Package 'strucvol'

May 23, 2023

Type Package

Encoding UTF-8

Title Structural stochastic volatility models.

Version 1.0

Date 2023-05-22

Author Alexander Back

Description Estimation and inference for structural stochastic volatility models.

License GPL (>= 2)

Imports Rcpp (>= 1.0.10),
FKF,
Rsolnp

LinkingTo Rcpp

Depends R (>= 2.10)

LazyData true

RoxygenNote 7.2.3

R topics documented:

`strucvol-package` *A short title line describing what the package does*

Description

A more detailed description of what the package does. A length of about one to five lines is recommended.

Details

This section should provide a more detailed overview of how to use the package, including the most important functions.

Author(s)

Alexander Back, email optional.

Maintainer: Alexander Back <your@email.com>

References

This optional section can contain literature or other references for background information.

See Also

Optional links to other man pages

Examples

```
  ## Not run:
     ## Optional simple examples of the most important functions
     ## These can be in \dontrun{} and \donttest{} blocks.

## End(Not run)
```

`fitmssv` *Fit a bivariate structural stochastic volatility model.*

Description

Fit a bivariate structural stochastic volatility model.

Usage

```
fitmssv(y, x, start = c(0.95, 0.95, 0.3, 0.3, 0.02))
```

Arguments

`y`	a bivariate numeric or time series containing log returns. The first column should contain the "structural" series, while the second corresponds to the market.
`x`	an explanatory variable, presumably the log of a leverage multiplier.
`start`	starting parameters for the optimization.

Value

A list containing the output from the solver ("model") and the outputs from the Kalman filter ("fit").

`fitssv` *Fit a structural stochastic volatility model.*

Description

Fit a structural stochastic volatility model.

Usage

```
fitssv(y, x, N = 5, start = c(0.95, 0.3))
```

Arguments

`y`	a numeric vector or time series containing log returns.
`x`	an explanatory variable, presumably the log of a leverage multiplier.
`N`	number of importance samples to draw for the Monte Carlo ll evaluation.
`start`	starting parameters for the optimization.

Value

A list containing the output from the solver ("model") and the outputs from the Kalman filter and monte carlo evaluation routine ("fit").

`fitsv` *Fit a standard stochastic volatility model.*

Description

Fit a standard stochastic volatility model.

Usage

```
fitsv(y, N = 5, start = c(0.95, 0.3))
```

Arguments

`y`	a numeric vector or time series containing log returns.
`N`	number of importance samples to draw for the Monte Carlo ll evaluation.
`start`	starting parameters for the optimization.

Value

A list containing the output from the solver (model) and the outputs from the Kalman filter and monte carlo evaluation routine (fit).

`levmulttest` *Test for misspecification in the form of an excluded leverage multiplier.*

Description

Test for misspecification in the form of an excluded leverage multiplier.

Usage

```
levmulttest(data, lmt, model)
```

Arguments

`data`	The data to be tested.
`model`	The null model.

Value

A test statistic with an asymptotic chi^2(1) distribution under the null model.

`llevtest` *Test for a leverage effect in the data.*

Description

Test for a leverage effect in the data.

Usage

```
llevtest(data, model)
```

Arguments

`data`	The data to be tested.
`model`	The null model.

Value

A test statistic with an asymptotic chi^2(1) distribution under the null model.

`llratiotest` *Likelihood ratio test for two competing stochastic volatility models.*

Description

Likelihood ratio test for two competing stochastic volatility models.

Usage

```
llratiotest(model0, model1)
```

Arguments

`model0`	The null model
`model1`	The alternative model

Value

The likelihood ratio test statistic, here presumably asymptotically distributed as chi^2(1) under the null.

`rcpp_hello_world` *Simple function using Rcpp*

Description

Simple function using Rcpp

Usage

```
rcpp_hello_world()
```

Examples

```
## Not run:
rcpp_hello_world()

## End(Not run)
```

Index