

# Package ‘HARModel’

October 12, 2022

**Type** Package

**Title** Heterogeneous Autoregressive Models

**Version** 1.0

**Date** 2019-08-30

**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

**Suggests** testthat

**Repository** CRAN

**Date/Publication** 2019-08-31 11:30:02 UTC

## R topics documented:

HARModel-package	2
DJIRM	3
HAREstimate	3
HARForecast	6
HARForecast-class	9
HARModel-class	10
HARSim-class	11
HARSimulate	11
SP500RM	12

<b>Index</b>	<b>13</b>
--------------	-----------

---

 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:      1.0
Date:         2019-08-30
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
Suggests:     testthat
  
```

Index of help topics:

DJIRM	Dow Jones Realized Measures
HAREstimate	HAR estimation
HARForecast	HAR forecasting
HARForecast-class	HARForecast
HARModel-class	HARModel
HARModel-package	Heterogeneous Autoregressive Models
HARSim-class	HARSim
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

HAR estimation

## Usage

```
HAREstimate(RM, BPV = NULL, RQ = NULL, periods = c(1,5,22),
             periodsJ = NULL, periodsRQ = NULL, type = "HAR",
             insanityFilter = TRUE, h = 1)
```

**Arguments**

RM	A numeric containing a realized measure of the integrated volatility.
BPV	A numeric containing the estimate of the continuous part of the integrated volatility used for HARJ and HARQ-J types.
RQ	A numeric containing the realized quarticity used for HARQ and HARQ-J types.
periods	A numeric denoting which lags should be used in the estimation, standard of $c(1, 5, 22)$ is in line with Corsi(2009).
periodsJ	A numeric denoting which lags should be used in Jump estimation, if applicable.
periodsRQ	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 & Quaadvlieg(2016) footnote 17.
h	A integer denoting the whether and how much to aggregate the realized variance estimator, if $h = 5$ the model is for the weekly volatility and if $h = 22$ , the model is for the monthly volatility, the default of 1 designates no aggregation.

**Details**

The estimates for the HARQ and HARQ-J models differ slightly from the results of BPQ (2016). This is due to a small difference in the demeaning approach for the realized quarticity. Here, the demeaning is done with  $\text{mean}(RQ)$  over all periods.

**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., Quaadvlieg, 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(RM = SP500rv, periods = c(1,5,22))

#extract the estimated coefficients:
coef(FitHAR)
#plot the fitted values
plot(FitHAR)

#calculate the Q-like loss-function:
mean(qlike(FitHAR))

#HAR-J:
#load data
data("SP500RM")
SP500rv = SP500RM$RV
SP500bpv = SP500RM$BPV

#Estimate the HAR-J model:
FitHARJ = HAREstimate(RM = SP500rv, BPV = SP500bpv,
                     periods = c(1,5,22), periodsJ = c(1,5,22), type = "HARJ" )

#Calculate the Q-like loss-function:
mean(qlike(FitHARJ))

#HAR-Q of BPQ(2016) with weekly aggregation
#load data
data("SP500RM")
SP500rv = SP500RM$RV
SP500rq = SP500RM$RQ
#Estimate the HAR-Q model:
FitHARQ = HAREstimate(RM = SP500rv, RQ = SP500rq, periods = c(1,5,22),
                     periodsRQ = c(1,5,22), type = "HARQ", h = 5)

#Show the model:
show(FitHARQ)

#Extract the coefficients:
HARQcoef = coef(FitHARQ)

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

```
#Estimate the HARQ-J model:
FitHARQJ = HAREstimate(RM = SP500rv, BPV = SP500bpv,
                      RQ = SP500rq, periods = c(1,5,22),
                      periodsJ = c(1), periodsRQ = c(1),
                      type = "HARQ-J", h = 22)

#show the model:
show(FitHARQJ)
```

---

HARForecast

*HAR forecasting*


---

### Description

Rolling out of sample forecasting of a HAR model.

### Usage

```
HARForecast(RM, BPV= NULL, RQ = NULL , periods = c(1,5,22),
            periodsJ = NULL, periodsRQ = NULL, nRoll=10 , nAhead=1 , type = "HAR",
            windowType = "rolling", insanityFilter = TRUE, h = 1)
```

### Arguments

RM	An xts object containing a realized measure of the integrated volatility.
BPV	A numeric containing the jump proportion of the realized measure used for HARJ and HARQ-J types.
RQ	A numeric containing the realized quarticity used for HARQ and HARQ-J types.
periods	A vector denoting which lags should be used in the estimation, standard of c(1,5,22) is in line with Corsi(2009).
periodsJ	A numeric denoting which lags should be used in Jump estimation, if applicable.
periodsRQ	A numeric denoting which lags should be used in Realized Quarticity estimation, if applicable.
nRoll	How many rolling forecasts should be performed.
nAhead	The length of each rolling forecast.
type	A character denoting which type of HAR model to estimate.
windowType	A character denoting which kind of window to use, either "rolling"/"fixed" or "increasing"/"expanding". 2-letter abbreviations can be used.
insanityFilter	A logical denoting whether the insanity filter should be used for the forecasted values see Bollerslev, Patton & Quaedvlieg(2016) footnote 17.
h	A integer denoting the whether and how much to aggregate the realized variance estimator, if h = 5 the model is forecasting the weekly volatility and if h = 22, the model is forecasting the monthly volatility, the default of 1 designates no aggregation..

## Details

Not all models in this package are 'complete', which means some models use AR(1) processes to forecast e.g. realized quarticity in order to construct more than one step ahead forecasts.

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

The estimates for the HARQ and HARQ-J models differ slightly from the results of BPQ (2016). This is due to a small difference in the demeaning approach for the realized quarticity. Here, the demeaning is done with  $\text{mean}(RQ)$  over all periods.

If  $h$  is greater than 1, then  $nAhead$  must be one, as multi-period ahead forecasts have not been implemented.

## 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, periods = c(1,5,22), nRoll =50,
                          nAhead = 50, type = "HAR")

#plot the forecasted series along with the actual realizations:
plot(ForecastHAR)

#Calculate the MSE:
mean(forecastRes(ForecastHAR)^2)

#Calculate the Q-like loss function:
mean(qlike(ForecastHAR))
```





---

HARForecast-class	<i>HARForecast</i>
-------------------	--------------------

---

**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:

- **elapsedTime:** 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

**qlike:** `signature(object = "HARForecast")`: Calculate the out of sample 'qlike' loss function for a HARForecast object

**forecastres:** `signature(object = "HARForecast")`: Retrieve the forecast residuals from HARForecast object

**forc:** `signature(object = "HARForecast")`: Retrieve the forecasted series.

**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.

info: Object of class list containing:

- periods: numeric containing the lags used to create the model. If the type isn't "HAR", then the related periods-(RQ) and/or (J) will also be included.
- dates: Date object containing the dates for which the estimation was done, only applicable if the Model was estimated using an "xts" object.

### 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, only available when type = "HAR"

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

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

qlike: signature(object = "HARModel"): Calculate the in sample 'qlike' loss function for a HARModel object

logLik: A wrapper for the "lm" subclass of the HARModel object

confint: A wrapper for the "lm" subclass of the HARModel object

residuals: A wrapper for the "lm" subclass of the HARModel object

summary: A wrapper for the "lm" subclass of the HARModel object

### 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:

- len: Object of class `numeric` containing the length of the simulated series
- periods: 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
- elapsedTime: 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(len=1500, periods = c(1, 5, 22),
            coef = c(0.01, 0.36 ,0.28 , 0.28), errorTermSD = 0.001)
```

**Arguments**

len	An integer determining the length of the simulated process.
periods	A numeric of lags for constructing the model, standard is c(1,5,22).
coef	A numeric of coefficients which will be used to simulate the process.
errorTermSD	A numeric determining the standard deviation of the error term.

**Value**

A `HARSim` object

**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(len = 10000, periods = c(1, 5, 22),
                    coef = c(0.01, 0.36, 0.28, 0.28), errorTermSD = 0.001)
HARFit = HAREstimate(HARSim@simulation, periods = 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

- \* **HAR**
  - HARForecast, 6
  - HARSimulate, 11
- \* **Heterogeneous Autoregressive model**
  - HARModel-package, 2
- \* **classes**
  - HARForecast-class, 9
  - HARModel-class, 10
  - HARSim-class, 11
- \* **datasets**
  - DJIRM, 3
  - SP500RM, 12
- \* **forecast**
  - HARForecast, 6
- \* **simulation**
  - HARSimulate, 11
- coef, HARForecast-method (HARForecast-class), 9
- coef, HARModel-method (HARModel-class), 10
- coef, HARSim-method (HARSim-class), 11
- confint, HARModel-method (HARModel-class), 10
- DJIRM, 3
- fitted.values, HARModel-method (HARModel-class), 10
- forecastRes (HARForecast-class), 9
- forecastRes, ANY-method (HARForecast-class), 9
- forecastRes, HARForecast-method (HARForecast-class), 9
- getForc (HARForecast-class), 9
- getForc, ANY-method (HARForecast-class), 9
- getForc, HARForecast-method (HARForecast-class), 9
- HAREstimate, 3, 7
- HARForecast, 6, 7
- HARForecast-class, 9
- HARModel, 4, 9
- HARModel (HARModel-package), 2
- HARModel-class, 10
- HARModel-package, 2
- HARSim, 12
- HARSim-class, 11
- HARSimulate, 11
- logLik, HARModel-method (HARModel-class), 10
- plot, HARForecast, missing-method (HARForecast-class), 9
- plot, HARModel, missing-method (HARModel-class), 10
- plot, HARSim, missing-method (HARSim-class), 11
- qlike (HARModel-class), 10
- qlike, ANY-method (HARModel-class), 10
- qlike, HARForecast-method (HARForecast-class), 9
- qlike, HARModel-method (HARModel-class), 10
- residuals, HARModel-method (HARModel-class), 10
- sandwichNeweyWest (HARModel-class), 10
- sandwichNeweyWest, ANY-method (HARModel-class), 10
- sandwichNeweyWest, HARModel-method (HARModel-class), 10
- show, HARForecast-method (HARForecast-class), 9
- show, HARModel-method (HARModel-class), 10
- show, HARSim-method (HARSim-class), 11

SP500RM, [12](#)

summary, HARModel-method  
(HARModel-class), [10](#)

uncmean (HARModel-class), [10](#)

uncmean, ANY-method (HARModel-class), [10](#)

uncmean, HARForecast-method  
(HARForecast-class), [9](#)

uncmean, HARModel-method  
(HARModel-class), [10](#)

uncmean, HARSim-method (HARSim-class), [11](#)