Package: garchf (via r-universe)

October 27, 2026

Type Package

Title Forecasting with GARCH

Version 0.1.1

Description Forecasting with GARCH, using 'rugarch' with an interface like 'forecast' package's.

Imports forecast, rugarch

Suggests knitr, rmarkdown

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Remotes Techtonique/ahead, Techtonique/crossvalidation

URL https://techtonique.r-universe.dev/garchf

VignetteBuilder knitr

Depends R (>= 3.5)

Config/pak/sysreqs cmake libgmp3-dev

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

Date/Publication 2026-08-04 12:52:45 UTC

RemoteUrl https://github.com/thierrymoudiki/garchf

RemoteRef HEAD

RemoteSha bb9cfd6d710bcdf247ea3200be5a4f05ea0a461b

Contents

condvolf 2
twostagef 4
xgarchf 5

Index **7**

condvolf *Model-agnostic statistical probabilistic forecasting with conditional volatility*

Description

Combines a mean model and a volatility model to produce probabilistic forecasts. Uncertainty is propagated from two sources: simulated innovation draws and simulated volatility paths derived from the variance model's residuals via bootstrap.

Usage

```
condvolf(
  y,
  h = 10L,
  mean_model = forecast::thetaf,
  sigma_model = forecast::thetaf,
  innovation = c("empirical", "gaussian", "student"),
  level = 95,
  B = 500L,
  bootstrap_vol = TRUE,
  ...
)
```

Arguments

y	A numeric vector or time series of class ts.
h	Forecasting horizon.
mean_model	Function to fit the mean model (default: forecast::auto.arima). Options include forecast::auto.arima, forecast::ets, forecast::thetaf, or any custom function that returns an object with forecast method.
sigma_model	Function to fit the variance model on squared residuals (default: forecast::auto.arima).
innovation	Distribution for standardized innovations. One of "gaussian" (fastest, often sufficient), "student" (heavy-tailed), or "empirical" (non-parametric, most flexible). Only used for conformal=FALSE
level	Confidence level for prediction intervals (default: 95).
B	Number of simulated paths (default: 2000).
bootstrap_vol	Whether to bootstrap volatility residuals (default: TRUE). If FALSE, uses parametric normal approximation for volatility.
...	Additional arguments passed to mean_model and sigma_model.
conformal	Logical. If TRUE, enables split conformal prediction to calibrate interval width. The series is split into a training portion and a calibration portion (controlled by cal_frac); a base condvolf fit on the training portion produces calibration residuals, which are normalized by a volatility model, resampled via moving

block bootstrap, re-inflated by the forecast volatility, and added to the full-data point forecast. This typically yields tighter and better- calibrated intervals than the raw simulation approach. Default is FALSE.

Value

A forecast object with components:

x Original time series.

mean Point forecast (mean of simulations).

lower Lower prediction interval bound.

upper Upper prediction interval bound.

level Confidence level.

sims Matrix of simulated forecast paths (h x B).

model List containing mean and sigma model fits.

residuals Residuals from the mean model.

standardized_residuals Scaled standardized residuals used for innovation fitting.

method Character string describing the method used.

Examples

```
library(forecast)

# Basic usage with Google stock data
y <- fpp2::goog200

# Gaussian innovations (fastest, often sufficient)
fc1 <- condvolf(y, h = 20, innovation = "gaussian")
plot(fc1)

# Student-t innovations (heavy-tailed)
fc2 <- condvolf(y, h = 20, innovation = "student")
plot(fc2)

# Empirical innovations (non-parametric)
fc3 <- condvolf(y, h = 20, innovation = "empirical")
plot(fc3)

# Using different mean and volatility models
fc4 <- condvolf(y, h = 20,
                mean_model = forecast::thetaf,
                sigma_model = forecast::ets,
                innovation = "gaussian")

# Compare prediction intervals
par(mfrow = c(2, 2))
plot(fc1, main = "Gaussian innovations")
plot(fc2, main = "Student-t innovations")
plot(fc3, main = "Empirical innovations")
```

```
plot(fc4, main = "Different mean and volatility models")
par(mfrow = c(1, 1))
```

`twostagef` *Two-stage forecasting model*

Description

Two-stage forecasting model

Usage

```
twostagef(
  y,
  h = 10,
  mean_model = forecast::thetaf,
  sigma_model = forecast::thetaf,
  residuals_model = forecast::thetaf,
  innovation = c("bootstrap", "fitdistr"),
  cal_frac = 0.5,
  level = 95,
  B = 500L,
  ...
)
```

Arguments

`y`	A numeric vector or time series of class `ts`.
`h`	Forecasting horizon.
`mean_model`	Function to fit the mean model (default: `forecast::auto.arima`). Options include `forecast::auto.arima`, `forecast::ets`, `forecast::thetaf`, or any custom function that returns an object with `forecast` method.
`sigma_model`	Function to fit the variance model on squared residuals (default: `forecast::auto.arima`).
`residuals_model`	A function from package forecast, the model adjusted to residuals for 2-stage forecasting
`innovation`	Either "bootstrap" or "fitdistr"; the latter is parametric
`cal_frac`	Numeric in `(0, 1)`. Fraction of observations reserved for the calibration set. The calibration set is always at least `h` observations long regardless of `cal_frac`. Larger values improve calibration stability at the cost of a shorter training set for the base model. Default is `0.5`.
`level`	Confidence level for prediction intervals (default: 95).
`B`	Number of simulated paths (default: 2000).
`...`	Additional arguments passed to `mean_model` and `sigma_model`.

Value

A `forecast` object with components:

`x` Original time series.

`mean` Point forecast (mean of simulations).

`lower` Lower prediction interval bound.

`upper` Upper prediction interval bound.

`level` Confidence level.

`sims` Matrix of simulated forecast paths (`h` x `B`).

`model` List containing `mean` and `sigma` model fits.

`residuals` Residuals from the mean model.

`standardized_residuals` Scaled standardized residuals used for innovation fitting.

`method` Character string describing the method used.

`xgarchf` *GARCH forecasting wrapper returning a forecast object*

Description

Similar spirit to forecast::thetaf(), but using rugarch models.

Usage

```
xgarchf(
  y,
  h = 10L,
  model = c("eGARCH", "sGARCH", "gjrGARCH", "apARCH", "iGARCH"),
  arma_order = c(0, 0),
  garch_order = c(1, 1),
  distribution_model = c("norm", "std", "ged"),
  level = 95,
  B = 500L,
  ...
)
```

Arguments

`y`	numeric vector or ts
`h`	forecast horizon
`model`	mean model ("sGARCH", "eGARCH", "gjrGARCH", ...)
`arma_order`	ARMA order for the conditional mean; a vector with 2 elements
`garch_order`	GARCH order; a vector with 2 elements

distribution_model	conditional distribution "norm", "std", "ged"
level	confidence level(s)
B	number of simulations to return if sim=TRUE
...	Additional parameters passed to rugarch::ugarchfit

Value

object of class forecast

Examples

```
## Not run:
y <- diff(log(fpp2::goog200))
fit <- xgarchf(y, h = 20, model = "eGARCH")
plot(fit)

## End(Not run)
```

Index

condvolf, 2

twostagef, 4

xgarchf, 5