Package ‘Spillover’

June 17, 2021

Type Package
Title Spillover Index Based on VAR Modelling
Version 0.1.0.1
Date 2021-06-17
Author Jilber Urbina
Maintainer Jilber Urbina <jurbina@cipadla.com>
License GPL-2
Depends R (>= 3.5.0), vars, zoo, fastSOM
Encoding UTF-8
RoxygenNote 7.1.1
NeedsCompilation no
Repository CRAN
Date/Publication 2021-06-17 16:30:05 UTC

R topics documented:

   dy2009 ................................................................. 2
ey2012 ................................................................. 2
g.fevd ................................................................. 3
g.spillover .......................................................... 4
net ................................................................. 5
O.spillover .......................................................... 6
rol Returns ......................................................... 8
rol.vol .............................................................. 8
roll.net ............................................................. 9
Description


Format

a data.frame-class dataset

References


Examples

data(dy2009)
head(dy2009)
summary(dy2009)  # Same as Diebold and Yilmaz (2012) summary statistics

Description

A dataset consisting of 2771 log volatility daily observations of 4 variables: Stocks (SP500), Bonds (R_10Y), Commodities (DJUBSCOM) and FX (USDX). The period for this dataset is from Jan 25, 1999 to Jan 29, 2010.

Format

a data.frame-class dataset

References

Examples

data(dy2012)
head(dy2012)
summary(dy2012)  # Same as Diebold and Yilmaz (2012) summary statistics

---

**g.fevd**  
*Generalized Forecast Error Variance Decomposition*

**Description**

Computes the generalized forecast error variance decomposition of a VAR(p) for n.ahead steps.

**Usage**

```r
g.fevd(x, n.ahead = 10, normalized = TRUE)
```

**Arguments**

- `x`  
  Object of class `varest` generated by `VAR()` from vars package.
- `n.ahead`  
  Integer specifying the steps ahead.
- `normalized`  
  a logical value indicating whether the result should be normalized to sum up to 1, see Details

**Details**

When `normalized=FALSE` this function computes the generalized forecast error variance decomposition proposed by Pesaran and Shin (1998) which takes the form:

\[
\alpha_{ij}^g(h) = \frac{\sigma_{ii}^{-1} \sum_{l=0}^{h-1} (e_i' \Theta_l \Sigma \varepsilon e_j)^2}{\sum_{l=0}^{h-1} (e_i' \Theta_l \Sigma \varepsilon e_i)}, \quad i, j = 0, 1, 2 \ldots, K
\]

Where \( \Theta_l \), are the coefficients matrix of the MA representation of the VAR model, \( \Sigma \varepsilon \) is the variance matrix of the reduced-form error vector \( \varepsilon \), \( \sigma_{ii} \) is the standard deviation of the error term for the \( ith \) equation and \( e_i \) and \( e_j \) are selection vectors with ones as the \( ith \) element and zeros elsewhere.

If `normalized=TRUE` (the default value) then \( g.fevd \) computes:

\[
\tilde{\alpha}_{ij}^g(h) = \frac{\alpha_{ij}^g(h)}{\sum_{j=1}^{K} \alpha_{ij}^g(h)}
\]

This fact implies the normalization is simply each entry of the generalized `fevd` divided by its corresponding row sum.

**Value**

A list of length \( K \) holding the generalized forecast error variances as matrices. This is an object of class `varfevd` from vars package.
G.spillover

**Author(s)**

Jilber Urbina

**References**


**Examples**

```r
library(vars)
data(stock.prices)
stocks <- stock.prices[,1:2]
VAR.1 <- VAR(stocks)
g.fevd(VAR.1, n.ahead = 10) # normalized
g.fevd(VAR.1, n.ahead = 10, normalized=FALSE) # Not normalized
```

**G.spillover**

*Generalized spillover index*

**Description**

Computes the generalized spillover index proposed in Diebold and Yilmaz (2012) which is based on the General Forecast Variance Decomposition introduced by Pesaran and Shin (1998).

**Usage**

```r
G.spillover(x, n.ahead = 10, standardized = TRUE)
```

**Arguments**

- `x`: Object of class `varest` generated by `VAR()` from `vars` package.
- `n.ahead`: Integer specifying the steps ahead.
- `standardized`: A logical value indicating whether the values should be divided by the number of columns to get a percentage.

**Details**

This function computes the Generalized Directional Spillover Table which has as its \(ij^{th}\) entry the estimated contribution to the forecast error variance of variable \(i\) coming from innovations to variable \(j\). The off-diagonal column sums are the *Contributions to Others*, while the row sums represent *Contributions from Others*, when these are totaled across countries then we have the numerator of the Spillover Index. Similarly, the columns sums or rows sums (including diagonal), when totaled across countries, give the denominator of the Spillover Index, which is 100%.

`G.spillover` is based upon the General Forecast Error Variance Decomposition introduced by Pesaran and Shin (1998) and its explicit formulation can be found in Diebold and Yilmaz (2010).
Value

A data.frame consisting of the spillover index.

Author(s)

Jilber Urbina

References


See Also

O.spillover

Examples

# Replicating Diebold and Yilmaz (2012)
data(dy2012)
VAR_4 <- VAR(dy2012[,,-1], p=4)
G.spillover(VAR_4, standardized = FALSE)

net(x)

Arguments

x Object of class ‘spillover.table’ generated by either O.spillover() or G.spillover().

Value

A list length $K$ holding the generalized forecast error variances as matrices.

Author(s)

Jilber Urbina
References


See Also

O.spillover G.spillover

---

**O.spillover**  
*Orthogonalized spillover index*

**Description**

Computes the orthogonalized spillover index proposed in Diebold and Yilmaz (2009) which is based on the Orthogonalized Forecast Error Variance Decomposition.

**Usage**

```r
O.spillover(
  x,
  n.ahead = 10,
  ortho.type = c("single", "partial", "total"),
  standardized = TRUE
)
```

**Arguments**

- **x**  
  Object of class ‘varest’ generated by `VAR()` from vars package.

- **n.ahead**  
  Integer specifying the steps ahead.

- **ortho.type**  
  A character string indicating the type of orthogonalized index is required. "single" takes the original ordering of variables in VAR model and applies Cholesky decomposition for the fevd. Whereas "partial" takes a random sample out of all the possible combinations generated for the Cholesky decomposition, while "total" uses all the combinations, therefore it takes more time to finish. Both, "partial" and "total" provide average results.

- **standardized**  
  A logical value indicating whether the values should be divided by the number of columns to get a percentage.

**Details**

This function computes the Orthogonalized Directional Spillover Table which has as its $ij^{th}$ entry the estimated contribution to the forecast error variance of variable $i$ coming from innovations to variable $j$. The off-diagonal column sums are the *Contributions to Others*, while the row sums represent *Contributions from Others*, when these are totaled across countries then we have the numerator of the Spillover Index. Similarly, the columns sums or rows sums (including diagonal), when totaled across countries, give the denominator of the Spillover Index, which is 100%.
0. spillover is based upon the Orthogonalized (using Cholesky orthogonalization) Forecast Error Variance Decomposition (see Lutkepohl, 2006) and its explicit formulation can be found in Diebold and Yilmaz (2009).

Since 0. spillover is based on orthogonalized FEVD, then the result is as many indeces as combinations is allowed according to the number of variables in the VAR model, this is exactly equal to $K!$, then output has three options: table, summary and all.ind. table produces a data.frame holding the (orthogonalized) directional mean spillover indices.

When output="table", a data.frame is generated consisting of either mean or median directional spillover indeces, this because for each possible order of the variables the o.fevd is computed and over this result a spillover index is generated and this procedure repeats until reaching the last order (this means all the possible combinations given by $K!$). When output="table" a mean directional spillover table is generated, but this can be changed using stat="median" for a median directional spillover to be genereated. Note that stat argument only affects the results of output="table".

When output="summary" an vector is generated, this contains Mean,Min,Max.

This is a user-friendly version of fastSOM::sot_avg_exact() function.

**Value**

When output="table", a data.frame consisting of the spillover index.

When output="summary", a summary of all spillover indeces.

**Author(s)**

Jilber Urbina

**References**


**See Also**

G.spillover

**Examples**

```r
library(vars)
data(stock.prices)
stocks <- stock.prices[,1:2]
VAR.1 <- VAR(stocks)
O.spillover(VAR.1, n.ahead=5)
O.spillover(VAR.1, n.ahead=5, ortho.type = "partial")
O.spillover(VAR.1, n.ahead=5, ortho.type = "total")

# Replicating Table 3, Diebold and Yilmaz (2009)

data(dy2009)
VAR.2 <- VAR(dy2009[,-1], p=2)
```
rol.vol

Two-days Rolling Average Intra-day Volatilities

Description

A dataset of class zoo consisting of 1633 two-days rolling average observations on intraday volatilities based on Garman and Klass (1980) for six leading stock indices: S&P 500 (US), FTSE 100 (UK), EURO STOXX 50 (Eurozone), BOVESPA (Brazil), NIKKEI 225 (Japan) and S&P ASX 200 (Australia). EURO STOXX 50 covers 50 stocks from 12 Eurozone countries: Austria, Belgium, Finland, France, Germany, Greece, Ireland, Italy, Luxembourg, the Netherlands, Portugal and Spain. The period for this dataset is from June 13, 2003 to September 15, 2009. All series are in US Dollars.

Format

a zoo-class dataset

Examples

data(rol.vol)
head(rol.vol)  # First 6 observations
tail(rol.vol)  # Last 6 observations
Description

Estimates the dynamic spillover index given a moving window as described in Diebold and Yilmaz (2012).

Usage

```r
roll.net(
  data, 
  width, 
  n.ahead = 10, 
  index = c("orthogonalized", "generalized"), 
  ortho.type = c("partial", "total"), 
  ... 
)
```

Arguments

- **data**: Object of class ‘zoo’.
- **width**: An integer specifying the window width which is aligned to the original sample.
- **n.ahead**: An integer indicating the how many steps ahead the spillover should be forecasted.
- **index**: A character string indicating whether the orthogonalized or the generalized index is computed.
- **ortho.type**: A character string indicating the type of orthogonalized index is required. "partial" takes a random sample out of all the possible combinations generated for the Choleski decomposition, while "total" uses all the combinations, therefore it takes more time to finish.
- **...**: Further arguments to be passed to VAR function from vars package.

Value

A zoo object holding all the net spillover index estimations.

Author(s)

Jilber Urbina

References

Examples

```r
data(dy2012)
G_net <- roll.net(as.zoo(dy2012[1:300, c(2,3,4)]), width = 200, index="generalized")
```

data(dy2012)
```
# orthogonalized rolling net spillover index, based on a VAR(2)
O_net_dy2012 <- roll.net(as.zoo(dy2012[, -1]), width = 200)
# Generalized rolling net spillover index, based on a VAR(2)
G_net_dy2012 <- roll.net(as.zoo(dy2012[, -1]), width = 200, index="generalized")
```

roll.spillover **Dynamic Spillover Index**

Description

Estimates the dynamic spillover index given a rolling window as described in Diebold and Yilmaz (2012).

Usage

```r
roll.spillover(
  data,
  width,
  n.ahead = 10,
  index = c("orthogonalized", "generalized"),
  ortho.type = c("single", "partial", "total"),
  ...
)
```

Arguments

- **data**: Object of class ‘zoo’.
- **width**: An integer specifying the window width which is aligned to the original sample.
- **n.ahead**: An integer indicating the how many steps ahead the spillover should be forecasted.
- **index**: A character string indicating whether the orthogonalized or the generalized index is computed.
- **ortho.type**: Applicable only if `index="orthogonalized"`. A character string indicating the type of orthogonalized index is required. "single" takes the original ordering of variables in VAR model and applies Cholesky decomposition for the fevd. Whereas "partial" takes a random sample out of all the possible combinations generated for the Cholesky decomposition, while "total" uses all the combinations, therefore it takes more time to finish. Both, "partial" and "total" provide average results.
- **...**: Further arguments to be passed to `VAR` function from `vars` package.
Value

A zoo object holding all the indexes.

Author(s)

Jilber Urbina

References


Examples

data(dy2012)
O_index <- roll.spillover(as.zoo(dy2012[1:300,c(2,3,4)]), width = 200, p=4)

# Orthogonalized rolling spillover index based on a VAR(4), single order
O_index <- roll.spillover(as.zoo(dy2012[,1:3]), width = 200, p=4)

# Generalized rolling spillover index based on a VAR(4)
G_index<- roll.spillover(as.zoo(dy2012[,1:3]), width = 200, index="generalized", p=4)

# A comparison: (warning: It can take several minutes.)
single <- roll.spillover(as.zoo(dy2012[1:1200,2:4]), width = 200, p=4)
partial <- roll.spillover(as.zoo(dy2012[1:1200,2:4]), width = 200, p=4, ortho.type = "partial")
total <- roll.spillover(as.zoo(dy2012[1:1200,2:4]), width = 200, p=4, ortho.type = "total")
out <- cbind(single, partial, total)
head(out)
plot(out, col=1:3, main="Spillover index")
References


stock.prices

Daily Stock Prices

Description

A dataset consisting of 3507 daily observations on closed price for six leading stock indices: S&P 500 (US), FTSE 100 (UK), EURO STOXX 50 (Eurozone), BOVESPA (Brazil), NIKKEI 225 (Japan) and S&P ASX 200 (Australia). EURO STOXX 50 covers 50 stocks from 12 Eurozone countries: Austria, Belgium, Finland, France, Germany, Greece, Ireland, Italy, Luxembourg, the Netherlands, Portugal and Spain. The period for this dataset is from December 31, 1999 to June 10, 2013. All series are in US Dollars.

Format

a zoo-class dataset

Examples

data(stock.prices)
head(stock.prices)  # First 6 observations
tail(stock.prices)  # Last 6 observations
Index

* Generalized
  g.fevd, 3
* Orthogonal
  O.spillover, 6
* dataset
  dy2009, 2
dy2012, 2
  ro1.returns, 8
  ro1.vol, 8
  stock.prices, 12
* decomposition
  g.fevd, 3
  O.spillover, 6
* error
  g.fevd, 3
  O.spillover, 6
* fevd
  g.fevd, 3
* forecast
  O.spillover, 6
* g.fevd
  g.fevd, 3
* htest
  roll.net, 9
  roll.spillover, 10
* index
  O.spillover, 6
* models
  G.spillover, 4
* o.fevd
  O.spillover, 6
* package
  Spillover, 11
* regression
  G.spillover, 4
* spillover
  O.spillover, 6
* ts
  roll.net, 9