
- There are currently no published references for G-StMAR model, but it's a straight forward generalization with theoretical properties similar to GMAR and StMAR models.

### See Also

[quantileResidualTests](#)

---

get_ar_roots	<i>Calculate absolute values of the roots of the AR characteristic polynomials</i>
--------------	--

---

### Description

get\_ar\_roots calculates absolute values of the roots of the AR characteristic polynomials for each component.

### Usage

```
get_ar_roots(gsmar)
```

### Arguments

gsmar            object of class 'gsmar' created with the function fitGSMAR or GSMAR.

### Value

Returns a list with M elements each containing the absolute values of the roots of the AR characteristic polynomial corresponding to each mixture component.

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for the G-StMAR model, but it's a straightforward generalization with theoretical properties similar to the GMAR and StMAR models.

### Examples

```
params13 <- c(1.4, 0.88, 0.26, 2.46, 0.82, 0.74, 5.0, 0.68, 5.2, 0.72, 0.2)
gmar13 <- GSMAR(data=VIX, p=1, M=3, params=params13, model="GMAR")
get_ar_roots(gmar13)
```

---

get_IC	<i>Calculate AIC, HQIC and BIC</i>
--------	------------------------------------

---

**Description**

get\_IC calculates AIC, HQIC and BIC

**Usage**

```
get_IC(loglik, npars, obs)
```

**Arguments**

loglik	log-likelihood value
npars	number of (freely estimated) parameters in the model.
obs	numbers of observations with starting values excluded for conditional models.

**Value**

Returns a data frame containing the information criteria values.

---

get_regime_autocovs	<i>Calculate regime specific autocovariances <math>\gamma_{m,p}</math></i>
---------------------	--

---

**Description**

get\_regime\_autocovs calculates the first  $p$  regime specific autocovariances  $\gamma_{m,p}$  of the given GMAR, StMAR or G-StMAR model.

**Usage**

```
get_regime_autocovs(gsmar)
```

**Arguments**

gsmar	object of class 'gsmar' created with the function fitGSMAR or GSMAR.
-------	--

**Value**

Returns a size  $(pxM)$  matrix containing the first  $p$  autocovariates of the components processes:  $i$ :th autocovariance in the  $i$ :th row and  $m$ :th component process in the  $m$ :th column.

## References

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [econ.EM].
- There are currently no published references for the G-StMAR model, but it's a straightforward generalization with theoretical properties similar to the GMAR and StMAR models.
- Lutkepohl H. 2005. New Introduction to Multiple Time Series Analysis. *Springer*.

## See Also

Other moment functions: [condMoments](#), [get\\_regime\\_means](#), [get\\_regime\\_vars](#), [uncondMoments](#)

## Examples

```
# GMAR model
params13 <- c(1.4, 0.88, 0.26, 2.46, 0.82, 0.74, 5.0, 0.68, 5.2, 0.72, 0.2)
gmar13 <- GSMAR(p=1, M=3, params=params13, model="GMAR")
get_regime_autocovs(gmar13)

# StMAR model
params12t <- c(1.38, 0.88, 0.27, 3.8, 0.74, 3.15, 0.8, 100, 3.6)
stmar12t <- GSMAR(p=1, M=2, params=params12t, model="StMAR")
get_regime_autocovs(stmar12t)

# G-StMAR model (similar to the StMAR model above)
params12gs <- c(1.38, 0.88, 0.27, 3.8, 0.74, 3.15, 0.8, 3.6)
gstmar12 <- GSMAR(p=1, M=c(1, 1), params=params12gs, model="G-StMAR")
get_regime_autocovs(gstmar12)
```

---

<code>get_regime_means</code>	<i>Calculate regime specific means <math>\mu_m</math></i>
-------------------------------	---

---

## Description

`get_regime_means` calculates regime means  $\mu_m = \phi_{m,0}/(1 - \sum \phi_{i,m})$  of the given GMAR, StMAR or G-StMAR model

## Usage

```
get_regime_means(gsmar)
```

## Arguments

`gsmar` object of class 'gsmar' created with the function `fitGSMAR` or `GSMAR`.

**Value**

Returns a length M vector containing regime mean  $\mu_m$  in the m:th column,  $m = 1, \dots, M$ .

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for the G-StMAR model, but it's a straightforward generalization with theoretical properties similar to the GMAR and StMAR models.

**See Also**

[condMoments](#), [uncondMoments](#), [get\\_regime\\_vars](#), [get\\_regime\\_autocovs](#)

Other moment functions: [condMoments](#), [get\\_regime\\_autocovs](#), [get\\_regime\\_vars](#), [uncondMoments](#)

**Examples**

```
# GMAR model
params13 <- c(1.4, 0.88, 0.26, 2.46, 0.82, 0.74, 5.0, 0.68, 5.2, 0.72, 0.2)
gmar13 <- GSMAR(p=1, M=3, params=params13, model="GMAR")
get_regime_means(gmar13)

# StMAR model
params12t <- c(1.38, 0.88, 0.27, 3.8, 0.74, 3.15, 0.8, 100, 3.6)
stmar12t <- GSMAR(p=1, M=2, params=params12t, model="StMAR")
get_regime_means(stmar12t)

# G-StMAR model (similar to the StMAR model above)
params12gs <- c(1.38, 0.88, 0.27, 3.8, 0.74, 3.15, 0.8, 3.6)
gstmar12 <- GSMAR(p=1, M=c(1, 1), params=params12gs, model="G-StMAR")
get_regime_means(gstmar12)
```

---

<code>get_regime_vars</code>	<i>Calculate regime specific variances <math>\gamma_{m,0}</math></i>
------------------------------	--

---

**Description**

`get_regime_vars` calculates the unconditional regime specific variances  $\gamma_{m,0}$  of the given GMAR, StMAR or G-StMAR model.

**Usage**

```
get_regime_vars(gsmar)
```

**Arguments**

gsmar                    object of class 'gsmar' created with the function `fitGSMAR` or `GSMAR`.

**Value**

Returns a length `M` vector containing the unconditional variances of the components processes: `m:th` slot for `m:th` regime.

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for the G-StMAR model, but it's a straightforward generalization with theoretical properties similar to the GMAR and StMAR models.
- Lutkepohl H. 2005. New Introduction to Multiple Time Series Analysis. *Springer*.

**See Also**

Other moment functions: [condMoments](#), [get\\_regime\\_autocovs](#), [get\\_regime\\_means](#), [uncondMoments](#)

**Examples**

```
# GMAR model
params13 <- c(1.4, 0.88, 0.26, 2.46, 0.82, 0.74, 5.0, 0.68, 5.2, 0.72, 0.2)
gmar13 <- GSMAR(p=1, M=3, params=params13, model="GMAR")
get_regime_vars(gmar13)

# StMAR model
params12t <- c(1.38, 0.88, 0.27, 3.8, 0.74, 3.15, 0.8, 100, 3.6)
stmar12t <- GSMAR(p=1, M=2, params=params12t, model="StMAR")
get_regime_vars(stmar12t)

# G-StMAR model (similar to the StMAR model above)
params12gs <- c(1.38, 0.88, 0.27, 3.8, 0.74, 3.15, 0.8, 3.6)
gstmar12 <- GSMAR(p=1, M=c(1, 1), params=params12gs, model="G-StMAR")
get_regime_vars(gstmar12)
```

---

get\_varying\_h

*Get differences 'h' which are adjusted for overly large degrees of freedom parameters*

---

**Description**

`get_varying_h` adjusts differences for overly large degrees of freedom parameters for finite difference approximation of the derivatives of the log-likelihood function of a StMAR or G-StMAR model.

**Usage**

```
get_varying_h(p, M, params, model)
```

**Arguments**

- p** a positive integer specifying the order of AR coefficients.
- M** **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.  
**For G-StMAR model:** a size  $(2 \times 1)$  vector specifying the number of *GMAR-type* components  $M1$  in the first element and *StMAR-type* components  $M2$  in the second. The total number of mixture components is  $M=M1+M2$ .
- params** a real valued parameter vector specifying the model.  
**For non-restricted models: For GMAR model:** Size  $(M(p+3) - 1 \times 1)$  vector  $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .  
**For StMAR model:** Size  $(M(p+4) - 1 \times 1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .  
**For G-StMAR model:** Size  $(M(p+3) + M2 - 1 \times 1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .  
**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices  $C$  that satisfy  $\phi_m = R_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .  
**For restricted models: For GMAR model:** Size  $(3M + p - 1 \times 1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .  
**For StMAR model:** Size  $(4M + p - 1 \times 1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .  
**For G-StMAR model:** Size  $(3M + M2 + p - 1 \times 1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .  
**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix  $C$  that satisfies  $\phi = R\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .
- Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If `parametrization=="mean"` just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first  $M1$  components are *GMAR-type* and the rest  $M2$  components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.
- model** is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first  $M1$  components are *GMAR-type* and the rest  $M2$  components are *StMAR-type*.

**Details**

This function is used for approximating gradient and Hessian of a StMAR or G-StMAR model. Large degrees of freedom parameters cause significant numerical error if too small differences are used.

**Value**

Returns a vector with the same length as `params`. For other than the degrees of freedom parameters larger than 100 the values will be  $6e-6$ , and for the large degrees of freedom parameters the values will be `signif(df/1000,digits=2)`.

---

GSMAR	<i>Create object of class 'gsmar' defining a GMAR, StMAR or G-StMAR model</i>
-------	---

---

**Description**

GSMAR creates an S3 object of class 'gsmar' that defines a GMAR, StMAR or G-StMAR model.

**Usage**

```
GSMAR(data, p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
       restricted = FALSE, constraints = NULL, conditional = TRUE,
       parametrization = c("intercept", "mean"), calc_qresiduals,
       calc_cond_moments, calc_std_errors = FALSE, custom_h = NULL)
```

```
## S3 method for class 'gsmar'
logLik(object, ...)
```

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

```
## S3 method for class 'gsmar'
summary(object, ..., digits = 2)
```

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

```
## S3 method for class 'gsmar'
print(x, ..., digits = 2, summary_print = FALSE)
```

**Arguments**

<code>data</code>	a numeric vector class 'ts' object containing the data. NA values are not supported.
<code>p</code>	a positive integer specifying the order of AR coefficients.
<code>M</code>	<b>For GMAR and StMAR models:</b> a positive integer specifying the number of mixture components. <b>For G-StMAR model:</b> a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is $M=M1+M2$ .
<code>params</code>	a real valued parameter vector specifying the model.



**For non-restricted models: For GMAR model:** Size  $(M(p+3) - 1x1)$  vector  $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .

**For StMAR model:** Size  $(M(p+4) - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(M(p+3) + M2 - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .

**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices  $\mathbf{C}$  that satisfy  $\phi_m = R_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models: For GMAR model:** Size  $(3M + p - 1x1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .

**For StMAR model:** Size  $(4M + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(3M + M2 + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .

**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix  $C$  that satisfies  $\phi = R\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.

model is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.

restricted a logical argument stating whether the AR coefficients  $\phi_{m,1}, \dots, \phi_{m,p}$  are restricted to be the same for all regimes.

constraints specifies linear constraints applied to the autoregressive parameters.

**For non-restricted models:** a list of size  $(pxq_m)$  constraint matrices  $C_m$  of full column rank satisfying  $\phi_m = C_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$  and  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models:** a size  $(pxq)$  constraint matrix  $C$  of full column rank satisfying  $\phi = C\psi$ , where  $\phi = (\phi_1, \dots, \phi_p)$  and  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always  $p$  for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

conditional a logical argument specifying whether the conditional or exact log-likelihood function should be used.

parametrization is the model parametrized with the "intercepts"  $\phi_{m,0}$  or "means"  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ ?

calc\_qresiduals should quantile residuals be calculated? Default is TRUE iff the model contains data.

<code>calc_cond_moments</code>	should conditional means and variances be calculated? Default is TRUE iff the model contains data.
<code>calc_std_errors</code>	should approximate standard errors be calculated?
<code>custom_h</code>	A numeric vector of with same length as the parameter vector of the estimated model: $i$ :th element of <code>custom_h</code> is the difference used in central difference approximation for differentials of the log-likelihood function for the $i$ :th parameter. If NULL (default), then the difference used for differentiating overly large degrees of freedom parameters is adjusted to avoid numerical problems, and the difference is $6e-6$ for the other parameters.
<code>object</code>	object of class 'gsmar' created with <code>fitGSMAR</code> or <code>GSMAR</code> .
<code>...</code>	graphical parameters passed to <code>ts.plot</code> .
<code>digits</code>	number of digits to be printed (max 20)
<code>x</code>	object of class 'gsmar' created with <code>fitGSMAR</code> or <code>GSMAR</code> .
<code>summary_print</code>	if set to TRUE then the print will include approximate standard errors, log-likelihood and information criteria values. Supported only for models with data.

### Details

Models can be built without data, e.g., in order to simulate from the process, but some elements such as quantile residuals and conditional moments can't be calculated without data.

### Value

Returns an object of class 'gsmar' defining the specified GMAR, StMAR or G-StMAR model. If data is supplied, the returned object contains (by default) empirical mixing weights, conditional means and variances and quantile residuals. Note that the first  $p$  observations are taken as the initial values so mixing weights, conditional moments and qresiduals start from the  $p+1$ :th observation (interpreted as  $t=1$ ).

### Methods (by generic)

- `logLik`: Log-likelihood method
- `residuals`: residuals method to extract multivariate quantile residuals
- `summary`: summary method, standard errors in brackets
- `plot`: plot method for class 'gsmar'
- `print`: print method

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's  $t$ -distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for the G-StMAR model, but it's a straightforward generalization with theoretical properties similar to the GMAR and StMAR models.

**See Also**

[fitGSMAR](#), [iterate\\_more](#), [add\\_data](#), [stmar\\_to\\_gstmar](#), [swap\\_parametrization](#), [get\\_gradient](#), [simulateGSMAR](#), [predict.gsmar](#), [condMoments](#), [uncondMoments](#)

**Examples**

```
# GMAR model
params12 <- c(0.18, 0.93, 0.01, 0.86, 0.68, 0.02, 0.88)
gmar12 <- GSMAR(data=logVIX, p=1, M=2, params=params12, model="GMAR")
gmar12

# Restricted GMAR model
params12r <- c(0.21, 0.23, 0.92, 0.01, 0.02, 0.86)
gmar12r <- GSMAR(data=logVIX, p=1, M=2, params=params12r, model="GMAR",
  restricted=TRUE)
gmar12r

# StMAR model, without data
params12t <- c(1.38, 0.88, 0.27, 3.8, 0.74, 3.15, 0.8, 300, 3.6)
stmar12t <- GSMAR(p=1, M=2, params=params12t, model="StMAR")
stmar12t

# G-StMAR model (similar to the StMAR model above), without data
params12gs <- c(1.38, 0.88, 0.27, 3.8, 0.74, 3.15, 0.8, 3.6)
gstmar12 <- GSMAR(p=1, M=c(1, 1), params=params12gs, model="G-StMAR")
gstmar12

# Restricted G-StMAR-model
params12gsr <- c(0.31, 0.33, 0.88, 0.01, 0.02, 0.77, 2.72)
gstmar12r <- GSMAR(data=logVIX, p=1, M=c(1, 1), params=params12gsr,
  model="G-StMAR", restricted=TRUE)
gstmar12r

# GMAR(p=2, M=2) model such that the second AR coefficient of the
# second regime is constrained to zero.
constraints <- list(diag(1, ncol=2, nrow=2), as.matrix(c(1, 0)))
params22c <- c(0.61, 0.83, -0.06, 0.02, 0.21, 0.91, 0.01, 0.16)
gmar22c <- GSMAR(logVIX, p=2, M=2, params=params22c,
  model="GMAR", constraints=constraints)
gmar22c

# Such StMAR(3,2) that the AR coefficients are restricted to be
# the same for both regimes and that the second AR coefficients are
# constrained to zero.
params32trc <- c(0.35, 0.33, 0.88, -0.02, 0.01, 0.01, 0.36, 4.53, 1000)
stmar32rc <- GSMAR(logVIX, p=3, M=2, params=params32trc, model="StMAR",
  restricted=TRUE, constraints=matrix(c(1, 0, 0, 0, 0, 1), ncol=2))
stmar32rc

# Mixture version of Heterogeneous autoregressive (HAR) model (without data)
paramsHAR <- c(1, 0.1, 0.2, 0.3, 1, 2, 0.15, 0.25, 0.35, 2, 0.55)
r1 = c(1, rep(0, 21)); r2 = c(rep(0.2, 5), rep(0, 17)); r3 = rep(1/22, 22)
```

```
R0 = cbind(r1, r2, r3)
mixhar <- GSMAR(p=22, M=2, params=paramsHAR, model="GMAR", constraints=list(R0, R0))
mixhar
```

---

IE *University of Michigan: inflation expectation time series: IE*

---

### Description

A dataset containing monthly inflation expectations from 1078I to 2018XII.

### Usage

IE

### Format

A class 'ts' time series object containing 492 observations.

### Source

<https://fred.stlouisfed.org/series/MICH>

---

isStationary *Check the stationary condition of specified GMAR, StMAR or G-StMAR model.*

---

### Description

isStationary checks the stationarity condition of the specified GMAR, StMAR or G-StMAR model.

### Usage

```
isStationary(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL)
```

### Arguments

p a positive integer specifying the order of AR coefficients.

M **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.  
**For G-StMAR model:** a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is M=M1+M2.

params a real valued parameter vector specifying the model.

**For non-restricted models: For GMAR model:** Size  $(M(p+3) - 1 \times 1)$  vector  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .

**For StMAR model:** Size  $(M(p+4) - 1 \times 1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(M(p+3) + M2 - 1 \times 1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .

**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices  $\mathbf{C}$  that satisfy  $\phi_m = R_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models: For GMAR model:** Size  $(3M + p - 1 \times 1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .

**For StMAR model:** Size  $(4M + p - 1 \times 1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(3M + M2 + p - 1 \times 1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .

**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix  $C$  that satisfies  $\phi = R\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If `parametrization=="mean"` just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.

model is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.

restricted a logical argument stating whether the AR coefficients  $\phi_{m,1}, \dots, \phi_{m,p}$  are restricted to be the same for all regimes.

constraints specifies linear constraints applied to the autoregressive parameters.

**For non-restricted models:** a list of size  $(pxq_m)$  constraint matrices  $C_m$  of full column rank satisfying  $\phi_m = C_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$  and  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models:** a size  $(pxq)$  constraint matrix  $C$  of full column rank satisfying  $\phi = C\psi$ , where  $\phi = (\phi_1, \dots, \phi_p)$  and  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always  $p$  for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

## Details

This function uses numerical approximations and it will falsely return FALSE for stationary models when the stationarity condition is really close to break.

## Value

Returns TRUE or FALSE accordingly.

## References

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for the G-StMAR model, but it's a straightforward generalization with theoretical properties similar to the GMAR and StMAR models.

## Examples

```
# GMAR model
params22 <- c(0.4, 0.39, 0.6, 0.3, 0.4, 0.1, 0.6, 0.3, 0.8)
isStationary(2, 2, params22)

# StMAR model
params12t <- c(-0.3, 1, 0.9, 0.1, 0.8, 0.6, 0.7, 10, 12)
isStationary(1, 2, params12t, model="StMAR")

# G-StMAR model
params12gs <- c(1, 0.1, 1, 2, 0.2, 2, 0.8, 20)
isStationary(1, c(1, 1), params12gs, model="G-StMAR")

# Restricted GMAR model
params13r <- c(0.1, 0.2, 0.3, -0.99, 0.1, 0.2, 0.3, 0.5, 0.3)
isStationary(1, 3, params13r, restricted=TRUE)

# Restricted StMAR model
params22tr <- c(-0.1, -0.2, 0.01, 0.99, 0.3, 0.4, 0.9, 3, 13)
isStationary(2, 2, params22tr, model="StMAR", restricted=TRUE)

# Restricted G-StMAR model
params13gsr <- c(1, 2, 3, -0.99, 1, 2, 3, 0.5, 0.4, 20, 30)
isStationary(1, c(1, 2), params13gsr, model="G-StMAR", restricted=TRUE)

# GMAR model as a mixture of AR(2) and AR(1) models
constraints <- list(diag(1, ncol=2, nrow=2), as.matrix(c(1, 0)))
params22c <- c(1.2, 0.8, 0.2, 0.3, 3.3, 0.7, 3, 0.8)
isStationary(2, 2, params22c, constraints=constraints)

# Such StMAR(3,2) that the AR coefficients are restricted to be the
# same for both regimes and that the second AR coefficients are
# constrained to zero.
params32trc <- c(1, 2, 0.8, -0.3, 1, 2, 0.7, 11, 12)
isStationary(3, 2, params32trc, model="StMAR", restricted=TRUE,
             constraints=matrix(c(1, 0, 0, 0, 0, 1), ncol=2))
```

---

isStationary\_int

*Check the stationary and identification conditions of specified GMAR, StMAR or G-StMAR model.*

---

**Description**

isStationary\_int checks the stationary condition and isIdentifiable checks the identification conditions of the specified GMAR, StMAR or G-StMAR model.

**Usage**

```
isStationary_int(p, M, params, restricted = FALSE)

isIdentifiable(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL)
```

**Arguments**

- p** a positive integer specifying the order of AR coefficients.
- M** **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.  
**For G-StMAR model:** a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is  $M=M1+M2$ .
- params** a real valued parameter vector specifying the model.  
**For non-restricted models: For GMAR model:** Size  $(M(p+3) - 1x1)$  vector  $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .  
**For StMAR model:** Size  $(M(p+4) - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .  
**For G-StMAR model:** Size  $(M(p+3) + M2 - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .  
**For restricted models: For GMAR model:** Size  $(3M + p - 1x1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .  
**For StMAR model:** Size  $(4M + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .  
**For G-StMAR model:** Size  $(3M + M2 + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .
- Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.
- restricted** a logical argument stating whether the AR coefficients  $\phi_{m,1}, \dots, \phi_{m,p}$  are restricted to be the same for all regimes.
- model** is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.
- constraints** specifies linear constraints applied to the autoregressive parameters.  
**For non-restricted models:** a list of size  $(pxq_m)$  constraint matrices  $C_m$  of full column rank satisfying  $\phi_m = C_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$  and  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models:** a size  $(p \times q)$  constraint matrix  $C$  of full column rank satisfying  $\phi = C\psi$ , where  $\phi = (\phi_1, \dots, \phi_p)$  and  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always  $p$  for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

### Details

These functions don't support models parametrized with general linear constraints.

### Value

Returns TRUE or FALSE accordingly.

### Warning

These functions don't have any argument checks!

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [econ.EM].
- There are currently no published references for the G-StMAR model, but it's a straightforward generalization with theoretical properties similar to the GMAR and StMAR models.

---

iterate_more	<i>Maximum likelihood estimation of GMAR, StMAR or G-StMAR model with preliminary estimates</i>
--------------	---

---

### Description

iterate\_more uses variable metric algorithm to finalize maximum likelihood estimation of GMAR, StMAR or G-StMAR model (object of class 'gsmarar') which already has preliminary estimates.

### Usage

```
iterate_more(gsmar, maxit = 100, custom_h = NULL,
             calc_std_errors = TRUE)
```



**Arguments**

gsmar	object of class 'gsmar' created with the function <code>fitGSMAR</code> or <code>GSMAR</code> .
maxit	maximum number of iterations in the variable metric algorithm.
custom_h	A numeric vector of with same length as the parameter vector of the estimated model: <i>i</i> :th element of <code>custom_h</code> is the difference used in central difference approximation for differentials of the log-likelihood function for the <i>i</i> :th parameter. If <code>NULL</code> (default), then the difference used for differentiating overly large degrees of freedom parameters is adjusted to avoid numerical problems, and the difference is $6e-6$ for the other parameters.
calc_std_errors	should approximate standard errors be calculated?

**Details**

The main purpose of `iterate_more` is to provide a simple and convenient tool to finalize the estimation when the maximum number of iterations is reached when estimating a model with the main estimation function `fitGSMAR`. It's just a simple wrapper around function `optim` from the package `stats` and `GSMAR` from the package `uGSMAR`.

**Value**

Returns an object of class 'gsmar' defining the estimated model. Can be used to work with other functions provided in `uGSMAR`.

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for the G-StMAR model, but it's a straightforward generalization with theoretical properties similar to the GMAR and StMAR models.

**See Also**

[fitGSMAR](#), [GSMAR](#), [stmar\\_to\\_gstmar](#), [optim](#)

**Examples**

```
# Estimate GMAR model with only 50 generations of genetic algorithm and
# only 1 iteration in variable metric algorithm
fit12 <- fitGSMAR(logVIX, 1, 2, maxit=1, ngen=50, ncalls=1, seeds=1)
fit12

# Iterate more since iteration limit was reached
fit12 <- iterate_more(fit12)
fit12
```

loglikelihood

Compute the log-likelihood of GMAR, StMAR or G-StMAR model

**Description**

loglikelihood computes the log-likelihood value of the specified GMAR, StMAR or G-StMAR model. Exists for convenience if one wants to for example plot profile log-likelihoods or employ other estimation algorithms. Use minval to control what happens when the parameter vector is outside the parameter space.

**Usage**

```
loglikelihood(data, p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL, conditional = TRUE,
  parametrization = c("intercept", "mean"), returnTerms = FALSE,
  minval = NA)
```

**Arguments**

**data** a numeric vector class 'ts' object containing the data. NA values are not supported.

**p** a positive integer specifying the order of AR coefficients.

**M** **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.

**For G-StMAR model:** a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is M=M1+M2.

**params** a real valued parameter vector specifying the model.

**For non-restricted models: For GMAR model:** Size  $(M(p+3) - 1x1)$  vector  $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .

**For StMAR model:** Size  $(M(p+4) - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(M(p+3) + M2 - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .

**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices **C** that satisfy  $\phi_m = R_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models: For GMAR model:** Size  $(3M + p - 1x1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .

**For StMAR model:** Size  $(4M + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(3M + M2 + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .

**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix **C** that satisfies  $\phi = R\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0}/(1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.

model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	specifies linear constraints applied to the autoregressive parameters.  <b>For non-restricted models:</b> a list of size $(pxq_m)$ constraint matrices $C_m$ of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$ , where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ . <b>For restricted models:</b> a size $(pxq)$ constraint matrix $C$ of full column rank satisfying $\phi = C\psi$ , where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$ .  Symbol $\phi$ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always $p$ for all regimes. Ignore or set to NULL if applying linear constraints is <b>not</b> desired.
conditional parametrization	a logical argument specifying whether the conditional or exact log-likelihood function should be used.  is the model parametrized with the "intercepts" $\phi_{m,0}$ or "means" $\mu_m = \phi_{m,0}/(1 - \sum \phi_{i,m})$ ?
returnTerms	should the terms $l_t : t = 1, \dots, T$ in the log-likelihood function (see <i>KMS 2015, eq.(13)</i> ) be returned instead of the log-likelihood value?
minval	this will be returned when the parameter vector is outside the parameter space.

## Value

Returns the log-likelihood value or the terms described in returnTerms.

## References

- Galbraith, R., Galbraith, J. 1974. On the inverses of some patterned matrices arising in the theory of stationary time series. *Journal of Applied Probability* **11**, 63-71.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for the G-StMAR model, but it's a straightforward generalization with theoretical properties similar to the GMAR and StMAR models.

**See Also**

[fitGSMAR](#), [GSMAR](#), [quantileResiduals](#), [mixingWeights](#), [calc\\_gradient](#)

**Examples**

```
# GMAR model
params12 <- c(0.18, 0.93, 0.01, 0.86, 0.68, 0.02, 0.88)
loglikelihood(logVIX, 1, 2, params12)

# Restricted GMAR model, outside parameter space
params12r <- c(0.21, 0.23, 0.92, 0.01, 0.02, 0.86)
loglikelihood(logVIX, 1, 2, params12r, restricted=TRUE)

# Non-mixture version of StMAR model, outside parameter space
params11t <- c(0.16, 0.93, 0.00, 3.01)
loglikelihood(logVIX, 1, 1, params11t, model="StMAR", minval="Hello")

# G-StMAR model
params12gs <- c(0.86, 0.68, 0.02, 0.18, 0.93, 0.01, 0.11, 44.36)
loglikelihood(logVIX, 1, c(1, 1), params12gs, model="G-StMAR")

# Restricted G-StMAR model
params12gsr <- c(0.31, 0.33, 0.88, 0.01, 0.02, 0.77, 2.72)
loglikelihood(logVIX, 1, c(1, 1), params12gsr, model="G-StMAR",
  restricted=TRUE)

# GMAR model as a mixture of AR(2) and AR(1) models
constraints <- list(diag(1, ncol=2, nrow=2), as.matrix(c(1, 0)))
params22c <- c(0.61, 0.83, -0.06, 0.02, 0.21, 0.91, 0.01, 0.16)
loglikelihood(logVIX, 2, 2, params22c, constraints=constraints)

# Such StMAR(3,2) that the AR coefficients are restricted to be
# the same for both regimes and that the second AR coefficients are
# constrained to zero.
params32trc <- c(0.35, 0.33, 0.88, -0.02, 0.01, 0.01, 0.36, 4.53, 1000)
loglikelihood(logVIX, 3, 2, params32trc, model="StMAR", restricted=TRUE,
  constraints=matrix(c(1, 0, 0, 0, 0, 1), ncol=2))
```

---

loglikelihood\_int

*Compute the log-likelihood of GMAR, StMAR or G-StMAR model*

---

**Description**

loglikelihood\_int computes the log-likelihood value of the specified GMAR, StMAR or G-StMAR model for the given data.

**Usage**

```
loglikelihood_int(data, p, M, params, model = c("GMAR", "StMAR",
"G-StMAR"), restricted = FALSE, constraints = NULL,
conditional = TRUE, parametrization = c("intercept", "mean"),
boundaries = TRUE, checks = TRUE, to_return = c("loglik", "mw",
"mw_tplus1", "loglik_and_mw", "terms", "regime_cmeans", "regime_cvars",
"total_cmeans", "total_cvars"), minval)
```

**Arguments**

- data** a numeric vector class 'ts' object containing the data. NA values are not supported.
- p** a positive integer specifying the order of AR coefficients.
- M** **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.  
**For G-StMAR model:** a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is M=M1+M2.
- params** a real valued parameter vector specifying the model.  
**For non-restricted models: For GMAR model:** Size  $(M(p+3) - 1x1)$  vector  $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .  
**For StMAR model:** Size  $(M(p+4) - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .  
**For G-StMAR model:** Size  $(M(p+3) + M2 - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .  
**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices **C** that satisfy  $\phi_m = R_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .  
**For restricted models: For GMAR model:** Size  $(3M + p - 1x1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .  
**For StMAR model:** Size  $(4M + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .  
**For G-StMAR model:** Size  $(3M + M2 + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .  
**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix **C** that satisfies  $\phi = R\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .
- Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.
- model** is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.

restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	<p>specifies linear constraints applied to the autoregressive parameters.</p> <p><b>For non-restricted models:</b> a list of size <math>(pxq_m)</math> constraint matrices <math>C_m</math> of full column rank satisfying <math>\phi_m = C_m \psi_m</math> for all <math>m = 1, \dots, M</math>, where <math>\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})</math> and <math>\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})</math>.</p> <p><b>For restricted models:</b> a size <math>(pxq)</math> constraint matrix <math>C</math> of full column rank satisfying <math>\phi = C\psi</math>, where <math>\phi = (\phi_1, \dots, \phi_p)</math> and <math>\psi = (\psi_1, \dots, \psi_q)</math>.</p> <p>Symbol <math>\phi</math> denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always <math>p</math> for all regimes. Ignore or set to NULL if applying linear constraints is <b>not</b> desired.</p>
conditional parametrization	a logical argument specifying whether the conditional or exact log-likelihood function should be used.
boundaries	<p>is the model parametrized with the "intercepts" <math>\phi_{m,0}</math> or "means" <math>\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})</math>?</p> <p>a logical argument. If TRUE then loglikelihood returns minval if...</p> <ul style="list-style-type: none"> <li>• any component variance is not larger than zero,</li> <li>• any parametrized mixing weight <math>\alpha_1, \dots, \alpha_{M-1}</math> is not larger than zero,</li> <li>• sum of the parametrized mixing weights is not smaller than one,</li> <li>• if the model is not stationary,</li> <li>• or if model=="StMAR" or model=="G-StMAR" and any degrees of freedom parameter <math>\nu_m</math> is not larger than two.</li> </ul> <p>Argument minval will be used only if boundaries==TRUE.</p>
checks	an (optional) logical argument defining whether argument checks are made. If FALSE then no argument checks such as stationary checks etc are made. The default is TRUE.
to_return	should the returned object be the log-likelihood value, mixing weights, mixing weights including value for $\alpha_{m,T+1}$ , a list containing log-likelihood value and mixing weights, the terms $l_t : t = 1, \dots, T$ in the log-likelihood function (see <i>KMS 2015, eq.(13)</i> ), regimewise conditional means, regimewise conditional variances, total conditional means or total conditional variances? Default is the log-likelihood value ("loglik").
minval	this will be returned when the parameter vector is outside the parameter space.

### Value

Note that the first  $p$  observations are taken as the initial values so mixing weights and conditional moments start from the  $p+1$ :th observation (interpreted as  $t=1$ ).

**By default:** log-likelihood value of the specified model,

**If to\_return=="mw":** a size  $((n\_obs-p) \times M)$  matrix containing the mixing weights: for  $m$ :th component in  $m$ :th column.

**If to\_return=="mw\_tplus1":** a size  $((n\_obs-p+1) \times M)$  matrix containing the mixing weights: for  $m$ :th component in  $m$ :th column. The last row is for  $\alpha_{m,T+1}$ .

- if** to\_return=="loglik\_and\_mw": a list of two elements. The first element contains the log-likelihood value and the second element contains the mixing weights.
- if** to\_return=="terms": a size ((n\_obs-p)x1) numeric vector containing the terms  $l_t$ .
- if** to\_return=="regime\_cmeans": a size ((n\_obs-p)xM) matrix containing the regime specific conditional means.
- if** to\_return=="regime\_cvars": a size ((n\_obs-p)xM) matrix containing the regime specific conditional variances.
- if** to\_return=="total\_cmeans": a size ((n\_obs-p)x1) vector containing the total conditional means.
- if** to\_return=="total\_cvars": a size ((n\_obs-p)x1) vector containing the total conditional variances.

## References

- Galbraith, R., Galbraith, J. 1974. On the inverses of some patterned matrices arising in the theory of stationary time series. *Journal of Applied Probability* **11**, 63-71.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for the G-StMAR model, but it's a straightforward generalization with theoretical properties similar to the GMAR and StMAR models.

---

logVIX

*CBOE Volatility Index: logVIX*

---

## Description

A dataset containing natural logarithm of daily CBOE Volatility Index from 5th of July 2016 to 4th of July 2017.

## Usage

logVIX

## Format

A numeric vector containing 252 observations.

## Source

<https://fred.stlouisfed.org/series/VIXCLS>

mixingWeights

Calculate mixing weights of GMAR, StMAR or G-StMAR model

**Description**

mixingWeights calculates the mixing weights of the specified GMAR, StMAR or G-StMAR model and returns them as a matrix.

**Usage**

```
mixingWeights(data, p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL,
  parametrization = c("intercept", "mean"))
```

**Arguments**

**data** a numeric vector class 'ts' object containing the data. NA values are not supported.

**p** a positive integer specifying the order of AR coefficients.

**M** **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.

**For G-StMAR model:** a size  $(2 \times 1)$  vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is  $M=M1+M2$ .

**params** a real valued parameter vector specifying the model.

**For non-restricted models: For GMAR model:** Size  $(M(p+3) - 1 \times 1)$  vector  $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .

**For StMAR model:** Size  $(M(p+4) - 1 \times 1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(M(p+3) + M2 - 1 \times 1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .

**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices **C** that satisfy  $\phi_m = R_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models: For GMAR model:** Size  $(3M + p - 1 \times 1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .

**For StMAR model:** Size  $(4M + p - 1 \times 1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(3M + M2 + p - 1 \times 1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .

**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix **C** that satisfies  $\phi = R\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest



M2 components are *StMAR-type*. Note that in the case  $\mathbf{M=1}$  the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.

model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	specifies linear constraints applied to the autoregressive parameters. <b>For non-restricted models:</b> a list of size $(pxq_m)$ constraint matrices $C_m$ of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$ , where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ . <b>For restricted models:</b> a size $(pxq)$ constraint matrix $C$ of full column rank satisfying $\phi = C\psi$ , where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$ . Symbol $\phi$ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always $p$ for all regimes. Ignore or set to NULL if applying linear constraints is <b>not</b> desired.
parametrization	is the model parametrized with the "intercepts" $\phi_{m,0}$ or "means" $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ ?

## Value

- If to\_return=="mw":** a size  $((n\_obs-p) \times M)$  matrix containing the mixing weights: for  $m$ :th component in  $m$ :th column.
- If to\_return=="mw\_tplus1":** a size  $((n\_obs-p+1) \times M)$  matrix containing the mixing weights: for  $m$ :th component in  $m$ :th column. The last row is for  $\alpha_{m,T+1}$ .

## References

- Galbraith, R., Galbraith, J. 1974. On the inverses of some patterned matrices arising in the theory of stationary time series. *Journal of Applied Probability* **11**, 63-71.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [**econ.EM**].
- There are currently no published references for the G-StMAR model, but it's a straightforward generalization with theoretical properties similar to the GMAR and StMAR models.

## Examples

```
# GMAR model
params12 <- c(0.18, 0.93, 0.01, 0.86, 0.68, 0.02, 0.88)
mixingWeights(logVIX, 1, 2, params12)

# Restricted GMAR model, outside parameter space
```

```

params12r <- c(0.21, 0.23, 0.92, 0.01, 0.02, 0.86)
mixingWeights(logVIX, 1, 2, params12r, restricted=TRUE)

# Non-mixture version of StMAR model, outside parameter space
params11t <- c(0.16, 0.93, 0.01, 3.01)
mixingWeights(logVIX, 1, 1, params11t, model="StMAR")

# G-StMAR model
params12gs <- c(0.86, 0.68, 0.02, 0.18, 0.93, 0.01, 0.11, 44.36)
mixingWeights(logVIX, 1, c(1, 1), params12gs, model="G-StMAR")

# Restricted G-StMAR model
params12gsr <- c(0.31, 0.33, 0.88, 0.01, 0.02, 0.77, 2.72)
mixingWeights(logVIX, 1, c(1, 1), params12gsr, model="G-StMAR", restricted=TRUE)

# GMAR model as a mixture of AR(2) and AR(1) models
constraints <- list(diag(1, ncol=2, nrow=2), as.matrix(c(1, 0)))
params22c <- c(0.61, 0.83, -0.06, 0.02, 0.21, 0.91, 0.01, 0.16)
mixingWeights(logVIX, 2, 2, params22c, constraints=constraints)

# Such StMAR(3,2) that the AR coefficients are restricted to be
# the same for both regimes and that the second AR coefficients are
# constrained to zero.
params32trc <- c(0.35, 0.33, 0.88, -0.02, 0.01, 0.01, 0.36, 4.53, 1000)
mixingWeights(logVIX, 3, 2, params32trc, model="StMAR", restricted=TRUE,
              constraints=matrix(c(1, 0, 0, 0, 0, 1), ncol=2))

```

---

mixingWeights\_int      *Calculate mixing weights of GMAR, StMAR or G-StMAR model*

---

## Description

mixingWeights\_int calculates the mixing weights of the specified GMAR, StMAR or G-StMAR model and returns them as a matrix.

## Usage

```

mixingWeights_int(data, p, M, params, model = c("GMAR", "StMAR",
        "G-StMAR"), restricted = FALSE, constraints = NULL,
        parametrization = c("intercept", "mean"), checks = TRUE,
        to_return = c("mw", "mw_tplus1"))

```

## Arguments

data	a numeric vector class 'ts' object containing the data. NA values are not supported.
p	a positive integer specifying the order of AR coefficients.
M	<b>For GMAR and StMAR models:</b> a positive integer specifying the number of mixture components.

	<p><b>For G-StMAR model:</b> a size <math>(2 \times 1)</math> vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is <math>M=M1+M2</math>.</p>
params	<p>a real valued parameter vector specifying the model.</p> <p><b>For non-restricted models: For GMAR model:</b> Size <math>(M(p+3) - 1 \times 1)</math> vector <math>\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})</math>, where <math>v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)</math> and <math>\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})</math>, <math>m = 1, \dots, M</math>.</p> <p><b>For StMAR model:</b> Size <math>(M(p+4) - 1 \times 1)</math> vector <math>(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)</math>.</p> <p><b>For G-StMAR model:</b> Size <math>(M(p+3) + M2 - 1 \times 1)</math> vector <math>(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)</math>.</p> <p><b>With linear constraints:</b> Replace the vectors <math>\phi_m</math> with vectors <math>\psi_m</math> and provide a list of constraint matrices <b>C</b> that satisfy <math>\phi_m = R_m \psi_m</math> for all <math>m = 1, \dots, M</math>, where <math>\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})</math>.</p> <p><b>For restricted models: For GMAR model:</b> Size <math>(3M + p - 1 \times 1)</math> vector <math>\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})</math>, where <math>\phi = (\phi_1, \dots, \phi_M)</math>.</p> <p><b>For StMAR model:</b> Size <math>(4M + p - 1 \times 1)</math> vector <math>(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)</math>.</p> <p><b>For G-StMAR model:</b> Size <math>(3M + M2 + p - 1 \times 1)</math> vector <math>(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)</math>.</p> <p><b>With linear constraints:</b> Replace the vector <math>\phi</math> with vector <math>\psi</math> and provide a constraint matrix <b>C</b> that satisfies <math>\phi = R\psi</math>, where <math>\psi = (\psi_1, \dots, \psi_q)</math>.</p> <p>Symbol <math>\phi</math> denotes an AR coefficient, <math>\sigma^2</math> a variance, <math>\alpha</math> a mixing weight and <math>\nu</math> a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term <math>\phi_{m,0}</math> with regimewise mean <math>\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})</math>. In the <b>G-StMAR</b> model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i>. Note that in the case <b>M=1</b> the parameter <math>\alpha</math> is dropped, and in the case of <b>StMAR</b> or <b>G-StMAR</b> model the degrees of freedom parameters <math>\nu_m</math> have to be larger than 2.</p>
model	<p>is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i>.</p>
restricted	<p>a logical argument stating whether the AR coefficients <math>\phi_{m,1}, \dots, \phi_{m,p}</math> are restricted to be the same for all regimes.</p>
constraints	<p>specifies linear constraints applied to the autoregressive parameters.</p> <p><b>For non-restricted models:</b> a list of size <math>(p \times q_m)</math> constraint matrices <math>C_m</math> of full column rank satisfying <math>\phi_m = C_m \psi_m</math> for all <math>m = 1, \dots, M</math>, where <math>\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})</math> and <math>\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})</math>.</p> <p><b>For restricted models:</b> a size <math>(p \times q)</math> constraint matrix <b>C</b> of full column rank satisfying <math>\phi = C\psi</math>, where <math>\phi = (\phi_1, \dots, \phi_p)</math> and <math>\psi = (\psi_1, \dots, \psi_q)</math>.</p> <p>Symbol <math>\phi</math> denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is <b>not</b> desired.</p>
parametrization	<p>is the model parametrized with the "intercepts" <math>\phi_{m,0}</math> or "means" <math>\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})</math>?</p>
checks	<p>an (optional) logical argument defining whether argument checks are made. If FALSE then no argument checks such as stationary checks etc are made. The default is TRUE.</p>

to\_return should the returned object the mixing weights or mixing weights ("mw") including value for  $\alpha_{m,T+1}$  ("mw\_tplus1")?

### Value

**If to\_return=="mw":** a size  $((n\_obs-p) \times M)$  matrix containing the mixing weights: for m:th component in m:th column.

**If to\_return=="mw\_tplus1":** a size  $((n\_obs-p+1) \times M)$  matrix containing the mixing weights: for m:th component in m:th column. The last row is for  $\alpha_{m,T+1}$ .

### References

- Galbraith, R., Galbraith, J. 1974. On the inverses of some patterned matrices arising in the theory of stationary time series. *Journal of Applied Probability* **11**, 63-71.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for the G-StMAR model, but it's a straightforward generalization with theoretical properties similar to the GMAR and StMAR models.

---

nParams

*Calculate the number of parameters*

---

### Description

nParams calculates the number of parameters that should be in the parameter vector.

### Usage

```
nParams(p, M, model = c("GMAR", "StMAR", "G-StMAR"),
        restricted = FALSE, constraints = NULL)
```

### Arguments

p a positive integer specifying the order of AR coefficients.

M **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.  
**For G-StMAR model:** a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is  $M=M1+M2$ .

model is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.

restricted a logical argument stating whether the AR coefficients  $\phi_{m,1}, \dots, \phi_{m,p}$  are restricted to be the same for all regimes.

constraints specifies linear constraints applied to the autoregressive parameters.

**For non-restricted models:** a list of size  $(pxq_m)$  constraint matrices  $C_m$  of full column rank satisfying  $\phi_m = C_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$  and  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models:** a size  $(pxq)$  constraint matrix  $C$  of full column rank satisfying  $\phi = C\psi$ , where  $\phi = (\phi_1, \dots, \phi_p)$  and  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always  $p$  for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

### Value

Returns the number of parameters.

---

parameterChecks	<i>Check the parameter vector is specified correctly</i>
-----------------	--

---

### Description

parameterChecks checks dimension, restrictions and stationarity of the given parameters of GMAR, StMAR or G-StMAR model. Throws an error if any check fails. Does NOT consider the identifiability condition!

### Usage

```
parameterChecks(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL)
```

### Arguments

**p** a positive integer specifying the order of AR coefficients.

**M** **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.  
**For G-StMAR model:** a size  $(2 \times 1)$  vector specifying the number of *GMAR-type* components  $M1$  in the first element and *StMAR-type* components  $M2$  in the second. The total number of mixture components is  $M = M1 + M2$ .

**params** a real valued parameter vector specifying the model.  
**For GMAR model:** Size  $(M(p+3) - 1 \times 1)$  vector  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .  
**For StMAR model:** Size  $(M(p+4) - 1 \times 1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .  
**For G-StMAR model:** Size  $(M(p+3) + M2 - 1 \times 1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .

**model** is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first  $M1$  components are *GMAR-type* and the rest  $M2$  components are *StMAR-type*.

restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	specifies linear constraints applied to the autoregressive parameters. <b>For non-restricted models:</b> a list of size $(pxq_m)$ constraint matrices $C_m$ of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$ , where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ . <b>For restricted models:</b> a size $(pxq)$ constraint matrix $C$ of full column rank satisfying $\phi = C\psi$ , where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$ . Symbol $\phi$ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always $p$ for all regimes. Ignore or set to NULL if applying linear constraints is <b>not</b> desired.

### Value

Throws an informative error if any check fails. Doesn't return anything.

---

pick_alphas	<i>Pick mixing weights parameters from parameter vector</i>
-------------	---

---

### Description

pick\_alphas picks and returns the mixing weights parameters (including the non-parametrized one for the last component) from parameter vector.

### Usage

```
pick_alphas(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL)
```

### Arguments

p	a positive integer specifying the order of AR coefficients.
M	<b>For GMAR and StMAR models:</b> a positive integer specifying the number of mixture components. <b>For G-StMAR model:</b> a size $(2 \times 1)$ vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is $M = M1 + M2$ .
params	a real valued parameter vector specifying the model. <b>For non-restricted models: For GMAR model:</b> Size $(M(p+3) - 1 \times 1)$ vector $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ , $m = 1, \dots, M$ . <b>For StMAR model:</b> Size $(M(p+4) - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ . <b>For G-StMAR model:</b> Size $(M(p+3) + M2 - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .

**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices  $C$  that satisfy  $\phi_m = R_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models: For GMAR model:** Size  $(3M + p - 1 \times 1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .

**For StMAR model:** Size  $(4M + p - 1 \times 1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(3M + M2 + p - 1 \times 1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix  $C$  that satisfies  $\phi = R\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If parametrization="mean" just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.

model is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.

restricted a logical argument stating whether the AR coefficients  $\phi_{m,1}, \dots, \phi_{m,p}$  are restricted to be the same for all regimes.

constraints specifies linear constraints applied to the autoregressive parameters.

**For non-restricted models:** a list of size  $(pq \times q_m)$  constraint matrices  $C_m$  of full column rank satisfying  $\phi_m = C_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$  and  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models:** a size  $(pq \times q)$  constraint matrix  $C$  of full column rank satisfying  $\phi = C\psi$ , where  $\phi = (\phi_1, \dots, \phi_p)$  and  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always  $p$  for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

## Value

Returns a vector of length  $M$  containing the mixing weights parameters  $\alpha_m$ .

---

pick\_dfs

*Pick degrees of freedom parameters from parameter vector*

---

## Description

pick\_dfs picks and returns the degrees of freedom parameters from parameter vector.

## Usage

```
pick_dfs(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"))
```

**Arguments**

p	a positive integer specifying the order of AR coefficients.
M	<b>For GMAR and StMAR models:</b> a positive integer specifying the number of mixture components. <b>For G-StMAR model:</b> a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is M=M1+M2.
params	a real valued parameter vector specifying the model. <b>For non-restricted models: For GMAR model:</b> Size $(M(p+3) - 1 \times 1)$ vector $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$ and $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ , $m = 1, \dots, M$ . <b>For StMAR model:</b> Size $(M(p+4) - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ . <b>For G-StMAR model:</b> Size $(M(p+3) + M2 - 1 \times 1)$ vector $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ . <b>With linear constraints:</b> Replace the vectors $\phi_m$ with vectors $\psi_m$ and provide a list of constraint matrices <b>C</b> that satisfy $\phi_m = R_m \psi_m$ for all $m = 1, \dots, M$ , where $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ . <b>For restricted models: For GMAR model:</b> Size $(3M + p - 1 \times 1)$ vector $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where $\phi = (\phi_1, \dots, \phi_M)$ . <b>For StMAR model:</b> Size $(4M + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ . <b>For G-StMAR model:</b> Size $(3M + M2 + p - 1 \times 1)$ vector $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ . <b>With linear constraints:</b> Replace the vector $\phi$ with vector $\psi$ and provide a constraint matrix <b>C</b> that satisfies $\phi = R\psi$ , where $\psi = (\psi_1, \dots, \psi_q)$ . Symbol $\phi$ denotes an AR coefficient, $\sigma^2$ a variance, $\alpha$ a mixing weight and $\nu$ a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term $\phi_{m,0}$ with regimewise mean $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ . In the <b>G-StMAR</b> model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> . Note that in the case <b>M=1</b> the parameter $\alpha$ is dropped, and in the case of <b>StMAR</b> or <b>G-StMAR</b> model the degrees of freedom parameters $\nu_m$ have to be larger than 2.
model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .

**Value**

Returns a vector of length M or M2 containing the degrees of freedom parameters

---

pick_pars	Pick $\phi_0$ (or $\mu$ ), AR-coefficients and variance parameters from parameter vector
-----------	--

---

**Description**

pick\_pars picks  $\phi_0/\mu$ , ar-coefficient and variance parameters from parameter vector



**Usage**

```
pick_pars(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
         restricted = FALSE, constraints = NULL)
```

**Arguments**

- p** a positive integer specifying the order of AR coefficients.
- M** **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.  
**For G-StMAR model:** a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is  $M=M1+M2$ .
- params** a real valued parameter vector specifying the model.  
**For non-restricted models: For GMAR model:** Size  $(M(p+3) - 1x1)$  vector  $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .  
**For StMAR model:** Size  $(M(p+4) - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .  
**For G-StMAR model:** Size  $(M(p+3) + M2 - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .  
**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices **C** that satisfy  $\phi_m = R_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .  
**For restricted models: For GMAR model:** Size  $(3M + p - 1x1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .  
**For StMAR model:** Size  $(4M + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .  
**For G-StMAR model:** Size  $(3M + M2 + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .  
**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix **C** that satisfies  $\phi = R\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .
- Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If `parameterization=="mean"` just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.
- model** is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.
- restricted** a logical argument stating whether the AR coefficients  $\phi_{m,1}, \dots, \phi_{m,p}$  are restricted to be the same for all regimes.
- constraints** specifies linear constraints applied to the autoregressive parameters.  
**For non-restricted models:** a list of size  $(pxq_m)$  constraint matrices  $C_m$  of full column rank satisfying  $\phi_m = C_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$  and  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models:** a size  $(pxq)$  constraint matrix  $C$  of full column rank satisfying  $\phi=C\psi$ , where  $\phi=(\phi_1, \dots, \phi_p)$  and  $\psi=(\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always  $p$  for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

### Value

Returns a  $(Mx(p+2))$  matrix containing the parameters, column for each component. First row for  $\phi_0$  or  $\mu$  depending on the parametrization, second row for  $\phi_1, \dots$ , second last row for  $\phi_p$  and last row for  $\sigma^2$ .

---

pick_phi0	<i>Pick phi0 or mean parameters from parameter vector</i>
-----------	---

---

### Description

pick\_phi0 picks and returns the phi0 or mean parameters from parameter vector.

### Usage

```
pick_phi0(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL)
```

### Arguments

- p** a positive integer specifying the order of AR coefficients.
- M** **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.  
**For G-StMAR model:** a size  $(2x1)$  vector specifying the number of *GMAR-type* components  $M1$  in the first element and *StMAR-type* components  $M2$  in the second. The total number of mixture components is  $M=M1+M2$ .
- params** a real valued parameter vector specifying the model.  
**For non-restricted models: For GMAR model:** Size  $(M(p+3)-1x1)$  vector  $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m=(\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m=(\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m=1, \dots, M$ .  
**For StMAR model:** Size  $(M(p+4)-1x1)$  vector  $(\theta, \nu)=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .  
**For G-StMAR model:** Size  $(M(p+3)+M2-1x1)$  vector  $(\theta, \nu)=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .  
**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices  $C$  that satisfy  $\phi_m=R_m\psi_m$  for all  $m=1, \dots, M$ , where  $\psi_m=(\psi_{m,1}, \dots, \psi_{m,q_m})$ .  
**For restricted models: For GMAR model:** Size  $(3M+p-1x1)$  vector  $\theta=(\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi=(\phi_1, \dots, \phi_M)$ .  
**For StMAR model:** Size  $(4M+p-1x1)$  vector  $(\theta, \nu)=(\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ .

**For G-StMAR model:** Size  $(3M+M2+p-1 \times 1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots)$ .

**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix  $C$  that satisfies  $\phi = R\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.

model is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.

restricted a logical argument stating whether the AR coefficients  $\phi_{m,1}, \dots, \phi_{m,p}$  are restricted to be the same for all regimes.

constraints specifies linear constraints applied to the autoregressive parameters.

**For non-restricted models:** a list of size  $(pxq_m)$  constraint matrices  $C_m$  of full column rank satisfying  $\phi_m = C_m\psi_m$  for all  $m = 1, \dots, M$ , where  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$  and  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models:** a size  $(pxq)$  constraint matrix  $C$  of full column rank satisfying  $\phi = C\psi$ , where  $\phi = (\phi_1, \dots, \phi_p)$  and  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always  $p$  for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

## Value

Returns a vector of length M containing the phi0 or mean parameters depending parametrization.

---

plot.gsmarpred *plot method for class 'gsmarpred' objects*

---

## Description

plot.gsmarpred is plot method for gsmarpred objects

## Usage

```
## S3 method for class 'gsmarpred'
plot(x, ..., nt, add_grid = TRUE)
```

**Arguments**

x	object of class 'gsmarpred' created with predict.gsmar.
...	arguments passed to grid
nt	a positive integer specifying the number of observations to be plotted along with the prediction. Default is round(length(data)*0.2).
add_grid	should grid be added to the plots?

**Details**

This method is used plot forecasts of gsmar processes

**Examples**

```
# GMAR-model
params12 <- c(0.18281409, 0.92657275, 0.00214552,
             0.85725129, 0.68210294, 0.01900299, 0.88342018)
gmar12 <- GSMAR(logVIX, 1, 2, params12)
pred <- predict(gmar12, n_ahead=10, plotRes=FALSE)
plot(pred, nt=50)
```

---

plot.qrtest

*Quantile residual tests for GMAR, StMAR or G-StMAR model*


---

**Description**

quantileResidualTests performs quantile residual tests for GMAR, StMAR or G-StMAR model, testing normality, autocorrelation and conditional heteroscedasticity.

**Usage**

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

```
## S3 method for class 'qrtest'
print(x, ..., digits = 3)
```

```
quantileResidualTests(gsmar, lagsAC = c(1, 2, 5, 10), lagsCH = lagsAC,
                      nsimu = 2000, printRes = TRUE)
```

**Arguments**

x	object of class 'qrtest' created with the function quantileResidualTests.
...	graphical parameters passed to segments in plot.qrtest. Currently not used in print.qrtest

digits	number of decimals to print
gsmar	object of class 'gsmar' created with the function fitGSMAR or GSMAR.
lagsAC	a numeric vector of positive integers specifying the lags for which autocorrelation is tested.
lagsCH	a numeric vector of positive integers specifying the lags for which conditional heteroscedasticity is tested.
nsimu	a positive integer specifying to how many simulated observations the covariance matrix Omega (see Kalliovirta (2012)) should be based on. If smaller than data size, then omega will be based on the given data.
printRes	a logical argument defining whether results should be printed or not.

### Details

For details about the quantile residual tests see the cited article by *Kalliovirta (2012)*.

### Value

Returns an object of class 'qrtest' containing the test results in data frames. In the cases of autocorrelation and conditional heteroscedasticity tests, the returned object also contains the associated individual statistics and their standard errors, discussed by *Kalliovirta (2012)* at the pages 369-370.

### Methods (by generic)

- plot: plot p-values of the autocorrelation and conditional heteroskedasticity tests.
- print: print method for class 'qrtest'

### Suggested packages

Install the suggested package "gsl" for faster evaluations in the cases of StMAR and G-StMAR models. For large StMAR and G-StMAR models with large data the evaluations may take significantly long time without the package "gsl".

### References

- Galbraith, R., Galbraith, J. 1974. On the inverses of some patterned matrices arising in the theory of stationary time series. *Journal of Applied Probability* **11**, 63-71.
- Kalliovirta L. (2012) Misspecification tests based on quantile residuals. *The Econometrics Journal*, **15**, 358-393.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [econ.EM].
- There are currently no published references for G-StMAR model, but it's a straight forward generalization with theoretical properties similar to GMAR and StMAR models.

**See Also**

[fitGSMAR](#), [GSMAR](#), [diagnosticPlot](#), [predict.gsmar](#), [getOmega](#),

**Examples**

```
# GMAR model
fit12 <- fitGSMAR(data=logVIX, p=1, M=2, model="GMAR")
qrtest12 <- quantileResidualTests(fit12)
plot(qrtest12)

# Restricted GMAR model
fit12r <- fitGSMAR(logVIX, 1, 2, model="GMAR", restricted=TRUE)
qrtest12r <- quantileResidualTests(fit12r, lagsAC=1:10, nsimu=1)
plot(qrtest12r)

# Non-mixture version of StMAR model
fit11t <- fitGSMAR(logVIX, 1, 1, model="StMAR", ncores=1, ncalls=1)
quantileResidualTests(fit11t, lagsAC=c(1, 2, 5), nsimu=1, printRes=FALSE)

# G-StMAR model
fit12gs <- fitGSMAR(logVIX, 1, M=c(1, 1), model="G-StMAR")
quantileResidualTests(fit12gs, lagsAC=c(1, 3), lagsCH=1:2,
  nsimu=1, printRes=FALSE)

# GMAR(p=2, M=2) model such that the second AR coefficient of the
# second regime is constrained to zero.
constraints <- list(diag(1, ncol=2, nrow=2), as.matrix(c(1, 0)))
fit22c <- fitGSMAR(logVIX, 2, 2, constraints=constraints)
quantileResidualTests(fit22c, lagsAC=c(1, 3), nsimu=1, printRes=FALSE)
```

---

predict.gsmar

*Forecast GMAR, StMAR or G-StMAR process*

---

**Description**

`predict.gsmar` forecasts the specified GMAR, StMAR or G-StMAR process by using the given data to simulate its possible future values. For one-step forecasts using the exact formula of conditional mean is supported.

**Usage**

```
## S3 method for class 'gsmar'
predict(object, ..., n_ahead, nsimu = 10000,
  pi = c(0.95, 0.8), pred_type = c("median", "mean", "cond_mean"),
  pi_type = c("two-sided", "upper", "lower", "none"), nt,
  plotRes = TRUE)
```

**Arguments**

object	object of class 'gsmar' created with function fitGSMAR or GSMAR.
...	additional arguments passed to grid (ignored if plot_res==FALSE).
n_ahead	a positive integer specifying how many steps in the future should be forecasted.
nsimu	a positive integer specifying to how many simulations the forecast should be based on.
pi	a numeric vector specifying the confidence levels of the prediction intervals.
pred_type	should the prediction be based on sample "mean" or "median"? Or should it be one-step-ahead forecast based on conditional mean ("cond_mean")? prediction intervals won't be calculated if conditional mean is used.
pi_type	should the prediction intervals be "two-sided", "upper" or "lower"?
nt	a positive integer specifying the number of observations to be plotted along with the prediction. Default is round(length(data)*0.2).
plotRes	a logical argument defining whether the forecast should be plotted or not.

**Details**

predict.gsmar uses the last  $p$  values of the data to simulate  $nsimu$  possible future values for each step. The point prediction is then obtained by calculating the sample median (or mean) of each step and the prediction intervals are obtained from the empirical fractiles.

We encourage directly using the function simulateGSMAR for quantile based forecasting. With simulateGSMAR it's easy to forecast the mixing weights too.

**Value**

Returns a data frame containing the empirical point prediction and prediction intervals accordingly to pi. Or if pred\_type=="cond\_mean" returns the optimal prediction as (1x1) numeric vector.

**References**

- Galbraith, R., Galbraith, J. 1974. On the inverses of some patterned matrices arising in the theory of stationary time series. *Journal of Applied Probability* **11**, 63-71.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [econ.EM].
- There are currently no published references for the G-StMAR model, but it's a straightforward generalization with theoretical properties similar to the GMAR and StMAR models.

**See Also**

[simulateGSMAR](#), [condMoments](#), [fitGSMAR](#), [GSMAR](#), [quantileResidualTests](#), [diagnosticPlot](#)

**Examples**

```

# GMAR model
fit12 <- fitGSMAR(data=logVIX, p=1, M=2, model="GMAR")
pred12 <- predict(fit12, n_ahead=10)
pred12

# Restricted GMAR model, one-step conditional mean prediction
fit12r <- fitGSMAR(logVIX, 1, 2, model="GMAR", restricted=TRUE)
pred12r <- predict(fit12r, pred_type="cond_mean", plotRes=FALSE)
pred12r

# Non-mixture StMAR model, upper prediction intervals
fit11t <- fitGSMAR(logVIX, 1, 1, model="StMAR", ncores=1, ncalls=1)
predict(fit11t, n_ahead=10, pi_type="upper", pi=c(0.99, 0.95, 0.9))

# G-StMAR model, no prediction intervals
fit12gs <- fitGSMAR(logVIX, 1, M=c(1, 1), model="G-StMAR")
pred12gs <- predict(fit12gs, n_ahead=10, pred_type="median",
  pi_type="none", plotRes=FALSE)
pred12gs
plot(pred12gs)

# Such StMAR(3,2) that the AR coefficients are restricted to be
# the same for both regimes and that the second AR coefficients are
# constrained to zero.
fit32rc <- fitGSMAR(logVIX, 3, 2, model="StMAR", restricted=TRUE,
  constraints=matrix(c(1, 0, 0, 0, 0, 1), ncol=2))
predict(fit32rc, n_ahead=3, pi_type="lower")

```

---

print.gsmarpred      *print method for class 'gsmarpred' objects*

---

**Description**

print.gsmarpred is print method for object created by predict.gsmar.

**Usage**

```

## S3 method for class 'gsmarpred'
print(x, ..., digits = 2)

```

**Arguments**

x	object of class 'gsmarpred' generated by predict.gsmar.
...	currently not used.
digits	number of decimals to print



## Examples

```
# GMAR-model
params12 <- c(0.18, 0.93, 0.01, 0.86, 0.68, 0.02, 0.88)
gmar12 <- GSMAR(logVIX, 1, 2, params12)
pred <- predict(gmar12, n_ahead=10, plotRes=FALSE)
pred
print(pred, digits=3)
```

---

print.gsmarsum	<i>Print method from objects of class 'gsmarsum'</i>
----------------	--

---

## Description

print.gsmarsum is a print method for object 'gsmarsum' created with summary.gsmar. Approximate standard errors are printed in brackets.

## Usage

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

## Arguments

x	object of class 'gsmarsum' generated by summary.gsmar.
...	currently not used.
digits	number of digits to be printed

## Examples

```
# GMAR model
fit12 <- fitGSMAR(VIX, 1, 2, nCalls=4)
gsmarsum12 <- summary(fit12)
gsmarsum12
print(gsmarsum12, digits=4)
```

---

quantileResidualPlot *Plot quantile residual time series and kernel density*

---

### Description

quantileResidualsPlot plots quantile residual time series and histogram.

### Usage

```
quantileResidualPlot(gsmar)
```

### Arguments

gsmar                    object of class 'gsmar' created with the function fitGSMAR or GSMAR.

### Value

Only plots to a graphical device and doesn't return anything.

### Suggested packages

Install the suggested package "gsl" for faster evaluation of the quantile residuals of StMAR and G-StMAR models.

### References

- Galbraith, R., Galbraith, J. 1974. On the inverses of some patterned matrices arising in the theory of stationary time series. *Journal of Applied Probability* **11**, 63-71.
- Kalliovirta L. (2012) Misspecification tests based on quantile residuals. *The Econometrics Journal*, **15**, 358-393.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for G-StMAR model, but it's a straight forward generalization with theoretical properties similar to GMAR and StMAR models.

### See Also

[diagnosticPlot](#), [fitGSMAR](#), [GSMAR](#), [quantileResidualTests](#), [simulateGSMAR](#)

**Examples**

```
# GMAR model
fit12 <- fitGSMAR(data=logVIX, p=1, M=2, model="GMAR")
quantileResidualPlot(fit12)

# Non-mixture version of StMAR model
fit11t <- fitGSMAR(logVIX, 1, 1, model="StMAR", ncores=1, ncalls=1)
quantileResidualPlot(fit11t)

# Restricted G-StMAR-model
fit12gsr <- fitGSMAR(logVIX, 1, M=c(1, 1), model="G-StMAR",
  restricted=TRUE)
quantileResidualPlot(fit12gsr)

# Such StMAR(3,2) that the AR coefficients are restricted to be
# the same for both regimes and that the second AR coefficients are
# constrained to zero.
fit32rc <- fitGSMAR(logVIX, 3, 2, model="StMAR", restricted=TRUE,
  constraints=matrix(c(1, 0, 0, 0, 0, 1), ncol=2))
quantileResidualPlot(fit32rc)
```

---

quantileResiduals      *Compute quantile residuals of GMAR, StMAR or G-StMAR model*

---

**Description**

quantileResiduals computes the quantile residuals of the specified GMAR, StMAR or G-StMAR model.

**Usage**

```
quantileResiduals(data, p, M, params, model = c("GMAR", "StMAR",
  "G-StMAR"), restricted = FALSE, constraints = NULL,
  parametrization = c("intercept", "mean"))
```

**Arguments**

data	a numeric vector class 'ts' object containing the data. NA values are not supported.
p	a positive integer specifying the order of AR coefficients.
M	<b>For GMAR and StMAR models:</b> a positive integer specifying the number of mixture components. <b>For G-StMAR model:</b> a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is $M=M1+M2$ .

params

a real valued parameter vector specifying the model.

**For non-restricted models: For GMAR model:** Size  $(M(p+3) - 1 \times 1)$  vector  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .

**For StMAR model:** Size  $(M(p+4) - 1 \times 1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(M(p+3) + M2 - 1 \times 1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .

**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices  $\mathbf{C}$  that satisfy  $\phi_m = R_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models: For GMAR model:** Size  $(3M + p - 1 \times 1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .

**For StMAR model:** Size  $(4M + p - 1 \times 1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(3M + M2 + p - 1 \times 1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .

**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix  $C$  that satisfies  $\phi = R\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.

model

is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.

restricted

a logical argument stating whether the AR coefficients  $\phi_{m,1}, \dots, \phi_{m,p}$  are restricted to be the same for all regimes.

constraints

specifies linear constraints applied to the autoregressive parameters.

**For non-restricted models:** a list of size  $(pxq_m)$  constraint matrices  $C_m$  of full column rank satisfying  $\phi_m = C_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$  and  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models:** a size  $(pxq)$  constraint matrix  $C$  of full column rank satisfying  $\phi = C\psi$ , where  $\phi = (\phi_1, \dots, \phi_p)$  and  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always  $p$  for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

parametrization

is the model parametrized with the "intercepts"  $\phi_{m,0}$  or "means"  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ ?

## Value

Returns a  $(Tx1)$  numeric vector containing the quantile residuals associated with the specified GMAR, StMAR or G-StMAR model.

### Suggested packages

Install the suggested package "gsl" for faster evaluation of the quantile residuals of StMAR and G-StMAR models.

### References

- Galbraith, R., Galbraith, J. 1974. On the inverses of some patterned matrices arising in the theory of stationary time series. *Journal of Applied Probability* **11**, 63-71.
- Kalliovirta L. (2012) Misspecification tests based on quantile residuals. *The Econometrics Journal*, **15**, 358-393.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for G-StMAR model, but it's a straight forward generalization with theoretical properties similar to GMAR and StMAR models.

### Examples

```
# GMAR model
params12 <- c(0.18, 0.93, 0.01, 0.86, 0.68, 0.02, 0.88)
quantileResiduals(logVIX, 1, 2, params12)

# Restricted GMAR model, outside parameter space
params12r <- c(0.21, 0.23, 0.92, 0.01, 0.02, 0.86)
quantileResiduals(logVIX, 1, 2, params12r, restricted=TRUE)

# Non-mixture version of StMAR model, outside parameter space
params11t <- c(0.16, 0.93, 0.01, 3.01)
quantileResiduals(logVIX, 1, 1, params11t, model="StMAR")

# G-StMAR model
params12gs <- c(0.86, 0.68, 0.02, 0.18, 0.93, 0.01, 0.11, 44.36)
quantileResiduals(logVIX, 1, c(1, 1), params12gs, model="G-StMAR")

# Restricted G-StMAR model
params12gsr <- c(0.31, 0.33, 0.88, 0.01, 0.02, 0.77, 2.72)
quantileResiduals(logVIX, 1, c(1, 1), params12gsr, model="G-StMAR",
  restricted=TRUE)

# GMAR model as a mixture of AR(2) and AR(1) models
constraints <- list(diag(1, ncol=2, nrow=2), as.matrix(c(1, 0)))
params22c <- c(0.61, 0.83, -0.06, 0.02, 0.21, 0.91, 0.01, 0.16)
quantileResiduals(logVIX, 2, 2, params22c, constraints=constraints)

# Such StMAR(3,2) that the AR coefficients are restricted to be
# the same for both regimes and that the second AR coefficients are
# constrained to zero.
params32trc <- c(0.35, 0.33, 0.88, -0.02, 0.01, 0.01, 0.36, 4.53, 1000)
quantileResiduals(logVIX, 3, 2, params32trc, model="StMAR",
```

```
restricted=TRUE, constraints=matrix(c(1, 0, 0, 0, 0, 1), ncol=2))
```

---

quantileResiduals\_int *Compute quantile residuals of GMAR, StMAR or G-StMAR model*

---

## Description

quantileResiduals\_int computes the quantile residuals of the specified GMAR, StMAR or G-StMAR model.

## Usage

```
quantileResiduals_int(data, p, M, params, model = c("GMAR", "StMAR",
  "G-StMAR"), restricted = FALSE, constraints = NULL,
  parametrization = c("intercept", "mean"))
```

## Arguments

**data** a numeric vector class 'ts' object containing the data. NA values are not supported.

**p** a positive integer specifying the order of AR coefficients.

**M** **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.

**For G-StMAR model:** a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is M=M1+M2.

**params** a real valued parameter vector specifying the model.

**For non-restricted models: For GMAR model:** Size  $(M(p+3) - 1x1)$  vector  $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .

**For StMAR model:** Size  $(M(p+4) - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(M(p+3) + M2 - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .

**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices **C** that satisfy  $\phi_m = R_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models: For GMAR model:** Size  $(3M + p - 1x1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .

**For StMAR model:** Size  $(4M + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(3M + M2 + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .

**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix **C** that satisfies  $\phi = R\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0}/(1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.

model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	specifies linear constraints applied to the autoregressive parameters. <b>For non-restricted models:</b> a list of size $(pxq_m)$ constraint matrices $C_m$ of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$ , where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ . <b>For restricted models:</b> a size $(pxq)$ constraint matrix $C$ of full column rank satisfying $\phi = C\psi$ , where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$ . Symbol $\phi$ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is <b>not</b> desired.
parametrization	is the model parametrized with the "intercepts" $\phi_{m,0}$ or "means" $\mu_m = \phi_{m,0}/(1 - \sum \phi_{i,m})$ ?

**Value**

Returns a  $(Tx1)$  numeric vector containing the quantile residuals associated with the specified GMAR, StMAR or G-StMAR model.

**Suggested packages**

Install the suggested package "gsl" for faster evaluation of the quantile residuals of StMAR and G-StMAR models.

**References**

- Galbraith, R., Galbraith, J. 1974. On the inverses of some patterned matrices arising in the theory of stationary time series. *Journal of Applied Probability* **11**, 63-71.
- Kalliovirta L. (2012) Misspecification tests based on quantile residuals. *The Econometrics Journal*, **15**, 358-393.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [**econ.EM**].
- There are currently no published references for G-StMAR model, but it's a straight forward generalization with theoretical properties similar to GMAR and StMAR models.

---

randomIndividual	<i>Create random GMAR, StMAR or G-StMAR model compatible parameter vector</i>
------------------	---

---

### Description

randomIndividual creates a random GMAR, StMAR, G-StMAR model compatible mean-parametrized parameter vector.

smartIndividual creates a random GMAR, StMAR or G-StMAR model compatible parameter vector close to argument params. Sometimes returns exactly the given parameter vector.

### Usage

```
randomIndividual(p, M, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL, meanscale, sigmascale,
  forcestat = FALSE)
```

```
smartIndividual(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL, meanscale, sigmascale,
  accuracy, whichRandom = numeric(0), forcestat = FALSE)
```

### Arguments

p	a positive integer specifying the order of AR coefficients.
M	<p><b>For GMAR and StMAR models:</b> a positive integer specifying the number of mixture components.</p> <p><b>For G-StMAR model:</b> a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is M=M1+M2.</p>
model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	<p>specifies linear constraints applied to the autoregressive parameters.</p> <p><b>For non-restricted models:</b> a list of size <math>(pxq_m)</math> constraint matrices <math>C_m</math> of full column rank satisfying <math>\phi_m = C_m \psi_m</math> for all <math>m = 1, \dots, M</math>, where <math>\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})</math> and <math>\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})</math>.</p> <p><b>For restricted models:</b> a size <math>(pxq)</math> constraint matrix <math>C</math> of full column rank satisfying <math>\phi = C\psi</math>, where <math>\phi = (\phi_1, \dots, \phi_p)</math> and <math>\psi = (\psi_1, \dots, \psi_q)</math>.</p> <p>Symbol <math>\phi</math> denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is <b>not</b> desired.</p>



meanscale	a real valued vector of length two specifying the mean (the first element) and standard deviation (the second element) of the normal distribution from which the $\phi_{m,0}$ or $\mu_m$ (depending on the desired parametrization) parameters (for random regimes) should be generated.
sigmascale	a positive real number specifying the standard deviation of the (zero mean, positive only) normal distribution from which the component variance parameters (for random regimes) should be generated.
forcestat	use the algorithm by Monahan (1984) to force stationarity on the AR parameters (slower) for random regimes? Not supported for constrained models.
params	a real valued parameter vector specifying the model.

**For non-restricted models: For GMAR model:** Size  $(M(p+3) - 1 \times 1)$  vector  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .

**For StMAR model:** Size  $(M(p+4) - 1 \times 1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(M(p+3) + M2 - 1 \times 1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .

**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices  $\mathbf{C}$  that satisfy  $\phi_m = R_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models: For GMAR model:** Size  $(3M + p - 1 \times 1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .

**For StMAR model:** Size  $(4M + p - 1 \times 1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(3M + M2 + p - 1 \times 1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .

**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix  $C$  that satisfies  $\phi = R\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.

accuracy	a real number larger than zero specifying how close to params the generated parameter vector should be. Standard deviation of the normal distribution from which new parameter values are drawn from will be corresponding parameter value divided by accuracy.
whichRandom	an (optional) numeric vector of max length M specifying which regimes should be random instead of "smart" when using smartIndividual. Does not affect on mixing weight parameters. Default in none.

## Details

The functions can be used for example to create initial populations for the genetic algorithm. Mean-parametrization (instead of intercept terms  $\phi_{m,0}$ ) is assumed.

**Value**

Returns estimated parameter vector described in `initpop`.

**References**

- Monahan J.F. 1984. A Note on Enforcing Stationarity in Autoregressive-Moving Average Models. *Biometrika* **71**, 403-404.

**Examples**

```
# GMAR model parameter vector
params22 <- randomIndividual(2, 2, meanscale=c(0, 1), sigmascale=1)
smart22 <- smartIndividual(2, 2, params22, accuracy=10)
cbind(params22, smart22)

# Restricted GMAR parameter vector
params12r <- randomIndividual(1, 2, restricted=TRUE, meanscale=c(-2, 2), sigmascale=2)
smart12r <- smartIndividual(1, 2, params12r, restricted=TRUE, accuracy=20)
cbind(params12r, smart12r)

# StMAR parameter vector: first regime is random in the "smart individual"
params13t <- randomIndividual(1, 3, model="StMAR", meanscale=c(3, 1), sigmascale=3)
smart13t <- smartIndividual(1, 3, params13t, model="StMAR", accuracy=15,
                           meanscale=c(3, 3), sigmascale=3, whichRandom=1)
cbind(params13t, smart13t)

# Restricted StMAR parameter vector
params22tr <- randomIndividual(2, 2, model="StMAR", restricted=TRUE,
                              meanscale=c(3, 2), sigmascale=0.5)
smart22tr <- smartIndividual(2, 2, params22tr, model="StMAR", restricted=TRUE,
                             accuracy=30)
cbind(params22tr, smart22tr)

# G-StMAR parameter vector
params12gs <- randomIndividual(1, c(1, 1), model="G-StMAR", meanscale=c(0, 1),
                              sigmascale=1)
smart12gs <- smartIndividual(1, c(1, 1), params12gs, model="G-StMAR", accuracy=20)
cbind(params12gs, smart12gs)

# Restricted G-StMAR parameter vector
params23gsr <- randomIndividual(2, c(1, 2), model="G-StMAR", restricted=TRUE,
                              meanscale=c(-1, 1), sigmascale=3)
smart23gsr <- smartIndividual(2, c(1, 2), params23gsr, model="G-StMAR", restricted=TRUE,
                              meanscale=c(0, 1), sigmascale=1, accuracy=20, whichRandom=2)
cbind(params23gsr, smart23gsr)
```

```

# GMAR model as a mixture of AR(2) and AR(1) models
C <- list(diag(1, ncol=2, nrow=2), as.matrix(c(1, 0)))
params22c <- randomIndividual(2, 2, constraints=C, meanscale=c(1, 1),
                             sigmascale=1)
smart22c <- smartIndividual(2, 2, params22c, constraints=C, accuracy=10)
cbind(params22c, smart22c)

# Such constrained StMAR(3, 2) model that the second order AR coefficients
# are constrained to zero.
C0 = matrix(c(1, 0, 0, 0, 0, 1), ncol=2)
C = list(C0, C0)
params32c <- randomIndividual(3, 2, model="StMAR", constraints=C,
                             meanscale=c(1, 1), sigmascale=1)
smart32c <- smartIndividual(3, 2, params32c, model="StMAR", constraints=C, accuracy=10)
cbind(params32c, smart32c)

# Such StMAR(3,2) that the AR coefficients are restricted to be
# the same for both regimes and that the second AR coefficients are
# constrained to zero. Second regime is random in the "smart individual".
params32trc <- randomIndividual(3, 2, model="StMAR", restricted=TRUE,
                              constraints=matrix(c(1, 0, 0, 0, 0, 1), ncol=2),
                              meanscale=c(-2, 0.5), sigmascale=4)
smart32trc <- smartIndividual(3, 2, params32trc, model="StMAR", restricted=TRUE,
                              constraints=matrix(c(1, 0, 0, 0, 0, 1), ncol=2),
                              meanscale=c(0, 0.1), sigmascale=0.1, whichRandom=2,
                              accuracy=20)
cbind(params32trc, smart32trc)

```

---

randomIndividual\_int    *Create random GMAR, StMAR or G-StMAR model compatible parameter vector*

---

### Description

randomIndividual\_int creates a random GMAR, StMAR or G-StMAR model compatible parameter vector.

smartIndividual\_int creates a random GMAR, StMAR or G-StMAR model compatible parameter vector close to argument params.

### Usage

```
randomIndividual_int(p, M, model = c("GMAR", "StMAR", "G-StMAR"),
                   restricted = FALSE, constraints = NULL, meanscale, sigmascale,
                   forcastat = FALSE)
```

```
smartIndividual_int(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
                  restricted = FALSE, constraints = NULL, meanscale, sigmascale,
                  accuracy, whichRandom, forcastat = FALSE)
```

**Arguments**

- p** a positive integer specifying the order of AR coefficients.
- M** **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.  
**For G-StMAR model:** a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is  $M=M1+M2$ .
- model** is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.
- restricted** a logical argument stating whether the AR coefficients  $\phi_{m,1}, \dots, \phi_{m,p}$  are restricted to be the same for all regimes.
- constraints** specifies linear constraints applied to the autoregressive parameters.  
**For non-restricted models:** a list of size  $(pxq_m)$  constraint matrices  $C_m$  of full column rank satisfying  $\phi_m=C_m\psi_m$  for all  $m = 1, \dots, M$ , where  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$  and  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .  
**For restricted models:** a size  $(pxq)$  constraint matrix  $C$  of full column rank satisfying  $\phi=C\psi$ , where  $\phi = (\phi_1, \dots, \phi_p)$  and  $\psi = \psi_1, \dots, \psi_q$ .  
 Symbol  $\phi$  denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.
- meanscale** a real valued vector of length two specifying the mean (the first element) and standard deviation (the second element) of the normal distribution from which the  $\phi_{m,0}$  or  $\mu_m$  (depending on the desired parametrization) parameters (for random regimes) should be generated.
- sigmascale** a positive real number specifying the standard deviation of the (zero mean, positive only) normal distribution from which the component variance parameters (for random regimes) should be generated.
- forcestat** use the algorithm by Monahan (1984) to force stationarity on the AR parameters (slower) for random regimes? Not supported for constrained models.
- params** a real valued parameter vector specifying the model.  
**For non-restricted models: For GMAR model:** Size  $(M(p+3) - 1x1)$  vector  $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .  
**For StMAR model:** Size  $(M(p+4) - 1x1)$  vector  $(\theta, \nu)=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .  
**For G-StMAR model:** Size  $(M(p+3) + M2 - 1x1)$  vector  $(\theta, \nu)=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .  
**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices  $C$  that satisfy  $\phi_m=R_m\psi_m$  for all  $m = 1, \dots, M$ , where  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .  
**For restricted models: For GMAR model:** Size  $(3M + p - 1x1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .  
**For StMAR model:** Size  $(4M + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ .

**For G-StMAR model:** Size  $(3M+M2+p-1 \times 1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots)$

**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix  $C$  that satisfies  $\phi = R\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If `parametrization=="mean"` just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first  $M1$  components are *GMAR-type* and the rest  $M2$  components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.

accuracy	a real number larger than zero specifying how close to params the generated parameter vector should be. Standard deviation of the normal distribution from which new parameter values are drawn from will be corresponding parameter value divided by accuracy.
whichRandom	an (optional) numeric vector of max length $M$ specifying which regimes should be random instead of "smart" when using <code>smartIndividual</code> . Does not affect on mixing weight parameters. Default in none.

### Value

Returns estimated parameter vector described in `initpop`.

### References

- Monahan J.F. 1984. A Note on Enforcing Stationarity in Autoregressive-Moving Average Models. *Biometrika* **71**, 403-404.

---

random_arcoefs	<i>Create random AR coefficients</i>
----------------	--------------------------------------

---

### Description

random\_arcoefs generates random AR coefficients

### Usage

```
random_arcoefs(p, forcastat = FALSE, sd = 0.6/p)
```

### Arguments

p	a positive integer specifying the order of AR coefficients.
forcastat	use the algorithm by Monahan (1984) to force stationarity on the AR parameters (slower)?
sd	if <code>forcastat==FALSE</code> then AR parameters are drawn from zero mean normal distribution with sd given by this parameter

**Details**

if forcastat==TRUE then the AR coefficients are relatively large, otherwise they are usually relatively small.

**Value**

Returns  $px1$  vector containing random AR coefficients.

**References**

- Monahan J.F. 1984. A Note on Enforcing Stationarity in Autoregressive-Moving Average Models. *Biometrika* **71**, 403-404.

---

random_regime	<i>Create random regime</i>
---------------	-----------------------------

---

**Description**

random\_regime generates random regime parameters

**Usage**

```
random_regime(p, meanscale, sigmascale, restricted = FALSE,
             constraints = NULL, m, forcastat = FALSE)
```

**Arguments**

- |             |  |
|-------------|--|
| p           | a positive integer specifying the order of AR coefficients.  |
| meanscale   | a real valued vector of length two specifying the mean (the first element) and standard deviation (the second element) of the normal distribution from which the $\mu_m$ mean-parameters are generated in random mutations in the genetic algorithm. Default is <code>c(mean(data), sd(data))</code> . Note that the genetic algorithm optimizes with mean-parametrization even when <code>parametrization=="intercept"</code> , but input (in <code>initpop</code> ) and output (return value) parameter vectors may be intercept-parametrized. |
| sigmascale  | a positive real number specifying the standard deviation of the (zero mean, positive only) normal distribution from which the component variance parameters are generated in the random mutations in the genetic algorithm. Default is <code>var(stats::ar(data, order.max=10)\$resid, na.rm=TRUE)</code> .  |
| restricted  | a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.  |
| constraints | specifies linear constraints applied to the autoregressive parameters.   |

**For non-restricted models:** a list of size  $(pxq_m)$  constraint matrices  $C_m$  of full column rank satisfying  $\phi_m = C_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$  and  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models:** a size  $(p \times q)$  constraint matrix  $C$  of full column rank satisfying  $\phi = C\psi$ , where  $\phi = (\phi_1, \dots, \phi_p)$  and  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always  $p$  for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

`m` which regime?  
`forcestat` use the algorithm by Monahan (1984) to force stationarity on the AR parameters (slower)? Not supported for constrained models.

### Details

if `forcestat==TRUE` then the AR coefficients are relatively large, otherwise they are usually relatively small.

### Value

**Regular models:**  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  where  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ .

**Restricted models:** Not supported!

**Constrained models:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$ .

### References

- Monahan J.F. 1984. A Note on Enforcing Stationarity in Autoregressive-Moving Average Models. *Biometrika* **71**, 403-404.

---

reformConstrainedPars *Reform parameter vector with linear constraints to correspond non-constrained parameter vector.*

---

### Description

reformConstrainedPars reforms the parameter vector of a model with linear constraints to the "standard form" so that it's comparable with non-constrained models.

### Usage

```
reformConstrainedPars(p, M, params, model = c("GMAR", "StMAR",
"G-StMAR"), restricted = FALSE, constraints = NULL)
```

### Arguments

`p` a positive integer specifying the order of AR coefficients.  
`M` **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.

	<p><b>For G-StMAR model:</b> a size <math>(2 \times 1)</math> vector specifying the number of <i>GMAR-type</i> components <math>M1</math> in the first element and <i>StMAR-type</i> components <math>M2</math> in the second. The total number of mixture components is <math>M=M1+M2</math>.</p>
params	<p>a real valued parameter vector specifying the model.</p> <p><b>For non-restricted models: For GMAR model:</b> Size <math>(M(p+3) - 1 \times 1)</math> vector <math>\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})</math>, where <math>v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)</math> and <math>\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})</math>, <math>m = 1, \dots, M</math>.</p> <p><b>For StMAR model:</b> Size <math>(M(p+4) - 1 \times 1)</math> vector <math>(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)</math>.</p> <p><b>For G-StMAR model:</b> Size <math>(M(p+3) + M2 - 1 \times 1)</math> vector <math>(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)</math>.</p> <p><b>With linear constraints:</b> Replace the vectors <math>\phi_m</math> with vectors <math>\psi_m</math> and provide a list of constraint matrices <math>C</math> that satisfy <math>\phi_m = R_m \psi_m</math> for all <math>m = 1, \dots, M</math>, where <math>\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})</math>.</p> <p><b>For restricted models: For GMAR model:</b> Size <math>(3M + p - 1 \times 1)</math> vector <math>\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})</math>, where <math>\phi = (\phi_1, \dots, \phi_M)</math>.</p> <p><b>For StMAR model:</b> Size <math>(4M + p - 1 \times 1)</math> vector <math>(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)</math>.</p> <p><b>For G-StMAR model:</b> Size <math>(3M + M2 + p - 1 \times 1)</math> vector <math>(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)</math>.</p> <p><b>With linear constraints:</b> Replace the vector <math>\phi</math> with vector <math>\psi</math> and provide a constraint matrix <math>C</math> that satisfies <math>\phi = R\psi</math>, where <math>\psi = (\psi_1, \dots, \psi_q)</math>.</p> <p>Symbol <math>\phi</math> denotes an AR coefficient, <math>\sigma^2</math> a variance, <math>\alpha</math> a mixing weight and <math>\nu</math> a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term <math>\phi_{m,0}</math> with regimewise mean <math>\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})</math>. In the <b>G-StMAR</b> model the first <math>M1</math> components are <i>GMAR-type</i> and the rest <math>M2</math> components are <i>StMAR-type</i>. Note that in the case <math>M=1</math> the parameter <math>\alpha</math> is dropped, and in the case of <b>StMAR</b> or <b>G-StMAR</b> model the degrees of freedom parameters <math>\nu_m</math> have to be larger than 2.</p>
model	<p>is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first <math>M1</math> components are <i>GMAR-type</i> and the rest <math>M2</math> components are <i>StMAR-type</i>.</p>
restricted	<p>a logical argument stating whether the AR coefficients <math>\phi_{m,1}, \dots, \phi_{m,p}</math> are restricted to be the same for all regimes.</p>
constraints	<p>specifies linear constraints applied to the autoregressive parameters.</p> <p><b>For non-restricted models:</b> a list of size <math>(pxq_m)</math> constraint matrices <math>C_m</math> of full column rank satisfying <math>\phi_m = C_m \psi_m</math> for all <math>m = 1, \dots, M</math>, where <math>\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})</math> and <math>\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})</math>.</p> <p><b>For restricted models:</b> a size <math>(pxq)</math> constraint matrix <math>C</math> of full column rank satisfying <math>\phi = C\psi</math>, where <math>\phi = (\phi_1, \dots, \phi_p)</math> and <math>\psi = (\psi_1, \dots, \psi_q)</math>.</p> <p>Symbol <math>\phi</math> denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always <math>p</math> for all regimes. Ignore or set to NULL if applying linear constraints is <b>not</b> desired.</p>

### Value

Returns such parameter vector corresponding to the input vector that is the form described in params for non-restricted or restricted models (for non-constrained models), and can hence be used just as the parameter vectors of non-constrained models.



---

reformParameters	<i>Reform any parameter vector into standard form.</i>
------------------	--

---

### Description

reformParameters takes a parameter vector of any (non-constrained) GMAR, StMAR or G-StMAR model and returns a list with the parameter vector in the standard form, parameter matrix containing AR coefficients and component variances, mixing weights alphas and in case of StMAR or G-StMAR model also degrees of freedom parameters.

### Usage

```
reformParameters(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE)
```

### Arguments

- p** a positive integer specifying the order of AR coefficients.
- M** **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.  
**For G-StMAR model:** a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is M=M1+M2.
- params** a real valued parameter vector specifying the model.  
**For non-restricted models: For GMAR model:** Size  $(M(p+3) - 1x1)$  vector  $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .  
**For StMAR model:** Size  $(M(p+4) - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .  
**For G-StMAR model:** Size  $(M(p+3) + M2 - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .  
**For restricted models: For GMAR model:** Size  $(3M + p - 1x1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .  
**For StMAR model:** Size  $(4M + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .  
**For G-StMAR model:** Size  $(3M + M2 + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .
- Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.
- model** is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.
- restricted** a logical argument stating whether the AR coefficients  $\phi_{m,1}, \dots, \phi_{m,p}$  are restricted to be the same for all regimes.

**Details**

This function does not support models parametrized with general linear constraints! Nor does it have any argument checks.

**Value**

Returns a list with...

`$params` parameter vector in the standard form.

`$pars` corresponding parameter matrix containing AR coefficients and component variances. First row for `phi0` or means depending on the parametrization. Column for each component.

`$alphas` numeric vector containing mixing weights for all components (also for the last one).

`$dfs` numeric vector containing degrees of freedom parameters for all components. Returned only if `model == "StMAR"` or `model == "G-StMAR"`.

---

`reformRestrictedPars` *Reform parameter vector with restricted autoregressive parameters to correspond non-restricted parameter vector.*

---

**Description**

`reformRestrictedPars` reforms parameter vector with restricted autoregressive parameters to correspond non-restricted parameter vector.

**Usage**

```
reformRestrictedPars(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE)
```

**Arguments**

`p` a positive integer specifying the order of AR coefficients.

`M` **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.

**For G-StMAR model:** a size  $(2 \times 1)$  vector specifying the number of *GMAR-type* components `M1` in the first element and *StMAR-type* components `M2` in the second. The total number of mixture components is  $M = M1 + M2$ .

`params` a real valued parameter vector specifying the model.

**For non-restricted models: For GMAR model:** Size  $(M(p+3) - 1 \times 1)$  vector  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .

**For StMAR model:** Size  $(M(p+4) - 1 \times 1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(M(p+3) + M2 - 1 \times 1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .

**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices  $\mathbf{C}$  that satisfy  $\phi_m = R_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models: For GMAR model:** Size  $(3M + p - 1 \times 1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .

**For StMAR model:** Size  $(4M + p - 1 \times 1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(3M + M2 + p - 1 \times 1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_{M2})$ .

**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix  $C$  that satisfies  $\phi = R\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If parametrization="mean" just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.

model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.

**Value**

Returns such parameter vector corresponding to the input vector that is the form described in params for non-restricted models (for non-constrained models). Linear constraints are not supported.

---

regime_distance	<i>Calculate "distance" between two regimes</i>
-----------------	---

---

**Description**

regime\_distance scales each regime parameter and then calculates distance between scaled regime\_pars1 and regime\_pars2.

**Usage**

```
regime_distance(regime_pars1, regime_pars2)
```

**Arguments**

- regime\_pars1    a numeric vector representing a regime.
- regime\_pars2    a numeric vector representing another regime, same length as regime\_pars1

**Value**

Returns "distance" between regime\_pars1 and regime\_pars2. Values are scaled before calculating the "distance". Read the source code for details.

---

removeAllConstraints    *Transform constrained and restricted parameter vector into the regular form*

---

**Description**

removeAllConstraints transforms constrained and restricted parameter vector into the regular form.

**Usage**

```
removeAllConstraints(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL)
```

**Arguments**

p                    a positive integer specifying the order of AR coefficients.

M                    **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.

**For G-StMAR model:** a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is M=M1+M2.

params              a real valued parameter vector specifying the model.

**For non-restricted models: For GMAR model:** Size  $(M(p+3) - 1x1)$  vector  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .

**For StMAR model:** Size  $(M(p+4) - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(M(p+3) + M2 - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .

**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices **C** that satisfy  $\phi_m = R_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models: For GMAR model:** Size  $(3M + p - 1x1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .

**For StMAR model:** Size  $(4M + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(3M + M2 + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .

**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix **C** that satisfies  $\phi = R\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0}/(1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.

model is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.

restricted a logical argument stating whether the AR coefficients  $\phi_{m,1}, \dots, \phi_{m,p}$  are restricted to be the same for all regimes.

constraints specifies linear constraints applied to the autoregressive parameters.

**For non-restricted models:** a list of size  $(pxq_m)$  constraint matrices  $C_m$  of full column rank satisfying  $\phi_m = C_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$  and  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models:** a size  $(pxq)$  constraint matrix  $C$  of full column rank satisfying  $\phi = C\psi$ , where  $\phi = (\phi_1, \dots, \phi_p)$  and  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always  $p$  for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

**Value**

Returns such parameter vector corresponding to the input vector that is the form described in params for non-restricted and non-constrained models.

---

simulateGSMAR	<i>Simulate values from GMAR, StMAR or G-StMAR process</i>
---------------	--

---

**Description**

simulateGSMAR simulates values from the specified GMAR, StMAR or G-StMAR process. Can be used for forecasting future values of the process.

**Usage**

```
simulateGSMAR(gsmar, nsimu, initvalues, ntimes = 1, drop = TRUE)
```

**Arguments**

gsmar object of class 'gsmar' created with the function fitGSMAR or GSMAR.

nsimu a positive integer specifying how many values will be simulated.

initvalues	a numeric vector with length $\geq p$ specifying the initial values for the simulation. The <b>last</b> element will be used as the initial value for the first lag, the second last element will be initial value for the second lag, etc. If not specified, initial values will be simulated from the process's stationary distribution.
ntimes	a positive integer specifying how many sets of simulations should be performed.
drop	if TRUE (default) then the components of the returned list are coerced to lower dimension if <code>ntimes==1</code> , i.e., <code>\$sample</code> and <code>\$component</code> will be vectors and <code>\$mixing_weights</code> will be matrix.

### Details

The argument `ntimes` is intended for forecasting: a GSMAR process can be forecasted by simulating it's possible future values. One can easily perform a large number simulations and calculate the sample quantiles from the simulated values to obtain prediction intervals (see the forecasting example).

### Value

If `drop==TRUE` and `ntimes==1` (default): `$sample` and `$component` are vectors and `$mixing_weights` is matrix. Otherwise, returns a list with...

`$sample` a size  $(n_{\text{sim}} \times n_{\text{times}})$  matrix containing the simulated values.

`$component` a size  $(n_{\text{sim}} \times n_{\text{times}})$  matrix containing the information from which component each value was generated from.

`$mixing_weights` a size  $(n_{\text{sim}} \times M \times n_{\text{times}})$  array containing the mixing weights corresponding to the sample: the dimension  $[i, , ]$  is the time index, the dimension  $[ , i, ]$  indicates the regime, and the dimension  $[ , , i]$  indicates the  $i$ :th set of simulations.

### References

- Galbraith, R., Galbraith, J. 1974. On the inverses of some patterned matrices arising in the theory of stationary time series. *Journal of Applied Probability* **11**, 63-71.
- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [econ.EM].
- There are currently no published references for the G-StMAR model, but it's a straightforward generalization with theoretical properties similar to the GMAR and StMAR models.

### See Also

[fitGSMAR](#), [GSMAR](#), [predict.gsmar](#), [add\\_data](#), [condMoments](#), [mixingWeights](#)

**Examples**

```

# GMAR model
params13 <- c(1.4, 0.88, 0.26, 2.46, 0.82, 0.74, 5.0, 0.68, 5.2, 0.72, 0.2)
gmar13 <- GSMAR(p=1, M=3, params=params13, model="GMAR")
sim13 <- simulateGSMAR(gmar13, nsimu=500)
ts.plot(sim13$sample)
ts.plot(sim13$component)
ts.plot(sim13$mixing_weights, col=rainbow(3), lty=2)

# FORECASTING EXAMPLE:
# Restricted GMAR model, 10000 sets of simulations with initial values 6 and 6.2.
params22r <- c(1.4, 1.8, 0.8, -0.1, 0.29, 3.18, 0.84)
gmar22r <- GSMAR(p=2, M=2, params=params22r, model="GMAR",
  restricted=TRUE)
sim22r <- simulateGSMAR(gmar22r, nsimu=5, initval=c(6, 6.2), ntimes=10000)
apply(sim22r$sample, 1, median) # Point forecast
apply(sim22r$sample, 1, quantile, probs=c(0.025, 0.975)) # 95% interval
apply(sim22r$mixing_weights, MARGIN=1:2, FUN=median) # mix.weight point forecast
apply(sim22r$mixing_weights, MARGIN=1:2, FUN=quantile,
  probs=c(0.025, 0.975)) # mix.weight 95% intervals

# G-StMAR model, with initial values
params12gs <- c(1.38, 0.88, 0.27, 3.8, 0.74, 3.15, 0.8, 3.6)
gstmar12 <- GSMAR(p=1, M=c(1, 1), params=params12gs,
  model="G-StMAR")
sim12gs <- simulateGSMAR(gstmar12, nsimu=500, initvalues=5:6)
ts.plot(sim12gs$sample)
ts.plot(sim12gs$component)
ts.plot(sim12gs$mixing_weights, col=rainbow(3), lty=2)

# Mixture version of Heterogeneous Autoregressive (HAR) model (without data)
paramsHAR <- c(1, 0.1, 0.2, 0.3, 1, 1, 0.15, 0.25, 0.35, 1, 0.55)
r1 = c(1, rep(0, 21)); r2 = c(rep(0.2, 5), rep(0, 17)); r3 = rep(1/22, 22)
R0 = cbind(r1, r2, r3)
mixhar <- GSMAR(p=22, M=2, params=paramsHAR, model="GMAR", constraints=list(R0, R0))
simhar <- simulateGSMAR(mixhar, nsimu=1000)
ts.plot(simhar$sample)
ts.plot(simhar$component)
ts.plot(simhar$mixing_weights, col=rainbow(3), lty=2)

```

---

 sortComponents

*Sort the mixture components of GMAR, StMAR or G-StMAR model*


---

**Description**

sortComponents sorts mixture components of the specified GMAR, StMAR or G-StMAR model by the mixing weights when the parameter vector is in the "standard form" for restricted or non-restricted models.

**Usage**

```
sortComponents(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE)
```

**Arguments**

- p** a positive integer specifying the order of AR coefficients.
- M** **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.  
**For G-StMAR model:** a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is  $M=M1+M2$ .
- params** a real valued parameter vector specifying the model.  
**For non-restricted models: For GMAR model:** Size  $(M(p+3) - 1x1)$  vector  $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .  
**For StMAR model:** Size  $(M(p+4) - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .  
**For G-StMAR model:** Size  $(M(p+3) + M2 - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .  
**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices **C** that satisfy  $\phi_m = R_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .  
**For restricted models: For GMAR model:** Size  $(3M + p - 1x1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .  
**For StMAR model:** Size  $(4M + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .  
**For G-StMAR model:** Size  $(3M + M2 + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .  
**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix **C** that satisfies  $\phi = R\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .
- Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If `parametrization=="mean"` just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.
- model** is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*.
- restricted** a logical argument stating whether the AR coefficients  $\phi_{m,1}, \dots, \phi_{m,p}$  are restricted to be the same for all regimes.

**Details**

This function does not support models parametrized with general linear constraints!

Models with general linear constraints are not supported.



**Value**

Returns a parameter vector sorted by it's mixing weights, described in params.

---

standardErrors	<i>Calculate standard errors for estimates of GMAR, StMAR or GStMAR model</i>
----------------	---

---

**Description**

standardErrors numerically approximates standard errors for the given estimates of GMAR, StMAR or GStMAR model

**Usage**

```
standardErrors(data, p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL, conditional = TRUE,
  parametrization = c("intercept", "mean"), custom_h = NULL, minval)
```

**Arguments**

**data** a numeric vector class 'ts' object containing the data. NA values are not supported.

**p** a positive integer specifying the order of AR coefficients.

**M** **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.

**For G-StMAR model:** a size (2x1) vector specifying the number of *GMAR-type* components M1 in the first element and *StMAR-type* components M2 in the second. The total number of mixture components is M=M1+M2.

**params** a real valued parameter vector specifying the model.

**For non-restricted models: For GMAR model:** Size  $(M(p+3) - 1x1)$  vector  $\theta = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .

**For StMAR model:** Size  $(M(p+4) - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(M(p+3) + M2 - 1x1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .

**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices **C** that satisfy  $\phi_m = R_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models: For GMAR model:** Size  $(3M + p - 1x1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .

**For StMAR model:** Size  $(4M + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .

**For G-StMAR model:** Size  $(3M + M2 + p - 1x1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .

**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix **C** that satisfies  $\phi = R\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If parametrization=="mean" just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0}/(1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.

model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	specifies linear constraints applied to the autoregressive parameters. <b>For non-restricted models:</b> a list of size $(pxq_m)$ constraint matrices $C_m$ of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$ , where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ . <b>For restricted models:</b> a size $(pxq)$ constraint matrix $C$ of full column rank satisfying $\phi = C\psi$ , where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$ . Symbol $\phi$ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always $p$ for all regimes. Ignore or set to NULL if applying linear constraints is <b>not</b> desired.
conditional parametrization	a logical argument specifying whether the conditional or exact log-likelihood function should be used.
custom_h	is the model parametrized with the "intercepts" $\phi_{m,0}$ or "means" $\mu_m = \phi_{m,0}/(1 - \sum \phi_{i,m})$ ? a numeric vector with the same length as params specifying the difference 'h' used in finite difference approximation for each parameter separately. If NULL (default), then the difference used for differentiating overly large degrees of freedom parameters is adjusted to avoid numerical problems, and the difference is 6e-6 for the other parameters.
minval	this will be returned when the parameter vector is outside the parameter space.

**Value**

Approximate standard errors of the parameter values

---

stmarpars_to_gstmar	<i>Transform a StMAR model parameter vector to a corresponding G-StMAR model parameter vector with large dfs parameters reduced.</i>
---------------------	--

---

**Description**

stmarpars\_to\_gstmar transforms a StMAR model parameter vector to a corresponding G-StMAR model parameter vector with large dfs parameters reduced by turning the related regimes GMAR type.

**Usage**

```
stmarpars_to_gstmar(p, M, params, restricted = FALSE,
  constraints = NULL, maxdf = 100)
```

**Arguments**

- p** a positive integer specifying the order of AR coefficients.
- M** **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.  
**For G-StMAR model:** a size  $(2 \times 1)$  vector specifying the number of *GMAR-type* components  $M1$  in the first element and *StMAR-type* components  $M2$  in the second. The total number of mixture components is  $M=M1+M2$ .
- params** a real valued parameter vector specifying the model.  
**For non-restricted models: For GMAR model:** Size  $(M(p+3) - 1 \times 1)$  vector  $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .  
**For StMAR model:** Size  $(M(p+4) - 1 \times 1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .  
**For G-StMAR model:** Size  $(M(p+3) + M2 - 1 \times 1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .  
**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices  $C$  that satisfy  $\phi_m = R_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .  
**For restricted models: For GMAR model:** Size  $(3M + p - 1 \times 1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .  
**For StMAR model:** Size  $(4M + p - 1 \times 1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .  
**For G-StMAR model:** Size  $(3M + M2 + p - 1 \times 1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)$ .  
**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix  $C$  that satisfies  $\phi = R\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .
- Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If `parametrization=="mean"` just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first  $M1$  components are *GMAR-type* and the rest  $M2$  components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.
- restricted** a logical argument stating whether the AR coefficients  $\phi_{m,1}, \dots, \phi_{m,p}$  are restricted to be the same for all regimes.
- constraints** specifies linear constraints applied to the autoregressive parameters.  
**For non-restricted models:** a list of size  $(p \times q_m)$  constraint matrices  $C_m$  of full column rank satisfying  $\phi_m = C_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$  and  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .  
**For restricted models:** a size  $(p \times q)$  constraint matrix  $C$  of full column rank satisfying  $\phi = C\psi$ , where  $\phi = (\phi_1, \dots, \phi_p)$  and  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always  $p$  for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

maxdf regimes with degrees of freedom parameter value large than this will be turned into GMAR type.

### Value

Returns a list with three elements: \$params contains the corresponding G-StMAR model parameter vector, \$reg\_order contains the permutation that was applied to the regimes (GMAR type components first, and decreasing ordering by mixign weight parameters), and \$M a vector of length two containing the number of GMAR type regimes in the first element and the number of StMAR type components in the second.

### Examples

```
params12 <- c(2, 0.9, 0.1, 0.8, 0.5, 0.5, 0.4, 12, 300)
stmarpars_to_gstmar(1, 2, params12, maxdf=100)
```

---

stmar_to_gstmar	<i>Estimate a G-StMAR model based on StMAR model with large degrees of freedom parameters</i>
-----------------	---

---

### Description

stmar\_to\_gstmar estimates a G-StMAR model based on StMAR model with large degree of freedom parameters

### Usage

```
stmar_to_gstmar(gsmar, maxdf = 100, estimate, calc_std_errors,
  maxit = 100, custom_h = NULL)
```

### Arguments

gsmar	object of class 'gsmar' created with the function fitGSMAR or GSMAR.
maxdf	regimes with degrees of freedom parameter value large than this will be turned into GMAR type.
estimate	set TRUE if the new model should be estimated with variable metric algorithm using the StMAR model parameters as the initial values. By default TRUE iff the model contains data.
calc_std_errors	set TRUE if the approximate standard errors should be calculated. By default TRUE iff the model contains data.
maxit	the maximum number of iterations for the variable metric algorithm. Ignored if estimate==FALSE.

`custom_h` A numeric vector of with same length as the parameter vector of the estimated model:  $i$ :th element of `custom_h` is the difference used in central difference approximation for differentials of the log-likelihood function for the  $i$ :th parameter. If NULL (default), then the difference used for differentiating overly large degrees of freedom parameters is adjusted to avoid numerical problems, and the difference is  $6e-6$  for the other parameters.

### Details

If a StMAR model contains large estimates for the degrees of freedom parameters one should consider switching to the corresponding G-StMAR model that lets the corresponding regimes to be GMAR type. `stmar_to_gstmar` makes it convenient to do this switch.

### Value

Returns an object of class 'gsmar' defining the specified GMAR, StMAR or G-StMAR model. If data is supplied, the returned object contains (by default) empirical mixing weights, conditional means and variances and quantile residuals. Note that the first  $p$  observations are taken as the initial values so mixing weights, conditional moments and qresiduals start from the  $p+1$ :th observation (interpreted as  $t=1$ ).

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's  $t$ -distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for the G-StMAR model, but it's a straightforward generalization with theoretical properties similar to the GMAR and StMAR models.

### See Also

[fitGSMAR](#), [GSMAR](#), [iterate\\_more](#), [get\\_gradient](#), [get\\_regime\\_means](#), [swap\\_parametrization](#), [stmar\\_to\\_gstmar](#)

### Examples

```
# These are long running examples and use parallel computing
fit13tr <- fitGSMAR(logVIX, 1, 3, model="StMAR", restricted=TRUE,
  ncalls=1, seeds=1)
fit13tr
fit13gsr <- stmar_to_gstmar(fit13tr)
fit13gsr
```

---

swap\_parametrization *Swap the parametrization of object of class 'gsmar' defining a gsmar model*

---

### Description

swap\_parametrization swaps the parametrization of object of class 'gsmar' to "mean" if the current parametrization is "intercept", and vice versa.

### Usage

```
swap_parametrization(gsmar, calc_std_errors = TRUE, custom_h = NULL)
```

### Arguments

gsmar	object of class 'gsmar' created with the function fitGSMAR or GSMAR.
calc_std_errors	should approximate standard errors be calculated?
custom_h	A numeric vector of with same length as the parameter vector of the estimated model: i:th element of custom_h is the difference used in central difference approximation for differentials of the log-likelihood function for the i:th parameter. If NULL (default), then the difference used for differentiating overly large degrees of freedom parameters is adjusted to avoid numerical problems, and the difference is 6e-6 for the other parameters.

### Details

swap\_parametrization is convenient tool if you have estimated the model in "intercept"-parametrization, but wish to work with "mean"-parametrization in the future, or vice versa. In gsmarkit, for example the approximate standard errors are only available for parametrized parameters.

### Value

Returns an object of class 'gsmar' defining the specified GMAR, StMAR or G-StMAR model. If data is supplied, the returned object contains (by default) empirical mixing weights, conditional means and variances and quantile residuals. Note that the first p observations are taken as the initial values so mixing weights, conditional moments and qresiduals start from the p+1:th observation (interpreted as t=1).

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for the G-StMAR model, but it's a straightforward generalization with theoretical properties similar to the GMAR and StMAR models.

**See Also**

[fitGSMAR](#), [GSMAR](#), [iterate\\_more](#), [get\\_gradient](#), [get\\_regime\\_means](#), [swap\\_parametrization](#), [stmar\\_to\\_gstmar](#)

**Examples**

```
# GMAR model with intercept parametrization
params12 <- c(0.18, 0.93, 0.01, 0.86, 0.68, 0.02, 0.88)
gmar12 <- GSMAR(data=logVIX, p=1, M=2, params=params12, model="GMAR")
gmar12

# Swap to mean parametrization
gmar12 <- swap_parametrization(gmar12)
gmar12
```

uGMAR

*uGMAR: Estimate Univariate Gaussian or Student's t Mixture Autoregressive Model*

**Description**

uGMAR is a package for estimating univariate Gaussian Mixture Autoregressive (GMAR), Student's t Mixture Autoregressive (StMAR) and Gaussian and Student's t Mixture Autoregressive (G-StMAR) models. It provides tool for quantile residuals tests, graphical diagnostics, forecasting and simulation. Applying general linear constraints to the autoregressive parameters, or restricting them to be the same for all regimes is supported.

Many of the functions documented are not exported but for internal use only. The readme file is a good place to start, and the vignette might be useful too.

uncondMoments

*Calculate unconditional mean, variance, first p autocovariances and autocorrelations of the GSMAR process.*

**Description**

uncondMoments calculates the unconditional mean, variance, first p autocovariances and autocorrelations of the GSMAR process.

**Usage**

```
uncondMoments(gsmar)
```

**Arguments**

gsmar                    object of class 'gsmar' created with the function `fitGSMAR` or `GSMAR`.

**Value**

Returns a list containing the unconditional mean, variance, first  $p$  autocovariances and autocorrelations. Note that the lag-zero autocovariance/correlation is not included in the "first  $p$ " but is given in the `uncond_variance` component separately.

**References**

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's t-distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for the G-StMAR model, but it's a straightforward generalization with theoretical properties similar to the GMAR and StMAR models.
- Lutkepohl H. 2005. New Introduction to Multiple Time Series Analysis. *Springer*.

**See Also**

Other moment functions: [condMoments](#), [get\\_regime\\_autocovs](#), [get\\_regime\\_means](#), [get\\_regime\\_vars](#)

**Examples**

```
# GMAR model
params13 <- c(1.4, 0.88, 0.26, 2.46, 0.82, 0.74, 5.0, 0.68, 5.2, 0.72, 0.2)
gmar13 <- GSMAR(p=1, M=3, params=params13, model="GMAR")
uncondMoments(gmar13)

# StMAR model
params12t <- c(1.38, 0.88, 0.27, 3.8, 0.74, 3.15, 0.8, 100, 3.6)
stmar12t <- GSMAR(p=1, M=2, params=params12t, model="StMAR")
uncondMoments(stmar12t)

# G-StMAR model (similar to the StMAR model above)
params12gs <- c(1.38, 0.88, 0.27, 3.8, 0.74, 3.15, 0.8, 3.6)
gstmar12 <- GSMAR(p=1, M=c(1, 1), params=params12gs, model="G-StMAR")
uncondMoments(gstmar12)
```

---

<code>uncondMoments_int</code>	<i>Calculate unconditional mean, variance, first <math>p</math> autocovariances and autocorrelations of the GSMAR process.</i>
--------------------------------	--

---

**Description**

`uncondMoments_int` calculates the unconditional mean, variance, first  $p$  autocovariances and autocorrelations of the GSMAR process.



**Usage**

```
uncondMoments_int(p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
  restricted = FALSE, constraints = NULL,
  parametrization = c("intercept", "mean"))
```

**Arguments**

- p** a positive integer specifying the order of AR coefficients.
- M** **For GMAR and StMAR models:** a positive integer specifying the number of mixture components.  
**For G-StMAR model:** a size  $(2 \times 1)$  vector specifying the number of *GMAR-type* components  $M_1$  in the first element and *StMAR-type* components  $M_2$  in the second. The total number of mixture components is  $M=M_1+M_2$ .
- params** a real valued parameter vector specifying the model.  
**For non-restricted models: For GMAR model:** Size  $(M(p+3) - 1 \times 1)$  vector  $\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})$ , where  $v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)$  and  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ ,  $m = 1, \dots, M$ .  
**For StMAR model:** Size  $(M(p+4) - 1 \times 1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)$ .  
**For G-StMAR model:** Size  $(M(p+3) + M_2 - 1 \times 1)$  vector  $(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M_1+1}, \dots, \nu_M)$ .  
**With linear constraints:** Replace the vectors  $\phi_m$  with vectors  $\psi_m$  and provide a list of constraint matrices **C** that satisfy  $\phi_m = R_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .  
**For restricted models: For GMAR model:** Size  $(3M + p - 1 \times 1)$  vector  $\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})$ , where  $\phi = (\phi_1, \dots, \phi_M)$ .  
**For StMAR model:** Size  $(4M + p - 1 \times 1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu)$ .  
**For G-StMAR model:** Size  $(3M + M_2 + p - 1 \times 1)$  vector  $(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu)$ .  
**With linear constraints:** Replace the vector  $\phi$  with vector  $\psi$  and provide a constraint matrix **C** that satisfies  $\phi = R\psi$ , where  $\psi = (\psi_1, \dots, \psi_q)$ .
- Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If parametrization="mean" just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first  $M_1$  components are *GMAR-type* and the rest  $M_2$  components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.
- model** is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first  $M_1$  components are *GMAR-type* and the rest  $M_2$  components are *StMAR-type*.
- restricted** a logical argument stating whether the AR coefficients  $\phi_{m,1}, \dots, \phi_{m,p}$  are restricted to be the same for all regimes.
- constraints** specifies linear constraints applied to the autoregressive parameters.  
**For non-restricted models:** a list of size  $(p \times q_m)$  constraint matrices  $C_m$  of full column rank satisfying  $\phi_m = C_m \psi_m$  for all  $m = 1, \dots, M$ , where  $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$  and  $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ .

**For restricted models:** a size  $(p \times q)$  constraint matrix  $C$  of full column rank satisfying  $\phi = C\psi$ , where  $\phi = (\phi_1, \dots, \phi_p)$  and  $\psi = (\psi_1, \dots, \psi_q)$ .

Symbol  $\phi$  denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always  $p$  for all regimes. Ignore or set to NULL if applying linear constraints is **not** desired.

parametrization

is the model parametrized with the "intercepts"  $\phi_{m,0}$  or "means"  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ ?

### Details

Differs from the function `uncondMoments` in arguments. This function exists for technical reasons only.

### Value

Returns a list containing the unconditional mean, variance, first  $p$  autocovariances and autocorrelations. Note that the lag-zero autocovariance/correlation is not included in the "first  $p$ " but is given in the `uncond_variance` component separately.

### References

- Kalliovirta L., Meitz M. and Saikkonen P. 2015. Gaussian Mixture Autoregressive model for univariate time series. *Journal of Time Series Analysis*, **36**, 247-266.
- Meitz M., Preve D., Saikkonen P. 2018. A mixture autoregressive model based on Student's  $t$ -distribution. arXiv:1805.04010 [[econ.EM](#)].
- There are currently no published references for the G-StMAR model, but it's a straightforward generalization with theoretical properties similar to the GMAR and StMAR models.
- Lutkepohl H. 2005. New Introduction to Multiple Time Series Analysis. *Springer*.

---

VIX

*CBOE Volatility Index: VIX*

---

### Description

A dataset containing daily CBOE Volatility Index from 5th of July 2016 to 4th of July 2017.

### Usage

VIX

### Format

A numeric vector containing 252 observations.

### Source

<https://fred.stlouisfed.org/series/VIXCLS>

---

warn_dfs	<i>Warn about large degrees of freedom parameter values</i>
----------	---

---

**Description**

warn\_dfs warns if the model contains large degrees of freedom parameter values possibly indicating unreliable numerical derivatives.

**Usage**

```
warn_dfs(object, p, M, params, model = c("GMAR", "StMAR", "G-StMAR"),
         restricted = FALSE, constraints = NULL, warn_about = c("derivs",
         "errors"))
```

**Arguments**

object	an object to be tested
p	a positive integer specifying the order of AR coefficients.
M	<p><b>For GMAR and StMAR models:</b> a positive integer specifying the number of mixture components.</p> <p><b>For G-StMAR model:</b> a size (2x1) vector specifying the number of <i>GMAR-type</i> components M1 in the first element and <i>StMAR-type</i> components M2 in the second. The total number of mixture components is M=M1+M2.</p>
params	<p>a real valued parameter vector specifying the model.</p> <p><b>For non-restricted models: For GMAR model:</b> Size <math>(M(p+3) - 1x1)</math> vector <math>\theta=(v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1})</math>, where <math>v_m = (\phi_{m,0}, \phi_m, \sigma_m^2)</math> and <math>\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})</math>, <math>m = 1, \dots, M</math>.</p> <p><b>For StMAR model:</b> Size <math>(M(p+4) - 1x1)</math> vector <math>(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)</math>.</p> <p><b>For G-StMAR model:</b> Size <math>(M(p+3) + M2 - 1x1)</math> vector <math>(\theta, \nu) = (v_1, \dots, v_M, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)</math>.</p> <p><b>With linear constraints:</b> Replace the vectors <math>\phi_m</math> with vectors <math>\psi_m</math> and provide a list of constraint matrices <b>C</b> that satisfy <math>\phi_m = R_m \psi_m</math> for all <math>m = 1, \dots, M</math>, where <math>\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})</math>.</p> <p><b>For restricted models: For GMAR model:</b> Size <math>(3M + p - 1x1)</math> vector <math>\theta = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1})</math>, where <math>\phi = (\phi_1, \dots, \phi_M)</math>.</p> <p><b>For StMAR model:</b> Size <math>(4M + p - 1x1)</math> vector <math>(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_1, \dots, \nu_M)</math>.</p> <p><b>For G-StMAR model:</b> Size <math>(3M + M2 + p - 1x1)</math> vector <math>(\theta, \nu) = (\phi_{1,0}, \dots, \phi_{M,0}, \phi, \sigma_1^2, \dots, \sigma_M^2, \alpha_1, \dots, \alpha_{M-1}, \nu_{M1+1}, \dots, \nu_M)</math>.</p> <p><b>With linear constraints:</b> Replace the vector <math>\phi</math> with vector <math>\psi</math> and provide a constraint matrix <b>C</b> that satisfies <math>\phi = R\psi</math>, where <math>\psi = (\psi_1, \dots, \psi_q)</math>.</p>

Symbol  $\phi$  denotes an AR coefficient,  $\sigma^2$  a variance,  $\alpha$  a mixing weight and  $\nu$  a degrees of freedom parameter. If `parametrization=="mean"` just replace each intercept term  $\phi_{m,0}$  with regimewise mean  $\mu_m = \phi_{m,0} / (1 - \sum \phi_{i,m})$ . In the **G-StMAR** model the first M1 components are *GMAR-type* and the rest M2 components are *StMAR-type*. Note that in the case **M=1** the parameter  $\alpha$  is dropped, and in the case of **StMAR** or **G-StMAR** model the degrees of freedom parameters  $\nu_m$  have to be larger than 2.

model	is "GMAR", "StMAR" or "G-StMAR" model considered? In G-StMAR model the first M1 components are <i>GMAR-type</i> and the rest M2 components are <i>StMAR-type</i> .
restricted	a logical argument stating whether the AR coefficients $\phi_{m,1}, \dots, \phi_{m,p}$ are restricted to be the same for all regimes.
constraints	specifies linear constraints applied to the autoregressive parameters. <b>For non-restricted models:</b> a list of size $(pxq_m)$ constraint matrices $C_m$ of full column rank satisfying $\phi_m = C_m \psi_m$ for all $m = 1, \dots, M$ , where $\phi_m = (\phi_{m,1}, \dots, \phi_{m,p})$ and $\psi_m = (\psi_{m,1}, \dots, \psi_{m,q_m})$ . <b>For restricted models:</b> a size $(pxq)$ constraint matrix $C$ of full column rank satisfying $\phi = C\psi$ , where $\phi = (\phi_1, \dots, \phi_p)$ and $\psi = (\psi_1, \dots, \psi_q)$ . Symbol $\phi$ denotes an AR coefficient. Note that regardless of any constraints, the nominal order of AR coefficients is always p for all regimes. Ignore or set to NULL if applying linear constraints is <b>not</b> desired.
warn_about	warn about inaccurate derivatives or standard errors?

### Details

Either provide a class 'gsmar' object or specify the model by hand.

### Value

Doesn't return anything but throws a warning if any degrees of freedom parameters have value larger than 1000.

# Index

## \*Topic **datasets**

- IE, [44](#)
- logVIX, [55](#)
- VIX, [106](#)
  
- add\_data, [3](#), [27](#), [43](#), [94](#)
- add\_dfs, [4](#)
- all\_pos\_ints, [5](#)
- alt\_gsmar, [6](#)
  
- calc\_gradient, [7](#), [52](#)
- calc\_hessian (calc\_gradient), [7](#)
- change\_parametrization, [11](#)
- changeRegime, [9](#)
- check\_data, [14](#)
- check\_gsmar, [15](#)
- check\_model, [15](#)
- check\_params\_length, [16](#)
- checkAndCorrectData, [12](#)
- checkConstraintMat, [13](#)
- checkPM, [14](#)
- condMoments, [17](#), [27](#), [36–38](#), [43](#), [71](#), [94](#), [104](#)
  
- diagnosticPlot, [20](#), [27](#), [70](#), [71](#), [74](#)
  
- extractRegime, [22](#)
  
- fitGSMAR, [4](#), [7](#), [21](#), [23](#), [43](#), [49](#), [52](#), [70](#), [71](#), [74](#),  
[94](#), [101](#), [103](#)
- format\_valuef, [28](#)
  
- GAfit, [28](#)
- get\_ar\_roots, [34](#)
- get\_foc (calc\_gradient), [7](#)
- get\_gradient, [4](#), [7](#), [27](#), [43](#), [101](#), [103](#)
- get\_gradient (calc\_gradient), [7](#)
- get\_hessian (calc\_gradient), [7](#)
- get\_IC, [35](#)
- get\_regime\_autocovs, [19](#), [35](#), [37](#), [38](#), [104](#)
- get\_regime\_means, [4](#), [7](#), [19](#), [36](#), [36](#), [38](#), [101](#),  
[103](#), [104](#)
  
- get\_regime\_vars, [19](#), [36](#), [37](#), [37](#), [104](#)
- get\_soc (calc\_gradient), [7](#)
- get\_varying\_h, [38](#)
- getOmega, [32](#), [70](#)
- GSMAR, [4](#), [7](#), [21](#), [27](#), [40](#), [49](#), [52](#), [70](#), [71](#), [74](#), [94](#),  
[101](#), [103](#)
  
- IE, [44](#)
- isIdentifiable (isStationary\_int), [46](#)
- isStationary, [44](#)
- isStationary\_int, [46](#)
- iterate\_more, [4](#), [7](#), [27](#), [43](#), [48](#), [101](#), [103](#)
  
- logLik.gsmar (GSMAR), [40](#)
- loglikelihood, [50](#)
- loglikelihood\_int, [52](#)
- logVIX, [55](#)
  
- mixingWeights, [52](#), [56](#), [94](#)
- mixingWeights\_int, [58](#)
  
- nParams, [60](#)
  
- optim, [49](#)
  
- parameterChecks, [61](#)
- pick\_alphas, [62](#)
- pick\_dfs, [63](#)
- pick\_pars, [64](#)
- pick\_phi0, [66](#)
- plot.gsmar (GSMAR), [40](#)
- plot.gsmarpred, [67](#)
- plot.qrtest, [68](#)
- predict.gsmar, [27](#), [43](#), [70](#), [70](#), [94](#)
- print.gsmar (GSMAR), [40](#)
- print.gsmarpred, [72](#)
- print.gsmarsum, [73](#)
- print.qrtest (plot.qrtest), [68](#)
  
- quantileResidualPlot, [21](#), [74](#)
- quantileResiduals, [52](#), [75](#)

quantileResiduals\_int, 78  
quantileResidualTests, 21, 27, 33, 71, 74  
quantileResidualTests (plot.qrtest), 68

random\_arcoefs, 85  
random\_regime, 86  
randomIndividual, 80  
randomIndividual\_int, 83  
reformConstrainedPars, 87  
reformParameters, 89  
reformRestrictedPars, 90  
regime\_distance, 91  
removeAllConstraints, 92  
residuals.gsmar (GSMAR), 40

simulateGSMAR, 21, 27, 43, 71, 74, 93  
smartIndividual (randomIndividual), 80  
smartIndividual\_int  
    (randomIndividual\_int), 83  
sortComponents, 95  
standardErrors, 97  
stmar\_to\_gstmar, 4, 7, 27, 43, 49, 100, 101,  
    103  
stmarpars\_to\_gstmar, 98  
summary.gsmar (GSMAR), 40  
swap\_parametrization, 4, 7, 27, 43, 101,  
    102, 103

uGMAR, 103  
uGMAR-package (uGMAR), 103  
uncondMoments, 19, 27, 36–38, 43, 103  
uncondMoments\_int, 104

VIX, 106

warn\_dfs, 107