Package ‘dccmidas’

March 15, 2021

Type Package

Title DCC Models with GARCH-MIDAS Specifications in the Univariate Step

Version 0.1.0


License GPL-3

LinkingTo Rcpp, RcppArmadillo

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

RdMacros Rdpack

Depends R (>= 4.0.0)

Imports Rcpp (>= 1.0.5), maxLik (>= 1.3-8), rumidas (>= 0.1.1), rugarch (>= 1.4-4), roll (>= 1.1.4), xts (>= 0.12.0), tseries (>= 0.10.47), Rdpack (>= 1.0.0), lubridate (>= 1.7.9), zoo (>= 1.8.8), stats (>= 4.0.2), utils (>= 4.0.2)

Suggests knitr, rmarkdown

NeedsCompilation yes

Author Vincenzo Candila [aut, cre]

Maintainer Vincenzo Candila <vincenzo.candila@uniroma1.it>

Repository CRAN

Date/Publication 2021-03-15 10:00:07 UTC
R topics documented:

```
cov_eval ................................................................. 2
dcc_fit ................................................................. 3
Det ................................................................. 7
ftse100 ................................................................. 7
indpro ................................................................. 8
Inv ................................................................. 9
nasdaq ................................................................. 9
plot_dccmidas .......................................................... 10
sp500 ................................................................. 11
```

Index

```
cov_eval Var-cov matrix evaluation
```

Description

Evaluates the estimated var-cov matrix $H_t$ with respect to a covariance proxy, under different robust loss functions (Laurent et al. 2013). The losses considered are also used in Amendola et al. (2020).

Usage

```
cov_eval(H_t, cov_proxy = NULL, r_t = NULL, loss = "FROB")
```

Arguments

- **H_t**: Estimated covariance matrix, formatted as array
- **cov_proxy**: Optional Covariance matrix, formatted as array
- **r_t**: Optional List of daily returns used to calculate $H_t$. If parameter `cov_proxy` is not provided, then $r_t$ must be included. In this case, a (noise) proxy will be automatically used
- **loss**: Robust loss function to use. Valid choices are: "FROB" for Frobenius (by default), "SFROB" for Squared Frobenius, "EUCL" for Euclidean, "QLIKE" for QLIKE and "RMSE" for Root Mean Squared Errors

Value

The value of the loss for each $t$
References


Examples

```r
require(xts)
# open to close daily log-returns
r_t_s<-log(sp500['2010/2019'][,3])-log(sp500['2010/2019'][,1])
r_t_n<-log(nasdaq['2010/2019'][,3])-log(nasdaq['2010/2019'][,1])
r_t_f<-log(ftse100['2010/2019'][,3])-log(ftse100['2010/2019'][,1])
db_m<-merge.xts(r_t_s,r_t_n,r_t_f)
db_m<-db_m[complete.cases(db_m),]
colnames(db_m)<-c("S&P500","NASDAQ","FTSE100")
# list of returns
r_t<-list(db_m[,1],db_m[,2],db_m[,3])
# estimation
K_c<-144
N_c<-36
cdcc_est<-dcc_fit(r_t,univ_model="sGARCH",distribution="norm",
corr_model="DCCMIDAS",N_c=N_c,K_c=K_c)
cov_eval(cdcc_est$H_t,r_t=r_t)[(K_c+1):dim(cdcc_est$H_t)[3]]
```

---

dcc_fit

**DCC fit (first and second steps)**

**Description**

Obtains the estimation of a variety of DCC models, using as univariate models both GARCH and GARCH-MIDAS specifications.

**Usage**

dcc_fit(
  r_t,
  univ_model = "sGARCH",
  distribution = "norm",
  MV = NULL,
  K = NULL,
  corr_model = "cDCC",
  lag_fun = "Beta",
)
Arguments

`r_t` List of daily returns on which estimate a DCC model. Each daily return must be an 'xts' object. Note that the sample period of the returns should be the same. Otherwise, a merge is performed.

`univ_model` Specification of the univariate model. Valid choices are: some of the specifications used in the `rugarch` (`ugarchspec`) and `rumidas` (`ugmfit`) packages. More in detail, the models coming from `rugarch` are: 'sGARCH', 'eGARCH', 'gjrGARCH', 'iGARCH', and 'csGARCH'. The models implemented from `rumidas` are: 'GM_skew', 'GM_noskew', 'DAGM_skew', and 'DAGM_noskew'.

`distribution` optional Distribution chosen for the univariate estimation. Valid choices are: "norm" (by default) and "std", respectively, for the Normal and Student’s t distributions.

`MV` optional MIDAS variable to include in the univariate estimation, if the model specified is a GARCH-MIDAS (GM, Engle et al. (2013)) or a Double Asymmetric GM (DAGM, Engle et al. (2013)). In the case of MIDAS-based models, please provide a list of the MIDAS variables obtained from the `mv_into_mat` function. If the same MV variable is used, then provide always a list, with the same (transformed) variable repeated.

`K` optional The number of lagged realizaiton of MV variable to use, if `univ_model` has a MIDAS term.

`corr_model` Correlation model used. Valid choices are: "cDCC" (the corrected DCC of Aielli (2013)), "aDCC" (the asymmetric DCC model of Cappiello et al. (2006)), "DECO" (Dynamic equicorrelation of Engle and Kelly (2012)), and "DCCMIDAS" (the DCC-MIDAS of Colacito et al. (2011)). By default, it is "cDCC".

`lag_fun` optional. Lag function to use. Valid choices are "Beta" (by default) and "Almon", for the Beta and Exponential Almon lag functions, respectively, if `corr_model` has a MIDAS term and/or if `corr_model` is "DCCMIDAS".

`N_c` optional Number of (lagged) realizations to use for the standarized residuals forming the long-run correlation, if `corr_model` is "DCCMIDAS".

`K_c` optional Number of (lagged) realizations to use for the long-run correlation, if `corr_model` is "DCCMIDAS".

Details

Function `dcc_fit` implements the two-steps estimation of the DCC models. In the first step, a variety of univariate models are considered. These models can be selected using for the parameter `univ_model` one of the following choices: 'sGARCH' (standard GARCH of Bollerslev (1986)), 'eGARCH' of Nelson (1991), 'gjrGARCH' of Glosten et al. (1993), 'iGARCH' (Integrated GARCH of Engle and Bollerslev (1986)), 'csGARCH' (the Component GARCH of Engle and Lee (1999)), 'GM_noskew' and 'GM_skew' (the GARCH-MIDAS model of Engle et al. (2013), respectively, without and with the asymmetric term in the short-run component), and 'DAGM_noskew'.
and 'DAGM_skew' (the Double Asymmetric GARCH-MIDAS model of Amendola et al. (2019), respectively, without and with the asymmetric term in the short-run component).

Value
dcc_fit returns an object of class 'dccmidas'. The function summary.dccmidas can be used to print a summary of the results. Moreover, an object of class 'dccmidas' is a list containing the following components:

- assets: Names of the assets considered.
- model: Univariate model used in the first step.
- est_univ_model: List of matrixes of estimated coefficients of the univariate model, with the QML (Bollerslev and Wooldridge 1992) standard errors.
- corr_coef_mat: Matrix of estimated coefficients of the correlation model, with the QML standard errors.
- mult_model: Correlation model used in the second step.
- obs: The number of daily observations used for the estimation.
- period: The period of the estimation.
- H_t: Conditional covariance matrix, reported as an array.
- R_t: Conditional correlation matrix, reported as an array.
- R_t_bar: Conditional long-run correlation matrix, reported as an array, if the correlation matrix includes a MIDAS specification.
- est_time: Time of estimation.
- Days: Days of the sample period.
- llk: The value of the log-likelihood (for the second step) at the maximum.

References


Examples

```r
require(xts)
# open to close daily log-returns
r_t_s<-log(sp500[2005/2008][,3])-log(sp500[2005/2008][,1])
r_t_n<-log(nasdaq[2005/2008][,3])-log(nasdaq[2005/2008][,1])
r_t_f<-log(ftse100[2005/2008][,3])-log(ftse100[2005/2008][,1])
db_m<-merge.xts(r_t_s,r_t_n,r_t_f)
db_m<-db_m[complete.cases(db_m),]
colnames(db_m)<-c("S&P500","NASDAQ","FTSE100")
# list of returns
r_t<-list(db_m[,1],db_m[,2],db_m[,3])
# MV transformation (same MV for all the stocks)
require(rumidas)
mv_m<-mv_into_mat(r_t[[1]],diff(indpro),K=12,"monthly")
# list of MV
MV<-list(mv_m,mv_m,mv_m)
# estimation
K_c<-144
N_c<-36
dccmidas_est<-dcc_fit(r_t,univ_model="GM_noskew",distribution="norm",
MV=MV,K=12,corr_model="DCCMIDAS",N_c=N_c,K_c=K_c)
dccmidas_est
summary.dccmidas(dccmidas_est)
```
**Det**

*Matrix determinant*

**Description**

Calculates the determinant of a numeric matrix.

**Usage**

\[ \text{Det}(x) \]

**Arguments**

- **x**: a numeric matrix

**Value**

The determinant of \( x \).

**Examples**

\[
x \leftarrow \text{matrix}(\text{sample}(1:25,25,\text{replace}=\text{TRUE}),\text{ncol}=5)
\]

\[
\text{Det}(x)
\]

---

**ftse100**

*FTSE 100 data*

**Description**

Daily data on FTSE 100 collected from the realized library of the Oxford-Man Institute (Heber et al. 2009).

**Usage**

\[ \text{data(ftse100)} \]

**Format**

An object of class "xts".

**Details**

\text{ftse100} includes the open price (\text{open\_price}), the realized variance (\text{rv5}), and the close price (\text{close\_price}). The realized variance has been calculated using intradaily intervals of five minutes (Andersen and Bollerslev 1998).
Source
Realized library of the Oxford-Man Institute

References


Examples
head(ftse100)
summary(ftse100)

indpro Monthly U.S. Industrial Production

Description
Monthly data on the U.S. Industrial Production index (IP, index 2012=100, seasonally adjusted) collected from the Federal Reserve Economic Data (FRED) archive. The IP has been used as MIDAS term in different contributions (see, for instance, Engle et al. (2013), Conrad and Loch (2015), and Amendola et al. (2017)).

Usage
data(indpro)

Format
An object of class "xts".

Source
Archive of the Federal Reserve Economic Data (FRED)

References


**Inv**

*Inverse of a matrix*

**Examples**

```r
head(indpro)
summary(indpro)
plot(indpro)
```

**Description**

Calculates the inverse of a numeric matrix.

**Usage**

```r
Inv(x)
```

**Arguments**

- `x`: a numeric matrix.

**Value**

The inverse of `x`.

**Examples**

```r
x<-matrix(sample(1:25,25,replace=TRUE),ncol=5)
Inv(x)
```

---

**nasdaq**

*NASDAQ data*

**Description**


**Usage**

```r
data(nasdaq)
```

**Format**

An object of class "xts".
Details
	nasdaq includes the open price (open_price), the realized variance (rv5), and the close price (close_price). The realized variance has been calculated using intradaily intervals of five minutes (Andersen and Bollerslev 1998).

Source

Realized library of the Oxford-Man Institute

References


Examples

head(nasdaq)
summary(nasdaq)

plot_dccmidas

Plot method for ‘dccmidas’ class

Description

Plots of the conditional volatilities on the main diagonal and of the conditional correlations on the extra-diagonal elements.

Usage

plot_dccmidas(
  x,
  K_c = NULL,
  vol_col = "black",
  long_run_col = "red",
  cex_axis = 0.75,
  LWD = 2,
  asset_sub = NULL
)
Arguments

- **x**: An object of class 'dccmidas', that is the result of a call to `dcc_fit`.
- **K_c**: *optional* Number of (lagged) realizations to use for the long-run correlation, if 'corr_model' is "DCCMIDAS"
- **vol_col**: *optional* Color of the volatility and correlation plots. "black" by default
- **long_run_col**: *optional* Color of the long-run correlation plots, if present. "red" by default
- **cex_axis**: *optional* Size of the x-axis. Default to 0.75
- **LWD**: *optional* Width of the plotted lines. Default to 2
- **asset_sub**: *optional* Numeric vector of selected assets to consider for the plot. NULL by default

Value

No return value, called for side effects

Examples

```r
require(xts)
# open to close daily log-returns
r_t_s<-log(sp500['2010/2019',[3]]-log(sp500['2010/2019',[1]])
r_t_n<-log(nasdaq['2010/2019',[3]]-log(nasdaq['2010/2019',[1]])
r_t_f<-log(ftse100['2010/2019',[3]]-log(ftse100['2010/2019',[1]])
db_m<-merge.xts(r_t_s,r_t_n,r_t_f)
db_m<-db_m[complete.cases(db_m),]
colnames(db_m)<-c("S&P500","NASDAQ","FTSE100")
# list of returns
r_t<-list(db_m[,1],db_m[,2],db_m[,3])
# estimation
K_c<-144
N_c<-36
cdcc_est<-dcc_fit(r_t,univ_model="sGARCH",distribution="norm",corr_model="DCCMIDAS",N_c=N_c,K_c=K_c)
plot_dccmidas(cdcc_est,K_c=144)
```

sp500

*S&P 500 data*

Description


Usage

data(sp500)
Format

An object of class "xts".

Details

sp500 includes the open price (open_price), the realized variance (rv5), and the close price (close_price). The realized variance has been calculated using intradaily intervals of five minutes (Andersen and Bollerslev 1998).

Source

Realized library of the Oxford-Man Institute

References


Examples

head(sp500)
summary(sp500)
Index

* datasets
  ftse100, 7
  indpro, 8
  nasdaq, 9
  sp500, 11

cov_eval, 2

dcc_fit, 3, 11
  Det, 7

ftse100, 7

indpro, 8
  Inv, 9

mv_into_mat, 4

nasdaq, 9

plot_dccmidas, 10

sp500, 11
  summary.dccmidas, 5

ugarchspec, 4
  ugmfit, 4