

Package ‘HARModel’

January 24, 2019

Type Package

Title Heterogeneous Autoregressive Models

Version 0.2

Date 2019-01-24

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> and extensions.

BugReports <https://github.com/emilsjoerup/HARModel/issues>

URL <https://github.com/emilsjoerup/HARModel>

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 2019-01-24 16:40:03 UTC

R topics documented:

HARModel-package	2
DJIRM	3
HARestimate	3
HARforecast	5
HARForecast-class	8
HARModel-class	9
HARSim-class	10
HARsimulate	10
SP500RM	11

Index	13
--------------	-----------

 HARModel-package

Heterogeneous Autoregressive Models

Description

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

Details

The DESCRIPTION file:

```

Package:      HARModel
Type:         Package
Title:        Heterogeneous Autoregressive Models
Version:      0.2
Date:         2019-01-24
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
BugReports:   https://github.com/emilsjoerup/HARModel/issues
URL:          https://github.com/emilsjoerup/HARModel
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:

DJIRM	Dow Jones Realized Measures
HARForecast-class	HARForecast
HARModel-class	HARModel
HARModel-package	Heterogeneous Autoregressive Models
HARSim-class	HARSim
HARestimate	HAR estimation
HARforecast	HAR forecasting
HARsimulate	HAR simulation
SP500RM	SP500 Realized Measures

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 .

DJIRM

Dow Jones Realized Measures

Description

Realized measures for the Dow Jones Industrial index from 2001 to september 2018

Format

A large xts object

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

HARestimate

HAR estimation

Description

This function does HAR estimation in line with Corsi(2009)..

Usage

```
HARestimate(vRealizedMeasure, vJumpComponent = NULL, vAuxData = NULL, vLags = c(1,5,22),
            vJumpLags = NULL, vAuxLags = NULL, type = "HAR",
            InsanityFilter = TRUE, HARQargs = list(demean = TRUE ),show=TRUE )
```

Arguments

vRealizedMeasure	A numeric containing a realized measure of the integrated volatility.
vJumpComponent	A numeric containing the jump proportion of the realized measure used for HARJ and HARQ-J types.
vAuxData	A numeric containing the realized quarticity used for HARQ and HARQ-J types.
vLags	A numeric denoting which lags should be used in the estimation, standard of $c(1, 5, 22)$ is in line with Corsi(2009).
vJumpLags	A numeric denoting which lags should be used in Jump estimation, if applicable.
vAuxLags	A numeric denoting which lags should be used in Realized Quarticity estimation, if applicable.
type	A character denoting which type of HAR model to estimate.
InsanityFilter	A logical denoting whether the insanity filter should be used for the fitted values of the estimation see Bollerslev, Patton & Quaedvlieg(2016) footnote 17.
HARQargs	A list denoting the extra arguments of a type HARQ or Full-HARQ model. So far only the boolean demean is implemented, which determines whether the data should be demeaned as is the case in Bollerslev, Patton & Quaedvlieg(2016).
show	Logical to determine whether the output should be shown when done.

Value

A `HARModel` object

Author(s)

Emil Sjoerup

References

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

Bollerslev, T., Patton, A., Quaedvlieg, R. 2016, Exploiting the errors: A simple approach for improved volatility forecasting, *Journal of Econometrics*, vol.192, issue 1, 1-18.

Examples

```
#Vanilla HAR from Corsi(2009)
#load data
data("SP500RM")
SP500rv = SP500RM$RV
#Estimate the HAR model:
FitHAR = HARestimate(vRealizedMeasure = SP500rv, vLags = c(1,5,22))

#HAR-J:
#load data
```

```

data("SP500RM")
SP500rv = SP500RM$RV
SP500bpv = SP500RM$BPV

vJumpComponent = SP500rv - SP500bpv
vJumpComponent = ifelse(vJumpComponent>=0, vJumpComponent, 0)
#Estimate the HAR-J model:
FithARJ = HARestimate(vRealizedMeasure = SP500rv, vJumpComponent = vJumpComponent,
                     vLags = c(1,5,22), vJumpLags = c(1,5,22), type = "HARJ" )

#HAR-Q of BPQ(2016)
#load data
data("SP500RM")
SP500rv = SP500RM$RV
SP500rq = SP500RM$RQ
#Estimate the HAR-Q model:
FithARQ = HARestimate(vRealizedMeasure = SP500rv, vAuxData = SP500rq, vLags = c(1,5,22),
                     vAuxLags = c(1,5,22), type = "HARQ", HARQargs = list(demean =TRUE))

#HARQ-J of BPQ(2016)
#load data
data("SP500RM")
SP500rv = SP500RM$RV
SP500rq = SP500RM$RQ
SP500bpv = SP500RM$BPV

vJumpComponent = SP500rv - SP500bpv
vJumpComponent = ifelse(vJumpComponent>=0, vJumpComponent, 0)
#Estimate the HARQ-J model:
FithARQJ = HARestimate(vRealizedMeasure = SP500rv, vJumpComponent = vJumpComponent,
                      vAuxData = SP500rq, vLags = c(1,5,22),
                      vJumpLags = c(1,5,22), vAuxLags = c(1),
                      type = "HARQ-J", HARQargs = list(demean = TRUE))

```

HARforecast

HAR forecasting

Description

Does rolling out of sample forecasting of a HAR model.

Usage

```

HARforecast(vRealizedMeasure, vJumpComponent= NULL, vAuxData = NULL , vLags = c(1,5,22),
            vJumpLags = NULL, vAuxLags = NULL, iNRoll=10 , iNAhead=10 , type = "HAR",
            InsanityFilter = TRUE , HARQargs = list(demean = T))

```

Arguments

vRealizedMeasure	An xts object containing a realized measure of the integrated volatility.
vJumpComponent	A numeric containing the jump proportion of the realized measure used for HARJ and HARQ-J types.
vAuxData	A numeric containing the realized quarticity used for HARQ and HARQ-J types.
vLags	A vector denoting which lags should be used in the estimation, standard of $c(1, 5, 22)$ is in line with Corsi(2009).
vJumpLags	A numeric denoting which lags should be used in Jump estimation, if applicable.
vAuxLags	A numeric denoting which lags should be used in Realized Quarticity estimation, if applicable.
iNRoll	How many rolling forecasts should be performed.
iNAhead	The length of each rolling forecast.
type	A character denoting which type of HAR model to estimate.
InsanityFilter	A logical denoting whether the insanity filter should be used for the forecasted values see Bollerslev, Patton & Quaedvlieg(2016) footnote 17.
HARQargs	A list denoting the extra arguments of a type HARQ model. So far only the boolean demean is implemented, which determines whether the data should be demeaned as is the case in Bollerslev, Patton & Quaedvlieg(2016)

Details

Only the HAR type model is complete, which means to model more than one period ahead the realized quarticity and the jump component must be modeled separately. This is done by AR(1) processes in both cases.

The maximum lag of the jump or auxiliary data must be lower than the maximum of the realized measure lag vector, the other cases are not implemented yet.

Value

A `HARForecast` object

Author(s)

Emil Sjoerup

References

- Corsi, F. 2009, A Simple Approximate Long-Memory Model of Realized Volatility, *Journal of Financial Econometrics*, 174–196.
- Bollerslev, T., Patton, A., Quaedvlieg, R. 2016, Exploiting the errors: A simple approach for improved volatility forecasting, *Journal of Econometrics*, vol.192, issue 1, 1-18.

See Also

See Also [HARestimate](#)

Examples

```
#HAR of Corsi(2009)
#load data:
data("SP500RM")
SP500rv = SP500RM$RV

ForecastHAR = HARforecast(SP500rv, vLags = c(1,5,22), iNRoll =50,
                          iNAhead = 50, type = "HAR")

#HARJ
#load data:
data("SP500RM")
SP500rv = SP500RM$RV
SP500bpv = SP500RM$BPV
vJumpComponent = SP500rv - SP500bpv
vJumpComponent = ifelse(vJumpComponent>=0, vJumpComponent, 0)

ForecastHARJ = HARforecast(SP500rv, vJumpComponent = vJumpComponent, vLags = c(1,5,22),
                          vJumpLags = c(1,5,22) ,iNRoll = 50,
                          iNAhead = 50, type = "HARJ")

#HARQ BPQ(2016)
#load data
data("SP500RM")
SP500rv = SP500RM$RV
SP500rq = SP500RM$RQ

ForecastHARQ = HARforecast(SP500rv, vAuxData= SP500rq, vLags = c(1,5,22),
                          vAuxLags = c(1,5,22), iNRoll = 50, iNAhead = 50,
                          type = "HARQ")

#HARQ-J BPQ(2016)
#load data
data("SP500RM")
SP500rv = SP500RM$RV
SP500rq = SP500RM$RQ
SP500bpv = SP500RM$BPV
vJumpComponent = SP500rv - SP500bpv
vJumpComponent = ifelse(vJumpComponent>=0, vJumpComponent, 0)

ForecastHARQJ = HARforecast(SP500rv, vAuxData = SP500rq, vJumpComponent = vJumpComponent,
```

```
vLags = c(1,5,22), vJumpLags = c(1,5,22),
vAuxLags = c(1,5,22), iNRoll = 50,
iNAhead = 50, type = "HARQ-J")
```

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 `matrix` containing the forecasted series

Info: Object of class `list` containing:

- **Elapsed Time:** Object of class `difftime` containing the time elapsed in seconds
- **Rolls:** Integer containing the amount of rolls done in the forecasting routine
- **Horizon:** 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 `numeric` or `xts` containing the in-sample observations
- **ForecastComparison:** Object of type `numeric` or `xts` containing the observations kept out of sample for the first roll

Methods

show: `signature(object = "HARForecast")`: Shows summary

plot: `signature(x = "HARForecast", y = "missing")`: Plot the out of sample observed series with the forecasts overlaid

uncmean: `signature(object = "HARForecast")`: Extracts the unconditional mean from the Model

coef: `signature(object = "HARForecast")`: Extracts the coefficients from the first estimated Model

Author(s)

Emil Sjoerup

HARModel-class	<i>HARModel</i>
----------------	-----------------

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.

Info: Object of class `list` containing:

- **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: `signature(object = "HARModel")` Shows summary

plot: `signature(x = "HARModel", y = "missing")`: Plots the observed values with fitted values overlaid

uncmean: `signature(object = "HARModel")`: Extracts the unconditional mean from the Model

coef: `signature(object = "HARModel")`: Extracts the coefficients from the Model

SandwichNeweyWest: `signature(object = "HARModel")`: Utilize the sandwich package to create newey west standard errors

Author(s)

Emil Sjoerup

HARSim-class	<i>HARSim</i>
--------------	---------------

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 containing:

- 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: signature(object = "HARSim"): Shows summary

plot: signature(x = "HARSim" , y = "missing"): Plot the forecasted series and observed series as well as the residuals

uncmean: signature(object = "HARSim"): Extracts the unconditional mean from the simulation

coef: signature(object = "HARSim"): Extracts the coefficients from the simulation

Author(s)

Emil Sjoerup

HARsimulate	<i>HAR simulation</i>
-------------	-----------------------

Description

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

Usage

```
HARsimulate(iLength=1500, vLags = c(1, 5, 22),
vCoef = c(0.01, 0.36 ,0.28 , 0.28), dSigma = 0.001, show = TRUE)
```

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.
show	Logical to determine whether the output should be shown when done.

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.

Examples

```
set.seed(123)
#Simulate the process of size 10000
HARSim = HARSimulate(iLength=10000, vLags = c(1, 5, 22),
vCoef = c(0.01, 0.36, 0.28, 0.28), dSigma = 0.001)

HARFit = HARestimate(HARSim@Simulation, vLags = c(1, 5, 22))
```

SP500RM

SP500 Realized Measures

Description

Realized measures from the SP500 index from April 1997 to August 2013.

Format

A large xts object.

Source

<http://public.econ.duke.edu/~ap172/code.html>

References

Bollerslev, T., A. J. Patton, and R. Quaedvlieg, 2016, Exploiting the Errors: A Simple Approach for Improved Volatility Forecasting, *Journal of Econometrics*, 192, 1-18.

Index

*Topic **HAR**

HARforecast, 5
HARsimulate, 10

*Topic **Heterogeneous Autoregressive model**

HARModel-package, 2

*Topic **classes**

HARForecast-class, 8
HARModel-class, 9
HARSim-class, 10

*Topic **datasets**

DJIRM, 3
SP500RM, 11

*Topic **forecast**

HARforecast, 5

*Topic **simulation**

HARsimulate, 10

coef, HARForecast-method
(HARForecast-class), 8

coef, HARModel-method (HARModel-class), 9

coef, HARSim-method (HARSim-class), 10

DJIRM, 3

HARestimate, 3, 7

HARForecast, 6

HARforecast, 5

HARForecast-class, 8

HARModel, 4, 8

HARModel (HARModel-package), 2

HARModel-class, 9

HARModel-package, 2

HARSim-class, 10

HARsimulate, 10

plot, HARForecast, missing-method
(HARForecast-class), 8

plot, HARModel, missing-method
(HARModel-class), 9

plot, HARSim, missing-method
(HARSim-class), 10

SandwichNeweyWest (HARModel-class), 9

SandwichNeweyWest, ANY-method
(HARModel-class), 9

SandwichNeweyWest, HARModel-method
(HARModel-class), 9

show, HARForecast-method
(HARForecast-class), 8

show, HARModel-method (HARModel-class), 9

show, HARSim-method (HARSim-class), 10
SP500RM, 11

uncmean (HARModel-class), 9

uncmean, ANY-method (HARModel-class), 9

uncmean, HARForecast-method
(HARForecast-class), 8

uncmean, HARModel-method
(HARModel-class), 9

uncmean, HARSim-method (HARSim-class), 10