Package 'stochvolTMB'

August 13, 2021

Type Package

Title Likelihood Estimation of Stochastic Volatility Models

Version 0.2.0

Date 2021-08-09

Author Jens Christian Wahl <jens.c.wahl@gmail.com>

Maintainer Jens Christian Wahl <jens.c.wahl@gmail.com>

Description

Parameter estimation for stochastic volatility models using maximum likelihood. The latent log-volatility is
integrated out of the likelihood using the Laplace approximation. The models are fitted via 'TMB' (Template Model
Builder) (Kristensen, Nielsen, Berg, Skaug, and Bell (2016) <doi:10.18637/jss.v070.i05>).

License GPL-3

Depends R (>= 3.5.0)

Imports TMB, ggplot2, sn, stats, data.table, MASS

LinkingTo RcppEigen, TMB

Suggests testthat (>= 2.1.0), shiny, knitr, rmarkdown, stochvol

URL https://github.com/JensWahl/stochvolTMB

BugReports https://github.com/JensWahl/stochvolTMB/issues

RoxygenNote 7.1.1

Encoding UTF-8

LazyData true

VignetteBuilder knitr

NeedsCompilation yes

Repository CRAN

Date/Publication 2021-08-13 11:30:02 UTC

R topics documented:

`demo` *Run shiny demo*

Description

Run shiny demo

Usage

```
demo()
```

`estimate_parameters` *Estimate parameters for the stochastic volatility model*

Description

Estimate parameters of a stochastic volatility model with a latent log-volatility following an autoregressive process of order one with normally distributed noise. The following distributions are implemented for the observed process:

- Gaussian distribution
- t-distribution
- Leverage: Gaussian distribution with leverage where the noise of the latent process is correlated with the observational distribution
- Skew gaussian distribution

The parameters is estimated by minimizing the negative log-likelihood (nll) and the latent log-volatility is integrated out by applying the Laplace approximation.

Usage

```
estimate_parameters(data, model = "gaussian", opt.control = NULL, ...)
```

Arguments

data	A vector of observations.
model	A character specifying the model. Must be one of the following: "gaussian", "t", "leverage", "skew_gaussian".
opt.control	An optional list of parameters for nlminb.
...	additional arguments passed to MakeADFun.

Value

Object of class stochvolTMB

Examples

```
# load data
data("spy")

# estimate parameters
opt <- estimate_parameters(spy$log_return, model = "gaussian")

# get parameter estimates with standard error
estimates <- summary(opt)

# plot estimated volatility with 95 % confidence interval
plot(opt, include_ci = TRUE)
```

---

logit *Logit transformation from the real line to (-1, 1).*

Description

Logit transformation from the real line to (-1, 1).

Usage

```
logit(x)
```

Arguments

x	double

Value

double

plot.stochvolTMB *Plot the estimated latent volatility process*

Description

Displays the estimated latent volatility process over time.

Usage

```
## S3 method for class 'stochvolTMB'
plot(x, ..., include_ci = TRUE, plot_log = TRUE, dates = NULL, forecast = NULL)
```

Arguments

x	A stochvolTMB object returned from estimate_parameters.
...	Currently not used.
include_ci	Logical value indicating if volatility should be plotted with approximately 95% confidence interval.
plot_log	Logical value indicating if the estimated should be plotted on log or original scale. If plot_log = TRUE the process h is plotted. If plot_log = FALSE 100 sigma_y exp(h / 2) is plotted.
dates	Vector of length ncol(x$nobs), providing optional dates for labeling the x-axis. The default value is NULL; in this case, the axis will be labeled with numbers.
forecast	Integer specifying number of steps to forecast.

Value

ggplot object with plot of estimated estimated volatility.

predict.stochvolTMB *Predict future returns and future volatilities*

Description

Takes a stochvolTMB object and produces draws from the predictive distribution of the latent volatility and future log-returns.

Usage

```
## S3 method for class 'stochvolTMB'
predict(object, steps = 1L, nsim = 10000, include_parameters = TRUE, ...)
```

Arguments

object	A stochvolTMB object returned from estimate_parameters.
steps	Integer specifying number of steps to predict.
nsim	Number of draws from the predictive distribution.
include_parameters	Logical value indicating if fixed parameters should be simulated from their asymptotic distribution, i.e. multivariate normal with inverse hessian as covariance matrix.
...	Not is use.

Value

List of simulated values from the predictive distribution of the latent volatilities and log-returns.

---

residuals *Calculate one-step-ahead (OSA) residuals for stochastic volatility model.*

Description

This function is very time consuming and by default computes the one-step-ahead residual for the last 100 observations. See the function oneStepPredict and the paper in the references for more details.

Usage

```
residuals(object, conditional = 1:(object$nobs - 100), ...)
```

Arguments

object	A stochvolTMB object.
conditional	Index vector of observations that are fixed during OSA. By default the residuals of the last 100 observations are calculated. If set to NULL it will calculate one-step-ahead residuals for all observations.
...	Currently not used.

Value

Vector of one-step-ahead residuals. If the model is correctly specified, these should be standard normal.

References

https://www.researchgate.net/publication/316581864_Validation_of_ecological_state_space_models_using_the_Laplace_approximation

simulate_parameters *Simulate from the asymptotic distribution of the parameter estimates*

Description

Sampling is done on the scale the parameters were estimated. The standard deviations are simulated on log-scale and the persistence is simulated on logit scale. The same is true for the correlation parameter in the leverage model.

Usage

```
simulate_parameters(object, nsim = 1000)
```

Arguments

object	A stochvolTMB object.
nsim	Number of simulations.

Value

matrix of simulated values.

sim_sv *Simulate log-returns from a stochastic volatility model*

Description

This function draws the initial log-volatility (h_t) from its stationary distribution, meaning that h_0 is drawn from a gaussian distribution with mean zero and standard deviation sigma_h / sqrt(1 -phi^2). h_{t+1} is then simulated from its conditional distribution given h_t, which is N(phi*h_t, sigma_h). Log-returns (y_t) is simulated from its conditional distribution given the latent process h. If model = "gaussian", then y_t given h_t is gaussian with mean zero and standard deviation equal to sigma_y*exp(h_t / 2). Heavy tail returns can be obtained by simulating from the t-distribution by setting model = "t". How heavy of a tail is specified by the degree of freedom parameter df. Note that the observations are scaled by sqrt((df-2)/2) so that the error term has variance equal to one. Asymmetric returns are obtained from the "skew_gaussian" model. How asymmetric is governed by the skewness parameter alpha. The so called leverage model, where we allow for correlation between log-returns and volatility can be simulated by setting model to "leverage" and specifying the correlation parameter rho.

Usage

```
sim_sv(
 param = list(phi = 0.9, sigma_y = 0.4, sigma_h = 0.2, df = 4, alpha = -2, rho = -0.7),
  nobs = 1000L,
  seed = NULL,
  model = "gaussian"
)
```

Arguments

param	List of parameters. This includes the standard deviation of the observations, sigma_y, the standard deviation of the latent volatility process, sigma_h, the persistence parameter phi. If model = "t", the degree of freedom df must be specified. If model = "skew_gaussian", the skewness parameter alpha must be specified and if model = "leverage", the correlation rho between the latent error term and the observational error has to be specified.
nobs	Length of time series.
seed	Seed to reproduce simulation.
model	Distribution of error term.

Value

data.table with columns y (observations) and h (latent log-volatility).

---

spy *Daily closing prices for the S&P500 from 2005 to 2018.*

Description

A dataset containing the prices and log-returns of the S&P500 from 2005 to 2018

Usage

```
spy
```

Format

A data frame with 3522 rows and 3 variables:

date date

price price, in US dollars

log_return logarithmic return ...

`summary.stochvolTMB` *Summary tables of model parameters*

Description

Extract parameters, transformed parameters and latent log volatility along with standard error, z-value and p-value

Usage

```
## S3 method for class 'stochvolTMB'
summary(object, ..., report = c("all", "fixed", "transformed", "random"))
```

Arguments

`object`	A `stochvolTMB` object.
`...`	Currently not used.
`report`	Parameters to report with uncertainty estimates. Can be any subset of "fixed", "transformed" or "random" (see summary.sdreport). "fixed" report the parameters on the scale they were estimated, for example are all standard deviations estimated on log scale. "transformed" report all transformed parameters, for example estimated standard deviations transformed from log scale by taking the exponential. Lastly, "random" report the estimated latent log-volatility.

Value

`data.table` with parameter estimates, standard error, z-value and approximated p-value.

`summary.stochvolTMB_predict`
Calculate quantiles based on predictions from the predictive distribution

Description

Calculate quantiles based on predictions from the predictive distribution

Usage

```
## S3 method for class 'stochvolTMB_predict'
summary(object, ..., quantiles = c(0.025, 0.975), predict_mean = TRUE)
```

Arguments

object	A stochvolTMB_summary object.
...	Not used.
quantiles	A numeric vector specifying which quantiles to calculate.
predict_mean	bool. Should the mean be predicted?

Value

A list of data.tables. One for y, h and h_exp.

Index