

# Package ‘mfGARCH’

June 17, 2021

**Title** Mixed-Frequency GARCH Models

**Version** 0.2.1

**Description** Estimating GARCH-MIDAS (MIXed-DAta-Sampling) models (Engle, Ghysels, Sohn, 2013, <[doi:10.1162/REST\\_a\\_00300](https://doi.org/10.1162/REST_a_00300)>) and related statistical inference, accompanying the paper “Two are better than one: Volatility forecasting using multiplicative component GARCH models” by Conrad and Kleen (2020, <[doi:10.1002/jae.2742](https://doi.org/10.1002/jae.2742)>). The GARCH-MIDAS model decomposes the conditional variance of (daily) stock returns into a short- and long-term component, where the latter may depend on an exogenous covariate sampled at a lower frequency.

**Depends** R (>= 3.3.0)

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Imports** Rcpp, graphics, stats, numDeriv, zoo, maxLik

**LinkingTo** Rcpp

**URL** <https://github.com/onnoKleen/mfGARCH/>

**BugReports** <https://github.com/onnoKleen/mfGARCH/issues>

**Suggests** testthat, dplyr, ggplot2, covr, rmarkdown

**NeedsCompilation** yes

**Author** Onno Kleen [aut, cre] (<<https://orcid.org/0000-0003-4731-4640>>)

**Maintainer** Onno Kleen <r@onnoKleen.de>

**Repository** CRAN

**Date/Publication** 2021-06-17 14:20:02 UTC

## R topics documented:

df_financial	2
df_mfgarch	3

fit_mfgarch . . . . .	4
plot_weighting_scheme . . . . .	6
simulate_mfgarch . . . . .	6
simulate_mfgarch_diffusion . . . . .	7
simulate_mfgarch_rv_dependent . . . . .	8
<b>Index</b>	<b>10</b>

---

df_financial	<i>Stock returns and financial conditions.</i>
--------------	--

---

### Description

A dataset containing the S&P 500 stock returns and the NFCI

### Usage

```
df_financial
```

### Format

A data frame with 11,306 rows and 5 variables:

**date** date

**return** daily S&P 500 log returns times 100

**rv** 5-minute realized variances

**week** a dummy for each year/week combination

**nfc** National Financial Conditions Index

### Source

<https://github.com/onnoKleen/mfGARCH/>

<https://finance.yahoo.com/>

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

<https://realized.oxford-man.ox.ac.uk>

---

df_mfgarch	<i>Mixed-frequency data set.</i>
------------	----------------------------------

---

**Description**

A dataset containing the S&P 500 stock returns, realized variances and macroeconomic variables

**Usage**

df\_mfgarch

**Format**

A data frame with 11,938 rows and 11 variables:

**date** date

**return** daily S&P 500 log returns times 100

**open\_close** open-close returns

**rv** 5-minute realized variances

**vix** Cboe VIX

**year\_week** a dummy for each year/week combination

**dhousing** changes in housing starts

**dindpro** changes in industrial production

**nai** NAI

**nfcindex** National Financial Conditions Index

**year\_month** a dummy for each year/month combination

**Source**

<https://github.com/onnoKleen/mfGARCH/>

<https://finance.yahoo.com/>

<https://fred.stlouisfed.org>

<https://realized.oxford-man.ox.ac.uk>

---

fit_mfgarch	<i>This function estimates a multiplicative mixed-frequency GARCH model. For the sake of numerical stability, it is best to multiply log returns by 100.</i>
-------------	--

---

### Description

This function estimates a multiplicative mixed-frequency GARCH model. For the sake of numerical stability, it is best to multiply log returns by 100.

### Usage

```
fit_mfgarch(
  data,
  y,
  x = NULL,
  K = NULL,
  low.freq = "date",
  var.ratio.freq = NULL,
  gamma = TRUE,
  weighting = "beta.restricted",
  x.two = NULL,
  K.two = NULL,
  low.freq.two = NULL,
  weighting.two = NULL,
  multi.start = FALSE,
  control = list(par.start = NULL)
)
```

### Arguments

data	data frame containing a column named date of type 'Date'.
y	name of high frequency dependent variable in df.
x	covariate employed in mfGARCH.
K	an integer specifying lag length K in the long-term component.
low.freq	a string of the low frequency variable in the df.
var.ratio.freq	specify a frequency column on which the variance ratio should be calculated.
gamma	if TRUE, an asymmetric GJR-GARCH is used as the short-term component. If FALSE, a simple GARCH(1,1) is employed.
weighting	specifies the weighting scheme employed in the long-term component. Options are "beta.restricted" (default) or "beta.unrestricted"
x.two	optional second covariate
K.two	lag length of optional second covariate
low.freq.two	low frequency of optional second covariate

weighting.two	specifies the weighting scheme employed in the optional second long-term component. Currently, the only option is "beta.restricted"
multi.start	if TRUE, optimization is carried out with multiple starting values
control	a list

## Value

A list of class mfGARCH with letters and numbers.

- par - vector of estimated parameters
- rob.std.err - sandwich/HAC-type standard errors
- broom.mgarch - a broom-like data.frame with entries 1) estimate: column of estimated parameters 2) rob.std.err - sandwich/HAC-type standard errors 3) p.value - p-values derived from sandwich/HAC-type standard errors 4) opg.std.err - Bollerslev-Wooldrige/OPG standard errors for GARCH processes 5) opg.p.value - corresponding alternative p-values
- tau - fitted long-term component
- g - fitted short-term component
- df.fitted - data frame with fitted values and residuals
- K - chosen lag-length in the long-term component
- weighting.scheme - chosen weighting scheme
- llh - log-likelihood value at estimated parameter vector
- bic - corresponding BIC value
- y - dependent variable y
- optim - output of the optimization routine
- K.two - lag-length of x.two if two covariates are employed
- weighting.scheme.two - chosen weighting scheme of x.two (if K.two != NULL)
- tau.forecast - one-step ahead forecast of the long-term component
- variance.ratio - calculated variance ratio
- est.weighting - estimated weighting scheme
- est.weighting.two - estimated weighting scheme of x.two (if K.two != NULL)

## Examples

```
## Not run:
fit_mfgarch(data = df_financial, y = "return", x = "nfci", low.freq = "week", K = 52)
fit_mfgarch(data = df_mfgarch, y = "return", x = "nfci", low.freq = "year_week", K = 52,
x.two = "dindpro", K.two = 12, low.freq.two = "year_month", weighting.two = "beta.restricted")

## End(Not run)
```

---

plot\_weighting\_scheme *This function plots the weighting scheme of an estimated GARCH-MIDAS model*

---

### Description

This function plots the weighting scheme of an estimated GARCH-MIDAS model

### Usage

```
plot_weighting_scheme(x)
```

### Arguments

x                    mfGARCH object obtained by fit\_mfgarch

---

simulate\_mfgarch     *This function simulates a GARCH-MIDAS model. Innovations can follow a standard normal or student-t distribution.*

---

### Description

This function simulates a GARCH-MIDAS model. Innovations can follow a standard normal or student-t distribution.

### Usage

```
simulate_mfgarch(
  n.days,
  mu,
  alpha,
  beta,
  gamma,
  m,
  theta,
  w1 = 1,
  w2,
  K,
  psi,
  sigma.psi,
  low.freq = 1,
  n.intraday = 288,
  student.t = NULL,
  corr = 0
)
```

**Arguments**

n.days	number of days
mu	mu
alpha	alpha
beta	beta
gamma	gamma
m	m
theta	theta
w1	w1
w2	w2
K	K
psi	psi
sigma.psi	sigma.psi
low.freq	number of days per low-frequency period
n.intraday	number of maximum intraday returns
student.t	either NULL or degrees of freedom
corr	correlation between innovations (should only be used for daily tau)

**Examples**

```
simulate_mfgarch(n.days = 200, mu = 0, alpha = 0.06, beta = 0.92, gamma = 0, m = 0,
theta = 0.1, w1 = 1, w2 = 3, K = 12, psi = 0.98, sigma.psi = 0.1, low.freq = 10)
```

---

```
simulate_mfgarch_diffusion
```

*This function simulates a GARCH-MIDAS model where the short-term GARCH component is replaced by its diffusion limit, see Andersen (1998)*

---

**Description**

This function simulates a GARCH-MIDAS model where the short-term GARCH component is replaced by its diffusion limit, see Andersen (1998)

**Usage**

```
simulate_mfgarch_diffusion(
  n.days,
  mu,
  alpha,
  beta,
  m,
```

```

theta,
w1 = 1,
w2,
K,
psi,
sigma.psi,
low.freq = 1,
n.intraday = 288
)

```

### Arguments

n.days	number of days
mu	mu
alpha	alpha
beta	beta
m	m
theta	theta
w1	w1
w2	w2
K	K
psi	psi
sigma.psi	sigma.psi
low.freq	low.freq
n.intraday	n.intraday

### Examples

```

## Not run: simulate_mfgarch_diffusion(n.days = 200, mu = 0, alpha = 0.06, beta = 0.92, m = 0,
theta = 0.1, w1 = 1, w2 = 3, K = 12, psi = 0.98, sigma.psi = 0.1, low.freq = 10)
## End(Not run)

```

---

```
simulate_mfgarch_rv_dependent
```

*Simulate a GARCH-MIDAS similar to Wang/Ghysels with lagged RVol as covariate*

---

### Description

Simulate a GARCH-MIDAS similar to Wang/Ghysels with lagged RVol as covariate



**Usage**

```
simulate_mfgarch_rv_dependent(  
  n.days,  
  mu,  
  alpha,  
  beta,  
  gamma,  
  m,  
  theta,  
  w1 = 1,  
  w2,  
  K,  
  n.intraday = 288,  
  low.freq = 1,  
  rvol = FALSE  
)
```

**Arguments**

n.days	number of days
mu	mu
alpha	alpha
beta	beta
gamma	gamma
m	m
theta	theta
w1	w1
w2	w2
K	K
n.intraday	number of maximum intraday returns, default 288
low.freq	number of days per low frequency
rvol	if TRUE, the square root of the realized variance is used as a covariate

**Examples**

```
simulate_mfgarch_rv_dependent(n.days = 2200, mu = 0, alpha = 0.06, beta = 0.92, gamma = 0, m = 0,  
  theta = 0.1, w1 = 1, w2 = 3, K = 3, low.freq = 22)
```

# Index

- \* **datasets**

- df\_financial, 2

- df\_mfgarch, 3

- \* **fit\_mfgarch**

- fit\_mfgarch, 4

- \* **simulate\_mfgarch**

- simulate\_mfgarch, 6

- simulate\_mfgarch\_diffusion, 7

df\_financial, 2

df\_mfgarch, 3

fit\_mfgarch, 4

plot\_weighting\_scheme, 6

simulate\_mfgarch, 6

simulate\_mfgarch\_diffusion, 7

simulate\_mfgarch\_rv\_dependent, 8