Package 'HARModel'

October 12, 2018

Type Package

Title Heterogeneous Autoregressive Models

Version 0.1

Date 2018-09-26

Author Emil Sjoerup

Maintainer Emil Sjoerup <emilsjoerup@live.dk>

Description Estimation, simulation, and forecasting using the HAR model from Corsi(2009) <DOI:10.1093/jjfinec/nbp001>.

License GPL-3

Imports Rcpp (>= 0.12.17) , xts, zoo, sandwich,

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Depends R (>= 2.10), methods

Repository CRAN

Date/Publication 2018-10-12 14:00:02 UTC

R topics documented:

HARModel-package 2
DJIRealizedMeasures 3
HARDataCreationC 3
HARestimate 4
HARforecast 6
HARForecast-class 7
HARModel-class 8
HARMonteCarlo 9
HARSim-class 10
HARsimulate 11
SandwichNeweyWest 12
uncondmean 13

Index **14**

HARModel-package *Heterogeneous Autoregressive Models*

Description

Estimation, simulation, and forecasting using the HAR model from Corsi(2009) <DOI:10.1093/jjfinec/nbp001>.

Details

The DESCRIPTION file:

Package:	HARModel
Type:	Package
Title:	Heterogeneous Autoregressive Models
Version:	0.1
Date:	2018-09-26
Author:	Emil Sjoerup
Maintainer:	Emil Sjoerup <emilsjoerup@live.dk>
Description:	Estimation, simulation, and forecasting using the HAR model from Corsi(2009) <DOI:10.1093/jjfinec/n
License:	GPL-3
Imports:	Rcpp (>= 0.12.17) , xts, zoo, sandwich,
LinkingTo:	Rcpp, RcppArmadillo
NeedsCompilation:	Yes
Depends:	R (>= 2.10), methods

Index of help topics:

DJIRealizedMeasures	Dow Jones Realized Measures
HARDataCreationC	Data creation for HAR estimation and forecasting.
HARForecast-class	HARForecast
HARModel-class	HARModel
HARModel-package	Heterogeneous Autoregressive Models
HARMonteCarlo	Monte Carlo simulation of HAR model
HARSim-class	HARSim
HARestimate	HAR estimation
HARforecast	HAR forecasting
HARsimulate	HAR simulation
SandwichNeweyWest	HAC Variance Covariance Matrix For 'HARModel' Objects
uncondmean	Unconditional mean

Author(s)

Emil Sjoerup

Maintainer: Emil Sjoerup <emilsjoerup@live.dk>

References

Corsi, F. 2009, A Simple Approximate Long-Memory Model of Realized Volatility, *Journal of Financial Econometrics*, 174–196 .

`DJIRealizedMeasures` *Dow Jones Realized Measures*

Description

Realized measures

Format

A large `xts` object containing trading data for the Dow Jones Industial index from 2001 to september 2018

Details

See the website of the data set for details.

Source

https://realized.oxford-man.ox.ac.uk/data

References

Heber, Gerd, Asger Lunde, Neil Shephard and Kevin Sheppard (2009) "Oxford-Man Institute's realized library", Oxford-Man Institute, University of Oxford. Library version: 0.3 https://realized.oxford-man.ox.ac.uk/data

---

`HARDataCreationC` *Data creation for HAR estimation and forecasting.*

Description

Constructs a matrix containing the realized measure and the lagged moving averages according to the supplied lag-vector.

Usage

```
HARDataCreationC(vRealizedmeasure, vLags)
```

Arguments

vRealizedmeasure	A vector containing a realized measure of the integrated volatility.
vLags	A vector denoting which lags should be used in the creation `c(1,5,22)` is in line with Corsi(2009).

Details

None for now.

Value

A matrix containing the data used for HAR estimation or forecasting.

Author(s)

Emil Sjoerup

References

Corsi, F. 2009, A Simple Approximate Long-Memory Model of Realized Volatility, *Journal of Financial Econometrics*, 174–196.

See Also

See Also `HARestimate`

Examples

```
## Not run:
## Construct a  vector of size 1000:
vData = rnorm(1000)^2
## Construct a lag-vector
vLags = c(1,5,22)
Example = HARDataCreationC(vData , vLags)
## The dimensions of Example is (978 , 4)

## End(Not run)
```

HARestimate *HAR estimation*

Description

This function does HAR estimation in line with Corsi(2009). It is important to note that the maximum of the lag vector (22 in the standard case) observations are omitted because these are used to generate the data.

Usage

```
HARestimate(vRealizedMeasure, vLags = c(1,5,22),
show = TRUE)
```

Arguments

vRealizedMeasure	An xts object containing a realized measure of the integrated volatility.
vLags	A vector denoting which lags should be used in the estimation, standard of c(1,5,22) is in line with Corsi(2009).
show	Logical to determine whether the output should be shown when done. Setting to FALSE makes Monte Carlo simulation and forecasting much faster.

Value

Returns a HARModel object which is an S4 class object containing:

1	Model, an lm object containing the estimation details.
2	Info, a list containing information used, the vector of lags , vLags, Dates, which is a vector of the dates of the data (if the provided realized measure is an "xts" object), and ElapsedTime , the time elapsed in seconds.
3	The List Data contains the realized measure provided.

Author(s)

Emil Sjoerup

References

Corsi, F. 2009, A Simple Approximate Long-Memory Model of Realized Volatility, *Journal of Financial Econometrics*, 174–196.

See Also

See Also HARDataCreationC

Examples

```
set.seed(123)
#Simulate a HAR process:
HARSim = HARsimulate(iLength = 10000,
vLags = c(1,5,22), vCoef = c(0.5, 0.36 , 0.28 , 0.28 ))
#Estimate the HAR process:
HARModel = HARestimate(vRealizedMeasure = HARSim@Simulation,
vLags = c(1,5,22))
```

HARforecast *HAR forecasting*

Description

Does rolling out of sample forecasting of a HAR model.

Usage

```
HARforecast(vRealizedMeasure, vLags, iNRoll = 10, iNAhead = 10)
```

Arguments

vRealizedMeasure	An xts object containing a realized measure of the integrated volatility.
vLags	A vector denoting which lags should be used in the estimation, standard of c(1,5,22) is in line with Corsi(2009).
iNRoll	How many rolling forecasts should be performed.
iNAhead	The length of each rolling forecast.

Value

Returns a HARforecast, which is a subclass of HARmodel, object containing

1	A HARmodel object estimated from vRealizedMeasure less the rolling forecasts
2	Forecast data frame containing the forecasts. The first row is one-step ahead, the second row is two-step ahead etc. The first column contains the first row, the second column the second roll etc.
3	Info, a list containing information about the forecasting, namely the elapsed time, the rolls done and the horizon of the rolls
4	Data, a list containing ForecastDates, the dates over the rolling forecasts, Observations which is the in-sample observations, and ForecastComparison, containing the observations that are out of sample (for the first roll)

Author(s)

Emil Sjoerup

References

Corsi, F. 2009, A Simple Approximate Long-Memory Model of Realized Volatility, *Journal of Financial Econometrics*, 174–196.

See Also

See Also HARestimate

Examples

```
## Not run:
set.seed(123)
#Simulate a HAR process:
HARSim = HARsimulate(iLength = 10000 , iBurnin = 1000,
vLags = c(1,5,22), vCoef = c(0.5, 0.36 , 0.28 , 0.28 ))
#Do 200 rolling forecasts each of length 200 on the simulated process:
HARForecast = HARforecast(vRealizedMeasure = HARSim@Simulation ,
vLags = c(1, 5, 22), iNRoll = 200 , iNAhead = 200)
#Plot the forecast:
plot(HARForecast)

## End(Not run)
```

HARForecast-class *HARForecast*

Description

Class for HARForecast object

Objects from the Class

A virtual Class: No objects may be created from it

Slots

`Model`: Object of class `HARModel`. see HARModel

`Forecast`: Object of class `data.frame` containing the forecasted series

`Info`: Object of class `list` cointaining:

- `Elapsed Time`: Object of class `difftime` containing the time elapsed in seconds
- `Rolls`: `double` Integer containing the amount of rolls done in the forecasting routine
- `Horizon`: `double` Integer containing the length of the horizon used for forecasting during each of the rolls

`Data`: Object of class `list` containing:

- `Dates`:Object of type `Integer` or `Date` containing the indices of the forecasted series either in integer or date format
- `Observations`:Object of type `double` or `xts` containing the in-sample observations
- `ForecastComparison`: Object of type `double` or `xts` containing the observations kept out of sample

Methods

- `show`: Shows summary
- `plot`: Plot the forecasted series and observed series as well as the residuals
- `uncondmean`: Extracts the unconditional mean from the Model
- `coef`: Extracts the coefficients from the Model

Author(s)

Emil Sjoerup

`HARModel-class` *HARModel*

Description

Class for HARModel objects

Objects from the Class

A virtual Class: No objects may be created from it.

Slots

`Model`: Object of class `lm`. Contains the linear model fitted to which `mVarCovar`, the Newey-West variance-covariance matrix, is appended

`Info`: Object of class `list` cointaining:

- `NWLagOrder`: Integer denoting the used Lag order in the Newey-West Variance Covariance estimation.
- `Lags`: `numeric` containing the lags used to create the model
- `Dates`: `Date` object containing the dates for which the estimation was done, only applicable if the Model was estimated using an "xts" object.

`Data`: `list` object containing the provided data.

Methods

- `show`: Shows summary
- `plot`: Plots the HARModel
- `uncondmean`: Extracts the unconditional mean from the Model
- `coef`: Extracts the coefficients from the Model
- `SandwichNeweyWest`: Utilize the `sandwich` package to create newey west standard errors

Author(s)

Emil Sjoerup

HARMonteCarlo *Monte Carlo simulation of HAR model*

Description

Does simulation and fast estimation of a HAR process. Note that only coefficient estimates are returned.

Usage

```
HARMonteCarlo(iLength = 1000, vLags = c(1, 5, 22),
vCoef = c(1, 0.36, 0.28, 0.28), iBurnin = 100, dSigma = 1,
iLagsPlusOne = length(vLags) + 1)
```

Arguments

iLength	Length of the simulated process.
vLags	Vector of lags for constructing the model, standard is c(1,5,22) which is in line with Corsi(2009)
vCoef	Coefficient vector which will be used to simulate the process.
iBurnin	Integer to determine the length of burnin in the simulation. Increasing this value beyond 100 does not change much, but lowering it CAN be bad.
dSigma	Standard deviation of the error term.
iLagsPlusOne	Integer to denote the amount of lags + 1. This Parameter is used in a non-user-callable function. Setting the function to 4 (in the case of 3 lags) significantly speeds up MonteCarlo simulations.

Details

Does fast estimation but only returns coefficient estimates. This function is created with the intention of making a function that is easy to use with the `parallel` package.

Value

Returns a vector containing the parameter estimates.

Note

Using this function once makes little sense but an example is given none-theless. This function was made with parallel computing in mind and thus an example utilizing the `parallel` package is provided.

Author(s)

Emil Sjoerup

References

Corsi, F. 2009, A Simple Approximate Long-Memory Model of Realized Volatility, *Journal of Financial Econometrics*, 174–196 .

See Also

See Also HARestimate & HARsimulate

Examples

```
## Not run:
#A Single repetition:
HARMonteCarlo = HARMonteCarlo(iLength = 1000, vLags = c(1, 5, 22),
vCoef = c(1, 0.36, 0.28, 0.28), iBurnin = 100, dSigma = 1,
iLagsPlusOne = length(vLags) + 1)

## End(Not run)
```

HARSim-class *HARSim*

Description

Class for HARSim object

Objects from the Class

A virtual Class: No objects may be created from it

Slots

`Simulation`**:** Object of class `numeric` containing the simulated series

`Info`**:** Object of class `list` cointaining:

- `Length`: Object of class `numeric` containing the length of the simulated series
- `Lags`: Object of class `numeric` containing the lag-vector used for simulation
- `Coefficients`: Object of class `numeric` containing the coefficients used for simulation
- `ErrorTermSD`: Object of class `numeric` containing the standard error of the error term
- `Elapsed Time`: Object of class `difftime` containing the time elapsed in seconds

Methods

- `show`: Shows summary
- `plot`: Plot the forecasted series and observed series as well as the residuals
- `uncondmean`: Extracts the unconditional mean from the simulation
- `coef`: Extracts the coefficients from the simulation

Author(s)

Emil Sjoerup

`HARsimulate` *HAR simulation*

Description

Simulates a HAR model. From using the AR representation of the HAR model.

Usage

```
HARsimulate(iLength=10000, vLags = c(1, 5, 22) ,
vCoef = c(0.002, 0.36 ,0.28 , 0.28), dSigma = 0.001)
```

Arguments

`iLength`	Integer length of the simulated process.
`vLags`	Vector of lags for constructing the model, standard is c(1,5,22) which is in line with Corsi(2009)
`vCoef`	Coefficient vector which will be used to simulate the process.
`dSigma`	Standard deviation of the error term.

Value

Returns an S4 object of type `HARSim` which contains:

`Simulation`	The simulated process
`Info`	Information about the simulation (The input parameters) and the elapsed time.

Author(s)

Emil Sjoerup

References

Corsi, F. 2009, A Simple Approximate Long-Memory Model of Realized Volatility, *Journal of Financial Econometrics*, 174–196.

See Also

See Also `HARestimate`

Examples

```
set.seed(123)
#Simulate the process of size 10000
HARSim = HARsimulate(iLength=10000, vLags = c(1, 5, 22),
vCoef = c(0.002, 0.36 ,0.28 , 0.28), dSigma = 0.001)
#plot the Object
plot(HARSim)
```

SandwichNeweyWest *HAC Variance Covariance Matrix For* HARModel *Objects*

Description

A method for extracting the Newey-West variance-covariance matrix for a HARModel object

Usage

```
SandwichNeweyWest(object, lags)
```

Arguments

object	A HARModel object
lags	An integer to denote the allowed autocorrelation.

Details

Usual values for lags are 5 and $T^0.25$ rounding to the nearest integer. This function is merely a method to implement code from the sandwich package.

Value

HACMatrix	The Variance-Covariance matrix

References

sandwich package. https://CRAN.R-project.org/package=sandwich

Examples

```
set.seed(123)
#Simulate a HAR process:
HARSim = HARsimulate(iLength = 10000,
vLags = c(1,5,22), vCoef = c(0., 0.36 , 0.28 , 0.28 )
, dSigma = 0.001)
#Estimate the HAR process:
HARModel = HARestimate(vRealizedMeasure = HARSim@Simulation,
vLags = c(1,5,22))

SandwichNeweyWest(HARModel , lags = 5)
```

uncondmean *Unconditional mean*

Description

Extracts the unconditional mean of a HAR models and other objects

Usage

```
uncondmean(object)
```

Arguments

object	an object from the HARModel package

Value

Unconditional Mean
: a double containing the unconditional mean of the model or simulation

Note

This is a method

Author(s)

Emil Sjoerup

Examples

```
set.seed(123)
#Simulate a HAR process:
HARSim = HARsimulate(iLength = 10000,
vLags = c(1,5,22), vCoef = c(0., 0.36 , 0.28 , 0.28 )
, dSigma = 0.001)
#Estimate the HAR process:
HARModel = HARestimate(vRealizedMeasure = HARSim@Simulation,
vLags = c(1,5,22))
# uncondmean can be used on both Simulation and
# estimated models
uncondmean(HARSim)
uncondmean(HARModel)
```

Index

∗Topic **HAR**
HARforecast, 6
HARMonteCarlo, 9
HARsimulate, 11
uncondmean, 13
∗Topic **Heterogeneous Autoregressive model**
HARModel-package, 2
∗Topic **MonteCarlo**
HARMonteCarlo, 9
∗Topic **Unconditional**
uncondmean, 13
∗Topic **classes**
HARForecast-class, 7
HARModel-class, 8
HARSim-class, 10
∗Topic **datasets**
DJIRealizedMeasures, 3
∗Topic **forecast**
HARforecast, 6
∗Topic **simulation**
HARsimulate, 11

coef,HARForecast-method *(*HARForecast-class*)*, 7
coef,HARModel-method *(*HARModel-class*)*, 8
coef,HARSim-method *(*HARSim-class*)*, 10

DJIRealizedMeasures, 3

HARDataCreationC, 3, *5*
HARestimate, *4*, 4, *6*, *10, 11*
HARforecast, 6
HARForecast-class, 7
HARModel, *7*
HARModel *(*HARModel-package*)*, 2
HARModel-class, 8
HARModel-package, 2
HARMonteCarlo, 9
HARSim-class, 10
HARsimulate, *10*, 11

plot,HARForecast,missing-method *(*HARForecast-class*)*, 7
plot,HARModel,missing-method *(*HARModel-class*)*, 8
plot,HARSim,missing-method *(*HARSim-class*)*, 10

SandwichNeweyWest, 12
SandwichNeweyWest,HARModel-method *(*HARModel-class*)*, 8
show,HARForecast-method *(*HARForecast-class*)*, 7
show,HARModel-method *(*HARModel-class*)*, 8
show,HARSim-method *(*HARSim-class*)*, 10

uncondmean, 13
uncondmean,HARForecast-method *(*HARForecast-class*)*, 7
uncondmean,HARModel-method *(*HARModel-class*)*, 8
uncondmean,HARSim-method *(*HARSim-class*)*, 10