Package 'Spillover'

February 12, 2023

Type Package

Title Spillover/Connectedness Index Based on VAR Modelling

Version 0.1.0.3

Date 2023-02-7

Author Jilber Urbina

Maintainer Jilber Urbina <jurbina@cipadla.com>

Description A user-friendly tool for estimating both total and directional connectedness spillovers based on Diebold and Yilmaz (2009, 2012). It also provides the user with rolling estimation for total and net indices. User can find both orthogonalized and generalized versions for each kind of measures. See Diebold and Yilmaz (2009, 2012) find them at <doi:10.1111/j.1468-0297.2008.02208.x> and <doi:10.1016/j.ijforecast.2011.02.006>.

License GPL-2

Depends R (>= 3.5.0), vars

Imports dplyr, ggplot2, fastSOM, tidyr, zoo

Encoding UTF-8

RoxygenNote 7.2.3

Suggests rmarkdown, knitr

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2023-02-12 21:52:09 UTC

R topics documented:

`dy2009` *Diebold and Yilmaz (2009) dataset*

Description

A dataset consisting of 829 weekly observations of Global Stock Market Returns, 10/1/1992 – 23/11/2007. Find details in Diebold and Yilmaz (2009)

Format

a `data.frame`-class dataset

References

Diebold, F. X. & Yilmaz, K. (2009). *Measuring Financial Asset Return and Volatility Spillovers, with Application to Global Equity Markets*. The Economic Journal, 119, 158–171

Examples

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

`dy2012` *Diebold and Yilmaz (2012) dataset*

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

Diebold, F. X. & Yilmaz, K.(2012). *Better to Give than to Receive: Predictive Directional Measurement of Volatility Spillovers*. International Journal of Forecasting. 28, 57–66.

Examples

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

`dynamic.spillover` *dynamnic_spillover is a function for estimating DY directional spillovers.*

Description

dynamnic_spillover is a function for estimating DY directional spillovers.

Usage

```
dynamic.spillover(
  data,
  width,
  n.ahead = 10,
  standardized = TRUE,
  na.fill = FALSE,
  remove.own = TRUE,
  ...
)
```

Arguments

`data`	a data.frame consisting of dates in its first column and numeric variables for the others
`width`	a integer specifying the window width in number of observations.
`n.ahead`	An integer indicating the how many steps ahead the spillover should be forecasted.
`standardized`	A logical value indicating whether the values should be divided by the number of columns to get a percentage.
`na.fill`	A logical value for filling with NA at the begining window due to `width`.
`remove.own`	should own directional spillover be removed?
`...`	Further arguments to be passed to var function

Value

A list of data.frames holding all directional spillovers as described in Diebold and Yilmaz (2012)

Examples

```
data(dy2012)

dy_results <- dynamic.spillover(data=dy2012, width=200, remove.own = FALSE)
```

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

Description

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

Usage

```
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:

$$\alpha_{ij}^{g}(h) = \frac{\sigma_{ii}^{-1} \sum_{l=0}^{h-1} (\mathbf{e'}_i \Theta_l \Sigma_\varepsilon \mathbf{e}_j)^2}{\sum_{l=0}^{h-1} (\mathbf{e'}_i \Theta_l \Sigma_\varepsilon \Theta'_l \mathbf{e}_i)}, \quad i,j = 0,1,2\ldots,K$$

Where $\mathbf{\Theta}_l$, are the coefficients matrix of the MA representation of the VAR model, $\mathbf{\Sigma}_\varepsilon$ is the variance matrix of the reduced-form error vector ε, σ_{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:

$$\tilde{a}_{ij}^{g}(h) = \frac{a_{ij}^{g}(h)}{\sum_{j=1}^{K} a_{ij}^{g}(h)}$$

This fact implies the normalization is simply each entry of the generalized fevd divided by the 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.

Author(s)

Jilber Urbina

References

Pesaran, M. H. and Shin, Y. (1998). *Generalized impulse response analysis in linear multivariate models*. Economics Letters, 58(1):17-29.

Examples

```
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 Decompositon introduced by Pesaran and Shin (1998).

Usage

```
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 Decompositon 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

Diebold, F. X. & Yilmaz, K.(2012). *Better to Give than to Receive: Predictive Directional Measurement of Volatility Spillovers*. International Journal of Forecasting.

Pesaran, M. H. and Shin, Y. (1998). *Generalized impulse response analysis in linear multivariate models*. Economics Letters, 58(1):17-29.

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 *Net spillovers*

Description

Computes the net spillover index.

Usage

```
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

Pesaran, M. H. and Shin, Y. (1998). *Generalized impulse response analysis in linear multivariate models*. Economics Letters, 58(1):17-29.

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 Decompositon.

Usage

```
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%.

`O.spillover` is based upon the Orthogonalized (using Cholesky orthogonalization) Forecast Error Variance Decompositon (see Lutkepohl, 2006) and its explicit formulation can be found in Diebold and Yilmaz (2009).

Since `O.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-frendly 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

Diebold, F. X. & Yilmaz, K. (2009). *Measuring Financial Asset Return and Volatility Spillovers, with Application to Global Equity Markets*. The Economic Journal, 119, 158-171

Lutkepohl, H. (2006), *New Introduction to Multiple Time Series Analysis*, Springer, New York.

See Also

`G.spillover`

Examples

```
library(vars)

# Replicating Table 3, Diebold and Yilmaz (2009)

data(dy2009)
VAR.2 <- VAR(dy2009[,-1], p=2)
O.spillover(VAR.2, ortho.type  = "single", standardized = FALSE)
O.spillover(VAR.2, ortho.type  = "partial" )
```

plotdy *Plot directional spillover index*

Description

Plot directional spillover index

Usage

```
plotdy(
  data,
  direction = c("from", "to", "net", "net_pairwise", "from_to_pairwise")
)
```

Arguments

data	An object class directional.spillover resulting from applying dynamic.spillover
direction	A character string indicating the direction of the spillover, see Diebold and Yilmaz (2012)

Value

a ggplot2 object consisting of directional plots

References

Diebold, F. X. & Yilmaz, K.(2012). *Better to Give than to Receive: Predictive Directional Measurement of Volatility Spillovers*. International Journal of Forecasting.

Examples

```
library(Spillover)
require(tidyr)
require(dplyr)
require(ggplot2)

data(dy2012)
dy_results <- dynamic.spillover(data=dy2012, width=200, remove.own = FALSE)

pp_from <- plotdy(dy_results, direction = "from")

pp_from +
 labs(title="Any title") +
 facet_wrap(~variables, ncol = 2)

pp_to <- plotdy(dy_results, direction = "to")

pp_net <- plotdy(dy_results, direction = "net")
```

```
pp_netpairwise <- plotdy(dy_results, direction = "net_pairwise")

pp_from_to_pairwise <- plotdy(dy_results, direction = "from_to_pairwise")
```

`rol.returns` *Two-days Rolling Average Returns*

Description

A dataset of class `zoo` consisting of 1632 two-days rolling average observations on returns based 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 June 16, 2003 to September 15, 2009. All series are in US Dollars.

Format

a `zoo`-class dataset

Examples

```
data(rol.returns)
head(rol.returns)  # First 6 observations
tail(rol.returns)  # Last 6 observations
```

`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
```

---

roll.net *Dynamic Spillover Index*

Description

Estimates the dynamic spillover index given a moving window as described in Diebold and Yilmaz (2012). We recommend switching to `dynamic.spillover`.

Usage

```
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

Diebold, F. X. & Yilmaz, K.(2012). *Better to Give than to Receive: Predictive Directional Measurement of Volatility Spillovers*. International Journal of Forecasting.

Examples

```
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). We recommend switching to `dynamic.spillover`.

Usage

```
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 indeces.

Author(s)

Jilber Urbina

References

Diebold, F. X. & Yilmaz, K.(2012). *Better to Give than to Receive: Predictive Directional Measurement of Volatility Spillovers*. International Journal of Forecasting.

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

Spillover *Spillover*

Description

Spillover index based on both, orthogonalized and generalized forecast error variance decomposition of a VAR(p) for n step ahead forecast. It computes both orthogonalized and generalized directional spillover indices proposed by Diebold and Yilmaz (2009, 2012)

Author(s)

Jilber Urbina

References

Diebold, F. X. & Yilmaz, K. (2009). *Measuring Financial Asset Return and Volatility Spillovers, with Application to Global Equity Markets*. The Economic Journal, 119, 158-171.

Diebold, F. X. & Yilmaz, K.(2012). *Better to Give than to Receive: Predictive Directional Measurement of Volatility Spillovers*. International Journal of Forecasting. 28, 57–66.

Hamilton, J. (1994), *Time Series Analysis*, Princeton University Press, Princeton.

Lutkepohl, H. (2006), *New Introduction to Multiple Time Series Analysis*, Springer, New York.

Pesaran, M. H. and Shin, Y. (1998). *Generalized impulse response analysis in linear multivariate models*. Economics Letters, 58(1):17-29.

---

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
```

`total.dynamic.spillover` *Dynamic Spillover Index*

Description

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

Usage

```
total.dynamic.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 total dynamic index.

Author(s)

Jilber Urbina

References

Diebold, F. X. & Yilmaz, K.(2012). *Better to Give than to Receive: Predictive Directional Measurement of Volatility Spillovers*. International Journal of Forecasting.

Examples

```
data(dy2012)
O_index <- total.dynamic.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 <- total.dynamic.spillover(as.zoo(dy2012[,-1]), width = 200, p=4)
# Generalized rolling spillover index based on a VAR(4)
 G_index<- total.dynamic.spillover(as.zoo(dy2012[,-1]), width = 200, index="generalized", p=4)
# A comparison: (warning: It can take several minutes.)
 single <- total.dynamic.spillover(as.zoo(dy2012[1:1200,2:4]), width = 200, p=4)
 partial <- total.dynamic.spillover(as.zoo(dy2012[1:1200,2:4]), width = 200, p=4,
 ortho.type = "partial")
 total <- total.dynamic.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")
```

Index