Package ‘svars’

May 9, 2026

Type Package

Title Data-Driven Identification of SVAR Models

Version 1.3.12

Date 2025-08-24

Author Alexander Lange [aut, cre],
Bernhard Dalheimer [aut],
Helmut Herwartz [aut],
Simone Maxand [aut],
Hannes Riebl [ctb]

Maintainer Alexander Lange <alexander.lange@uni-goettingen.de>

Description Implements data-driven identification methods for structural vector autoregressive (SVAR) models as described in Lange et al. (2021) <doi:10.18637/jss.v097.i05>.
Based on an existing VAR model object (provided by e.g. VAR() from the 'vars' package), the structural
impact matrix is obtained via data-driven identification techniques (i.e. changes in volatility (Rigobon, R. (2003) <doi:10.1162/003465303772815727>), patterns of GARCH (Normadin, M., Phaneuf, L. (2004) <doi:10.1016/j.jmoneco.2003.11.002>),
independent component analysis (Matteson, D. S, Tsay, R. S., (2013) <doi:10.1080/01621459.2016.1150851>), least dependent innovations (Herwartz, H., Ploedt, M., (2016) <doi:10.1016/j.jimonfin.2015.11.001>),
smooth transition in variances (Luetkepohl, H., Netsunajev, A. (2017) <doi:10.1016/j.jedc.2017.09.001>) or non-Gaussian maximum likelihood (Lanne, M., Meitz, M., Saikkonen, P. (2017) <doi:10.1016/j.jeconom.2016.06.002>)).

Depends R (>= 2.10), vars (>= 1.5.3)

Encoding UTF-8

Imports expm, reshape2, ggplot2, copula, clue, pbapply, steadyICA,
DEoptim, zoo, strucchange, Rcpp, methods

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

License MIT + file LICENSE

LazyData TRUE

RoxygenNote 7.3.2

SystemRequirements C++17

Suggests testthat (>= 2.1.0), tsDyn

Repository CRAN

Date/Publication 2025-08-24 17:30:02 UTC

Contents

`ba.boot` *Bootstrap after Bootstrap*

Description

Bootstrap intervals based on bias-adjusted estimators

Usage

```
ba.boot(x, nc = 1)
```

Arguments

`x`	SVAR object of class "sboot"
`nc`	Integer. Number of processor cores

Value

A list of class "sboot" with elements

`true`	Point estimate of impulse response functions
`bootstrap`	List of length "nboot" holding bootstrap impulse response functions
`SE`	Bootstrapped standard errors of estimated covariance decomposition (only if "x" has method "Cramer von-Mises", or "Distance covariances")
`nboot`	Number of bootstrap iterations
`b_length`	Length of each block
`point_estimate`	Point estimate of covariance decomposition
`boot_mean`	Mean of bootstrapped covariance decompositions
`signrest`	Evaluated sign pattern
`sign_complete`	Frequency of appearance of the complete sign pattern in all bootstrapped covariance decompositions
`sign_part`	Frequency of bootstrapped covariance decompositions which conform the complete predetermined sign pattern. If signrest=NULL, the frequency of bootstrapped covariance decompositions that hold the same sign pattern as the point estimate is provided.
`sign_part`	Frequency of single shocks in all bootstrapped covariance decompositions which accord to a specific predetermined sign pattern
`cov_bs`	Covariance matrix of bootstrapped parameter in impact relations matrix
`method`	Used bootstrap method
`VAR`	Estimated input VAR object

References

Kilian, L. (1998). Small-sample confidence intervals for impulse response functions. Review of Economics and Statistics 80, 218-230.

See Also

`mb.boot`, `wild.boot`

Examples

```
# data contains quarterly observations from 1965Q1 to 2008Q3
# x = output gap
# pi = inflation
# i = interest rates
set.seed(23211)
v1 <- vars::VAR(USA, lag.max = 10, ic = "AIC" )
x1 <- id.dc(v1)
summary(x1)

# Bootstrap
bb <- mb.boot(x1, b.length = 15, nboot = 300, n.ahead = 30, nc = 1, signrest = NULL)
```

```
summary(bb)
plot(bb, lowerq = 0.16, upperq = 0.84)

# Bias-adjusted bootstrap
bb2 <- ba.boot(bb, nc = 1)
plot(bb2, lowerq = 0.16, upperq = 0.84)
```

---

cf *Counterfactuals for SVAR Models*

Description

Calculation of Counterfactuals for an identified SVAR object 'svars' derived by function id.st( ), id.cvm( ),id.cv( ),id.dc( ) or id.ngml( ).

Usage

```
cf(x, series = 1, transition = 0)
```

Arguments

`x`	SVAR object of class "svars"
`series`	Integer. indicating the series for which the counterfactuals should be calculated.
`transition`	Numeric. Value from [0, 1] indicating how many initial values should be discarded, i.e., 0.1 means that the first 10 per cent observations of the sample are considered as transient.

Value

A list with class attribute "hd" holding the Counterfactuals as data frame.

References

Kilian, L., Luetkepohl, H., 2017. Structural Vector Autoregressive Analysis, Cambridge University Press.

See Also

`id.cvm`, `id.dc`, `id.ngml`, `id.cv`, `id.garch` or `id.st`

Examples

```
v1 <- vars::VAR(USA, lag.max = 10, ic = "AIC" )
x1 <- id.dc(v1)
x2 <- cf(x1, series = 2)
plot(x2)
```

chow.test *Chow Test for Structural Break*

Description

The Chow test for structural change is implemented as sample-split and break-point test (see Luetkepohl and Kraetzig, 2004, p. 135). An estimated VAR model and the presupposed structural break need to be provided.

Usage

```
chow.test(
  x,
  SB,
  nboot = 500,
  start = NULL,
  end = NULL,
  frequency = NULL,
  format = NULL,
  dateVector = NULL
)
```

Arguments

x	An object of class 'vars', 'vec2var', 'nlVar'. Estimated VAR object. Or an object of class 'chowpretest' from stability()
SB	Integer, vector or date character. The structural break is specified either by an integer (number of observations in the pre-break period), a vector of ts() frequencies if a ts object is used in the VAR or a date character. If a date character is provided, either a date vector containing the whole time line in the corresponding format or common time parameters need to be provided
nboot	Integer. Number of bootstrap iterations to calculate quantiles and p-values
start	Character. Start of the time series (only if dateVector is empty)
end	Character. End of the time series (only if dateVector is empty)
frequency	Character. Frequency of the time series (only if dateVector is empty)
format	Character. Date format (only if dateVector is empty)
dateVector	Vector. Vector of time periods containing SB in corresponding format

Value

A list of class "chow" with elements

lambda_bp	Test statistic of the Chow test with break point
testcrit_bp	Critical value of the test statistic lambda_bp
p.value_bp	p-value of the test statistic lambda_bp

lambda_sp	Test statistic of the Chow test with sample split
testcrit_sp	Critical value of the test statistic lambda_sp
p.value_sp	p-value of the test statistic lambda_sp
SB	Structural break tested
SBcharacter	Structural break tested as character
p	Number of lags used

References

Luetkepohl, H., 2005. New introduction to multiple time series analysis, Springer-Verlag, Berlin.
Luetkepohl, H., Kraetzig, M., 2004. Applied time series econometrics, Cambridge University Press, Cambridge.

See Also

stability

Examples

```
# Testing for structural break in USA data
#' # data contains quartlery observations from 1965Q1 to 2008Q2
# assumed structural break in 1979Q3
# x = output gap
# pi = inflation
# i = interest rates
set.seed(23211)
v1 <- vars::VAR(USA, lag.max = 10, ic = "AIC" )
z1 <- chow.test(v1, SB = 59)
summary(z1)

#Using stability() to find potential break point and sample split
x1 <- stability(v1, type = "mv-chow-test")
plot(x1)
z1.1 <- chow.test(x1)
summary(z1.1)
#Or using sample split as benchmark
x1$break_point <- FALSE
z1.1 <- chow.test(x1)
summary(z1.1)

#Structural brake via Dates
#given that time series vector with dates is available
dateVector <- seq(as.Date("1965/1/1"), as.Date("2008/7/1"), "quarter")
z2 <- chow.test(v1, SB = "1979-07-01", format = "%Y-%m-%d", dateVector = dateVector)
summary(z2)

# alternatively pass sequence arguments directly
z3 <- chow.test(v1, SB = "1979-07-01", format = "%Y-%m-%d",
                start = "1965-01-01", end = "2008-07-01",
                frequency = "quarter")
```

```
summary(z3)

# or provide ts date format (For quarterly, monthly, weekly and daily frequencies only)
z4 <- chow.test(v1, SB = c(1979,3))
summary(z4)
```

---

fevd *Forecast error variance decomposition for SVAR Models*

Description

Calculation of forecast error variance decomposition for an identified SVAR object 'svars' derived by function id.st( ), id.cvm( ),id.cv( ),id.dc( ) or id.ngml( ).

Usage

```
## S3 method for class 'svars'
fevd(x, n.ahead = 10, ...)
```

Arguments

x	SVAR object of class "svars".
n.ahead	Integer specifying the steps.
...	Currently not used.

Value

A list with class attribute "svarfevd" holding the forecast error variance decompositions as data frames.

References

Kilian, L., Luetkepohl, H., 2017. Structural Vector Autoregressive Analysis, Cambridge University Press.

See Also

id.cvm, id.garch, id.dc, id.ngml, id.cv or id.st

Examples

```
v1 <- vars::VAR(USA, lag.max = 10, ic = "AIC" )
x1 <- id.dc(v1)
x2 <- fevd(x1, n.ahead = 30)
plot(x2)
```

hd *Historical decomposition for SVAR Models*

Description

Calculation of historical decomposition for an identified SVAR object 'svars' derived by function id.st( ), id.cvm( ),id.cv( ),id.dc( ) or id.ngml( ).

Usage

```
hd(x, series = 1, transition = 0)
```

Arguments

x	SVAR object of class "svars"
series	Integer. indicating the series that should be decomposed.
transition	Numeric. Value from [0, 1] indicating how many initial values should be discarded, i.e., 0.1 means that the first 10 per cent observations of the sample are considered as transient.

Value

A list with class attribute "hd" holding the historical decomposition as data frame.

References

Kilian, L., Luetkepohl, H., 2017. Structural Vector Autoregressive Analysis, Cambridge University Press.

See Also

id.cvm, id.dc, id.ngml, id.cv, id.garch or id.st

Examples

```
v1 <- vars::VAR(USA, lag.max = 10, ic = "AIC" )
x1 <- id.dc(v1)
x2 <- hd(x1, series = 2)
plot(x2)
```

`id.chol` *Recursive identification of SVAR models via Cholesky decomposition*

Description

Given an estimated VAR model, this function uses the Cholesky decomposition to identify the structural impact matrix B of the corresponding SVAR model

$$y_t = c_t + A_1 y_{t-1} + ... + A_p y_{t-p} + u_t = c_t + A_1 y_{t-1} + ... + A_p y_{t-p} + B\epsilon_t.$$

Matrix B corresponds to the decomposition of the least squares covariance matrix $\Sigma_u = B\Lambda_t B'$.

Usage

```
id.chol(x, order_k = NULL)
```

Arguments

`x`	An object of class 'vars', 'vec2var', 'nlVar'. Estimated VAR object
`order_k`	Vector. Vector of characters or integers specifying the assumed structure of the recursive causality. Change the causal ordering in the instantaneous effects without permuting variables and re-estimating the VAR model.

Value

A list of class "svars" with elements

`B`	Estimated structural impact matrix B, i.e. unique decomposition of the covariance matrix of reduced form residuals
`n`	Number of observations
`method`	Method applied for identification
`order_k`	Ordering of the variables as assumed for recursive causality
`A_hat`	Estimated VAR parameter
`type`	Type of the VAR model, e.g. 'const'
`y`	Data matrix
`p`	Number of lags
`K`	Dimension of the VAR
`VAR`	Estimated input VAR object

References

Luetkepohl, H., 2005. New introduction to multiple time series analysis, Springer-Verlag, Berlin.

See Also

For alternative identification approaches see `id.st`, `id.cvm`, `id.cv`, `id.dc` or `id.ngml`

Examples

```
# data contains quarterly observations from 1965Q1 to 2008Q3
# x = output gap
# pi = inflation
# i = interest rates
set.seed(23211)
v1 <- vars::VAR(USA, lag.max = 10, ic = "AIC" )
x1 <- id.chol(v1)
x2 <- id.chol(v1, order_k = c("pi", "x", "i")) ## order_k = c(2,1,3)
summary(x1)


# impulse response analysis
i1 <- irf(x1, n.ahead = 30)
i2 <- irf(x2, n.ahead = 30)
plot(i1, scales = 'free_y')
plot(i2, scales = 'free_y')
```

`id.cv` *Identification of SVAR models based on Changes in volatility (CV)*

Description

Given an estimated VAR model, this function applies changes in volatility to identify the structural impact matrix B of the corresponding SVAR model

$$y_t = c_t + A_1 y_{t-1} + ... + A_p y_{t-p} + u_t = c_t + A_1 y_{t-1} + ... + A_p y_{t-p} + B\epsilon_t.$$

Matrix B corresponds to the decomposition of the pre-break covariance matrix $\Sigma_1 = BB'$. The post-break covariance corresponds to $\Sigma_2 = B\Lambda B'$ where Λ is the estimated unconditional heteroskedasticity matrix.

Usage

```
id.cv(
  x,
  SB,
  SB2 = NULL,
  start = NULL,
  end = NULL,
  frequency = NULL,
  format = NULL,
  dateVector = NULL,
  max.iter = 50,
```

```
  crit = 0.001,
  restriction_matrix = NULL
)
```

Arguments

- `x` — An object of class 'vars', 'vec2var', 'nlVar'. Estimated VAR object
- `SB` — Integer, vector or date character. The structural break is specified either by an integer (number of observations in the pre-break period), a vector of ts() frequencies if a ts object is used in the VAR or a date character. If a date character is provided, either a date vector containing the whole time line in the corresponding format (see examples) or common time parameters need to be provided
- `SB2` — Integer, vector or date character. Optional if the model should be estimated with two volatility regimes. The structural break is specified either by an integer (number of observations in the pre-break period), a vector of ts() frequencies if a ts object is used in the VAR or a date character. If a date character is provided, either a date vector containing the whole time line in the corresponding format (see examples) or common time parameters need to be provided
- `start` — Character. Start of the time series (only if dateVector is empty)
- `end` — Character. End of the time series (only if dateVector is empty)
- `frequency` — Character. Frequency of the time series (only if dateVector is empty)
- `format` — Character. Date format (only if dateVector is empty)
- `dateVector` — Vector. Vector of time periods containing SB in corresponding format
- `max.iter` — Integer. Number of maximum GLS iterations
- `crit` — Numeric. Critical value for the precision of the GLS estimation
- `restriction_matrix` — Matrix. A matrix containing presupposed entries for matrix B, NA if no restriction is imposed (entries to be estimated). Alternatively, a K^2*K^2 matrix can be passed, where ones on the diagonal designate unrestricted and zeros restricted coefficients. (as suggested in Luetkepohl, 2017, section 5.2.1).

Value

A list of class "svars" with elements

- `Lambda` — Estimated unconditional heteroscedasticity matrix Λ
- `Lambda_SE` — Matrix of standard errors of Lambda
- `B` — Estimated structural impact matrix B, i.e. unique decomposition of the covariance matrix of reduced form residuals
- `B_SE` — Standard errors of matrix B
- `n` — Number of observations
- `Fish` — Observed Fisher information matrix
- `Lik` — Function value of likelihood

`wald_statistic`	Results of sequential Wald-type identification test on equal eigenvalues as described in Luetkepohl et. al. (2021). In case of more than two regimes, pairwise Wald-type tests of equal diagonal elements in the Lambda matrices are performed.
`iteration`	Number of GLS estimations
`method`	Method applied for identification
`SB`	Structural break (number of observations)
`A_hat`	Estimated VAR parameter via GLS
`type`	Type of the VAR model, e.g. 'const'
`SBcharacter`	Structural break (date; if provided in function arguments)
`restrictions`	Number of specified restrictions
`restriction_matrix`	Specified restriction matrix
`y`	Data matrix
`p`	Number of lags
`K`	Dimension of the VAR
`VAR`	Estimated input VAR object

References

Rigobon, R., 2003. Identification through Heteroskedasticity. The Review of Economics and Statistics, 85, 777-792.

Herwartz, H. & Ploedt, M., 2016. Simulation Evidence on Theory-based and Statistical Identification under Volatility Breaks. Oxford Bulletin of Economics and Statistics, 78, 94-112.

Luetkepohl, H. & Meitz, M. & Netsunajev, A. & and Saikkonen, P., 2021. Testing identification via heteroskedasticity in structural vector autoregressive models. Econometrics Journal, 24, 1-22.

See Also

For alternative identification approaches see `id.st`, `id.garch`, `id.cvm`, `id.dc` or `id.ngml`

Examples

```
#' # data contains quartlery observations from 1965Q1 to 2008Q2
# assumed structural break in 1979Q3
# x = output gap
# pi = inflation
# i = interest rates
set.seed(23211)
v1 <- vars::VAR(USA, lag.max = 10, ic = "AIC" )
x1 <- id.cv(v1, SB = 59)
summary(x1)

# switching columns according to sign patter
```

```
x1$B <- x1$B[,c(3,2,1)]
x1$B[,3] <- x1$B[,3]*(-1)

# Impulse response analysis
i1 <- irf(x1, n.ahead = 30)
plot(i1, scales = 'free_y')

# Restrictions
# Assuming that the interest rate doesn't influence the output gap on impact
restMat <- matrix(rep(NA, 9), ncol = 3)
restMat[1,3] <- 0
x2 <- id.cv(v1, SB = 59, restriction_matrix = restMat)
summary(x2)

# In alternative Form
restMat <- diag(rep(1,9))
restMat[7,7]= 0
x2 <- id.cv(v1, SB = 59, restriction_matrix = restMat)
summary(x2)

#Structural brake via Dates
# given that time series vector with dates is available
dateVector = seq(as.Date("1965/1/1"), as.Date("2008/7/1"), "quarter")
x3 <- id.cv(v1, SB = "1979-07-01", format = "%Y-%m-%d", dateVector = dateVector)
summary(x3)

# or pass sequence arguments directly
x4 <- id.cv(v1, SB = "1979-07-01", format = "%Y-%m-%d", start = "1965-01-01", end = "2008-07-01",
frequency = "quarter")
summary(x4)

# or provide ts date format (For quarterly, monthly, weekly and daily frequencies only)
x5 <- id.cv(v1, SB = c(1979, 3))
summary(x5)

#-----# Example with three covariance regimes

x6 <- id.cv(v1, SB = 59, SB2 = 110)
summary(x6)
```

`id.cvm` *Independence-based identification of SVAR models via Cramer-von Mises (CVM) distance*

Description

Given an estimated VAR model, this function applies independence-based identification for the structural impact matrix B of the corresponding SVAR model

$$y_t = c_t + A_1 y_{t-1} + ... + A_p y_{t-p} + u_t = c_t + A_1 y_{t-1} + ... + A_p y_{t-p} + B\epsilon_t.$$

Matrix B corresponds to the unique decomposition of the least squares covariance matrix $\Sigma_u = BB'$ if the vector of structural shocks ϵ_t contains at most one Gaussian shock (Comon, 1994). A nonparametric dependence measure, the Cramer-von Mises distance (Genest and Remillard, 2004), determines least dependent structural shocks. The minimum is obtained by a two step optimization algorithm similar to the technique described in Herwartz and Ploedt (2016).

Usage

```
id.cvm(x, dd = NULL, itermax = 500, steptol = 100, iter2 = 75)
```

Arguments

`x`	An object of class 'vars', 'vec2var', 'nlVar'. Estimated VAR object
`dd`	Object of class 'indepTestDist' (generated by 'indepTest' from package 'copula'). A simulated independent sample of the same size as the data. If not supplied, it will be calculated by the function
`itermax`	Integer. IMaximum number of iterations for DEoptim
`steptol`	Numeric. Tolerance for steps without improvement for DEoptim
`iter2`	Integer. Number of iterations for the second optimization

Value

A list of class "svars" with elements

`B`	Estimated structural impact matrix B, i.e. unique decomposition of the covariance matrix of reduced form errors
`A_hat`	Estimated VAR parameter
`method`	Method applied for identification
`n`	Number of observations
`type`	Type of the VAR model, e.g. 'const'
`y`	Data matrix
`p`	Number of lags
`K`	Dimension of the VAR
`rotation_angles`	Rotation angles, which lead to maximum independence
`inc`	Indicator. 1 = second optimization increased the estimation precision. 0 = second optimization did not increase the estimation precision
`test.stats`	Computed test statistics of independence test
`iter1`	Number of iterations of first optimization
`test1`	Minimum test statistic from first optimization
`test2`	Minimum test statistic from second optimization
`VAR`	Estimated input VAR object

References

Herwartz, H., 2018. Hodges Lehmann detection of structural shocks - An Analysis of macroeconomic dynamics in the Euro Area, Oxford Bulletin of Economics and Statistics

Herwartz, H. & Ploedt, M., 2016. The macroeconomic effects of oil price shocks: Evidence from a statistical identification approach, Journal of International Money and Finance, 61, 30-44

Comon, P., 1994. Independent component analysis, A new concept?, Signal Processing, 36, 287-314

Genest, C. & Remillard, B., 2004. Tests of independence and randomness based on the empirical copula process, Test, 13, 335-370

See Also

For alternative identification approaches see `id.st`, `id.garch`, `id.cv`, `id.dc` or `id.ngml`

Examples

```
# data contains quarterly observations from 1965Q1 to 2008Q3
# x = output gap
# pi = inflation
# i = interest rates
set.seed(23211)
v1 <- vars::VAR(USA, lag.max = 10, ic = "AIC" )
cob <- copula::indepTestSim(v1$obs, v1$K, verbose=FALSE)
x1 <- id.cvm(v1, dd = cob)
summary(x1)

# switching columns according to sign pattern
x1$B <- x1$B[,c(3,2,1)]
x1$B[,3] <- x1$B[,3]*(-1)

# impulse response analysis
i1 <- irf(x1, n.ahead = 30)
plot(i1, scales = 'free_y')
```

`id.dc` *Independence-based identification of SVAR models build on distance covariances (DC) statistic*

Description

Given an estimated VAR model, this function applies independence-based identification for the structural impact matrix B of the corresponding SVAR model

$$y_t = c_t + A_1 y_{t-1} + ... + A_p y_{t-p} + u_t = c_t + A_1 y_{t-1} + ... + A_p y_{t-p} + B\epsilon_t.$$

Matrix B corresponds to the unique decomposition of the least squares covariance matrix $\Sigma_u = BB'$ if the vector of structural shocks ϵ_t contains at most one Gaussian shock (Comon, 1994). A nonparametric dependence measure, the distance covariance (Szekely et al, 2007), determines least dependent structural shocks. The algorithm described in Matteson and Tsay (2013) is applied to calculate the matrix B.

Usage

```
id.dc(x, PIT = FALSE)
```

Arguments

`x`	An object of class 'vars', 'vec2var', 'nlVar'. Estimated VAR object
`PIT`	Logical. If PIT='TRUE', the distribution and density of the independent components are estimated using gaussian kernel density estimates

Value

A list of class "svars" with elements

`B`	Estimated structural impact matrix B, i.e. unique decomposition of the covariance matrix of reduced form errors
`A_hat`	Estimated VAR parameter
`method`	Method applied for identification
`n`	Number of observations
`type`	Type of the VAR model, e.g. 'const'
`y`	Data matrix
`p`	Number of lags
`K`	Dimension of the VAR
`PIT`	Logical, if PIT is used
`VAR`	Estimated input VAR object

References

Matteson, D. S. & Tsay, R. S., 2013. Independent Component Analysis via Distance Covariance, pre-print
Szekely, G. J.; Rizzo, M. L. & Bakirov, N. K., 2007. Measuring and testing dependence by correlation of distances Ann. Statist., 35, 2769-2794
Comon, P., 1994. Independent component analysis, A new concept?, Signal Processing, 36, 287-314

See Also

For alternative identification approaches see `id.st`, `id.garch`, `id.cvm`, `id.cv` or `id.ngml`

Examples

```
# data contains quarterly observations from 1965Q1 to 2008Q3
# x = output gap
# pi = inflation
# i = interest rates
set.seed(23211)
v1 <- vars::VAR(USA, lag.max = 10, ic = "AIC" )
x1 <- id.dc(v1)
summary(x1)

# switching columns according to sign pattern
x1$B <- x1$B[,c(3,2,1)]
x1$B[,3] <- x1$B[,3]*(-1)

# impulse response analysis
i1 <- irf(x1, n.ahead = 30)
plot(i1, scales = 'free_y')
```

---

id.garch — *Identification of SVAR models through patterns of GARCH*

Description

Given an estimated VAR model, this function uses GARCH-type variances to identify the structural impact matrix B of the corresponding SVAR model

$$y_t = c_t + A_1 y_{t-1} + ... + A_p y_{t-p} + u_t = c_t + A_1 y_{t-1} + ... + A_p y_{t-p} + B\epsilon_t.$$

Matrix B corresponds to the decomposition of the least squares covariance matrix $\Sigma_u = B\Lambda_t B'$, where Λ_t is the estimated conditional heteroskedasticity matrix.

Usage

```
id.garch(
  x,
  max.iter = 5,
  crit = 0.001,
  restriction_matrix = NULL,
  start_iter = 50
)
```

Arguments

`x`	An object of class 'vars', 'vec2var', 'nlVar'. Estimated VAR object
`max.iter`	Integer. Number of maximum likelihood optimizations
`crit`	Numeric. Critical value for the precision of the iterative procedure

`restriction_matrix`
: Matrix. A matrix containing presupposed entries for matrix B, NA if no restriction is imposed (entries to be estimated). Alternatively, a K^2*K^2 matrix can be passed, where ones on the diagonal designate unrestricted and zeros restricted coefficients. (as suggested in Luetkepohl, 2017, section 5.2.1).

`start_iter`
: Numeric. Number of random candidate initial values for univariate GRACH(1,1) optimization.

Value

A list of class "svars" with elements

`B`	Estimated structural impact matrix B, i.e. unique decomposition of the covariance matrix of reduced form residuals
`B_SE`	Standard errors of matrix B
`GARCH_parameter`	Estimated GARCH parameters of univariate GARCH models
`GARCH_SE`	Standard errors of GARCH parameters
`n`	Number of observations
`Fish`	Observed Fisher information matrix
`Lik`	Function value of likelihood
`iteration`	Number of likelihood optimizations
`method`	Method applied for identification
`A_hat`	Estimated VAR parameter via GLS
`type`	Type of the VAR model, e.g. 'const'
`restrictions`	Number of specified restrictions
`restriction_matrix`	Specified restriction matrix
`y`	Data matrix
`p`	Number of lags
`K`	Dimension of the VAR
`VAR`	Estimated input VAR object
`I_test`	Results of a series of sequential tests on the number of heteroskedastic shocks present in the system as described in Luetkepohl and Milunovich (2016).

References

Normadin, M. & Phaneuf, L., 2004. Monetary Policy Shocks: Testing Identification Conditions under Time-Varying Conditional Volatility. Journal of Monetary Economics, 51(6), 1217-1243.

Lanne, M. & Saikkonen, P., 2007. A Multivariate Generalized Orthogonal Factor GARCH Model. Journal of Business & Economic Statistics, 25(1), 61-75.

Luetkepohl, H. & Milunovich, G. 2016. Testing for identification in SVAR-GARCH models. Journal of Economic Dynamics and Control, 73(C):241-258

See Also

For alternative identification approaches see id.st, id.cvm, id.cv, id.dc or id.ngml

Examples

```
# data contains quartlery observations from 1965Q1 to 2008Q2
# assumed structural break in 1979Q3
# x = output gap
# pi = inflation
# i = interest rates
set.seed(23211)
v1 <- vars::VAR(USA, lag.max = 10, ic = "AIC" )
x1 <- id.garch(v1)
summary(x1)

# Impulse response analysis
i1 <- irf(x1, n.ahead = 30)
plot(i1, scales = 'free_y')

# Restrictions
# Assuming that the interest rate doesn't influence the output gap on impact
restMat <- matrix(rep(NA, 9), ncol = 3)
restMat[1,3] <- 0
x2 <- id.garch(v1, restriction_matrix = restMat)
summary(x2)
```

---

id.ngml *Non-Gaussian maximum likelihood (NGML) identification of SVAR models*

Description

Given an estimated VAR model, this function applies identification by means of a non-Gaussian likelihood for the structural impact matrix B of the corresponding SVAR model

$$y_t = c_t + A_1 y_{t-1} + ... + A_p y_{t-p} + u_t = c_t + A_1 y_{t-1} + ... + A_p y_{t-p} + B\epsilon_t.$$

Matrix B corresponds to the unique decomposition of the least squares covariance matrix $\Sigma_u = BB'$ if the vector of structural shocks ϵ_t contains at most one Gaussian shock (Comon, 94). A likelihood function of independent t-distributed structural shocks $\epsilon_t = B^{-1}u_t$ is maximized with respect to the entries of B and the degrees of freedom of the t-distribution (Lanne et al., 2017).

Usage

```
id.ngml(x, stage3 = FALSE, restriction_matrix = NULL)
```

Arguments

`x`	An object of class 'vars', 'vec2var', 'nlVar'. Estimated VAR object
`stage3`	Logical. If stage3="TRUE", the VAR parameters are estimated via non-gaussian maximum likelihood (computationally demanding)
`restriction_matrix`	Matrix. A matrix containing presupposed entries for matrix B, NA if no restriction is imposed (entries to be estimated). Alternatively, a K^2*K^2 matrix can be passed, where ones on the diagonal designate unrestricted and zeros restricted coefficients. (as suggested in Luetkepohl, 2017, section 5.2.1).

Value

A list of class "svars" with elements

`B`	Estimated structural impact matrix B, i.e. unique decomposition of the covariance matrix of reduced form errors
`sigma`	Estimated scale of the standardized matrix B_stand, i.e. $B = B_s tand * diag(\sigma_1, ..., \sigma_K)$
`sigma_SE`	Standard errors of the scale
`df`	Estimated degrees of freedom
`df_SE`	Standard errors of the degrees of freedom
`Fish`	Observed Fisher information matrix
`A_hat`	Estimated VAR parameter via ML
`B_stand`	Estimated standardized structural impact matrix
`B_stand_SE`	Standard errors of standardized matrix B_stand
`Lik`	Function value of likelihood
`method`	Method applied for identification
`n`	Number of observations
`type`	Type of the VAR model, e.g. 'const'
`y`	Data matrix
`p`	Number of lags
`K`	Dimension of the VAR
`restrictions`	Number of specified restrictions
`restriction_matrix`	Specified restriction matrix
`stage3`	Logical, whether Stage 3 is performed
`VAR`	Estimated input VAR object

References

Lanne, M., Meitz, M., Saikkonen, P., 2017. Identification and estimation of non-Gaussian structural vector autoregressions. J. Econometrics 196 (2), 288-304.
Comon, P., 1994. Independent component analysis, A new concept?, Signal Processing, 36, 287-314

See Also

For alternative identification approaches see `id.st`, `id.garch`, `id.cvm`, `id.dc` or `id.cv`

Examples

```
# data contains quarterly observations from 1965Q1 to 2008Q3
# x = output gap
# pi = inflation
# i = interest rates
set.seed(23211)
v1 <- vars::VAR(USA, lag.max = 10, ic = "AIC" )
x1 <- id.ngml(v1)
summary(x1)

# switching columns according to sign pattern
x1$B <- x1$B[,c(3,2,1)]
x1$B[,3] <- x1$B[,3]*(-1)

# impulse response analysis
i1 <- irf(x1, n.ahead = 30)
plot(i1, scales = 'free_y')
```

`id.st` *Identification of SVAR models by means of a smooth transition (ST) in covariance*

Description

Given an estimated VAR model, this function uses a smooth transition in the covariance to identify the structural impact matrix B of the corresponding SVAR model

$$y_t = c_t + A_1 y_{t-1} + ... + A_p y_{t-p} + u_t = c_t + A_1 y_{t-1} + ... + A_p y_{t-p} + B\epsilon_t.$$

Matrix B corresponds to the decomposition of the pre-break covariance matrix $\Sigma_1 = BB'$. The post-break covariance corresponds to $\Sigma_2 = B\Lambda B'$ where Λ is the estimated heteroskedasticity matrix.

Usage

```
id.st(
  x,
  c_lower = 0.3,
  c_upper = 0.7,
  c_step = 5,
  c_fix = NULL,
  transition_variable = NULL,
  gamma_lower = -3,
  gamma_upper = 2,
```

```
  gamma_step = 0.5,
  gamma_fix = NULL,
  nc = 1,
  max.iter = 5,
  crit = 0.001,
  restriction_matrix = NULL,
  lr_test = FALSE
)
```

Arguments

`x` An object of class 'vars', 'vec2var', 'nlVar'. Estimated VAR object

`c_lower` Numeric. Starting point for the algorithm to start searching for the volatility shift. Default is 0.3*(Total number of observations)

`c_upper` Numeric. Ending point for the algorithm to stop searching for the volatility shift. Default is 0.7*(Total number of observations). Note that in case of a stochastic transition variable, the input requires an absolute value

`c_step` Integer. Step width of c. Default is 5. Note that in case of a stochastic transition variable, the input requires an absolute value

`c_fix` Numeric. If the transition point is known, it can be passed as an argument where transition point = Number of observations - c_fix

`transition_variable` A numeric vector that represents the transition variable. By default (NULL), the time is used as transition variable. Note that c_lower,c_upper, c_step and/or c_fix have to be adjusted to the specified transition variable

`gamma_lower` Numeric. Lower bound for gamma. Small values indicate a flat transition function. Default is -3

`gamma_upper` Numeric. Upper bound for gamma. Large values indicate a steep transition function. Default is 2

`gamma_step` Numeric. Step width of gamma. Default is 0.5

`gamma_fix` Numeric. A fixed value for gamma, alternative to gamma found by the function

`nc` Integer. Number of processor cores Note that the smooth transition model is computationally extremely demanding.

`max.iter` Integer. Number of maximum GLS iterations

`crit` Numeric. Critical value for the precision of the GLS estimation

`restriction_matrix` Matrix. A matrix containing presupposed entries for matrix B, NA if no restriction is imposed (entries to be estimated). Alternatively, a K^2*K^2 matrix can be passed, where ones on the diagonal designate unrestricted and zeros restricted coefficients. (as suggested in Luetkepohl, 2017, section 5.2.1).

`lr_test` Logical. Indicates whether the restricted model should be tested against the unrestricted model via a likelihood ratio test

Value

A list of class "svars" with elements

`Lambda`	Estimated heteroscedasticity matrix Λ
`Lambda_SE`	Matrix of standard errors of Lambda
`B`	Estimated structural impact matrix B, i.e. unique decomposition of the covariance matrix of reduced form residuals
`B_SE`	Standard errors of matrix B
`n`	Number of observations
`Fish`	Observed Fisher information matrix
`Lik`	Function value of likelihood
`wald_statistic`	Results of pairwise Wald tests
`iteration`	Number of GLS estimations
`method`	Method applied for identification
`est_c`	Structural break (number of observations)
`est_g`	Transition coefficient
`transition_variable`	Vector of transition variable
`comb`	Number of all grid combinations of gamma and c
`transition_function`	Vector of transition function
`A_hat`	Estimated VAR parameter via GLS
`type`	Type of the VAR model e.g., 'const'
`y`	Data matrix
`p`	Number of lags
`K`	Dimension of the VAR
`restrictions`	Number of specified restrictions
`restriction_matrix`	Specified restriction matrix
`lr_test`	Logical, whether a likelihood ratio test is performed
`lRatioTest`	Results of likelihood ratio test
`VAR`	Estimated input VAR object

References

Luetkepohl H., Netsunajev A., 2017. Structural vector autoregressions with smooth transition in variances. Journal of Economic Dynamics and Control, 84, 43 - 57. ISSN 0165-1889.

See Also

For alternative identification approaches see `id.cv`, `id.garch`, `id.cvm`, `id.dc`, or `id.ngml`

Examples

```
# data contains quartlery observations from 1965Q1 to 2008Q2
# x = output gap
# pi = inflation
# i = interest rates
set.seed(23211)
v1 <- vars::VAR(USA, lag.max = 10, ic = "AIC" )
x1 <- id.st(v1, c_fix = 80, gamma_fix = 0)
summary(x1)
plot(x1)

# switching columns according to sign patter
x1$B <- x1$B[,c(3,2,1)]
x1$B[,3] <- x1$B[,3]*(-1)

# Impulse response analysis
i1 <- irf(x1, n.ahead = 30)
plot(i1, scales = 'free_y')

# Example with same data set as in Luetkepohl and Nestunajev 2017
v1 <- vars::VAR(LN, p = 3, type = 'const')
x1 <- id.st(v1, c_fix = 167, gamma_fix = -2.77)
summary(x1)
plot(x1)

# Using a lagged endogenous transition variable
# In this example inflation with two lags
inf <- LN[-c(1, 449, 450), 2]*(1/sd(LN[-c(1, 449, 450), 2]))
x1_inf <- id.st(v1, c_fix = 4.41, gamma_fix = 0.49, transition_variable = inf)
summary(x1_inf)
plot(x1_inf)
```

---

irf *Impulse Response Functions for SVAR Models*

Description

Calculation of impulse response functions for an identified SVAR object 'svars' derived by function id.cvm( ),id.cv( ),id.dc( ), id.ngml( ) or id.st( ).

Usage

```
## S3 method for class 'svars'
irf(x, ..., n.ahead = 20)
```

Arguments

`x`	SVAR object of class "svars".
`...`	Currently not used.
`n.ahead`	Integer specifying the steps.

Value

A list with class attribute "svarirf" holding the impulse response functions as data frame.

References

Luetkepohl, H., 2005. New introduction to multiple time series analysis, Springer-Verlag, Berlin.

See Also

`id.cvm`, `id.dc`, `id.ngml`, `id.cv` or `id.st`

Examples

```
v1 <- vars::VAR(USA, lag.max = 10, ic = "AIC" )
x1 <- id.ngml(v1)
x2 <- irf(x1, n.ahead = 20)
plot(x2)
```

`js.test` *Chi-square test for joint hypotheses*

Description

Based on an existing bootstrap object, the test statistic allows to test joint hypotheses for selected entries of the structural matrix B. The test statistic reads as

$$(Rvec(\widehat{B}) - r)'R(\widehat{\mathrm{Cov}}[vec(B^*)])^{-1}R'(Rvec(\widehat{b} - r)) \sim \chi^2_J,$$

where $\widehat{\mathrm{Cov}}[vec(B^*)]$ is the estimated covariance of vectorized bootstrap estimates of structural parameters. The composite null hypothesis is $H_0 : Rvec(B) = r$.

Usage

```
js.test(x, R, r = NULL)
```

Arguments

`x`	Object of class 'sboot'
`R`	A J*K^2 selection matrix, where J is the number of hypotheses and K the number of time series.
`r`	A J*1 vector of restrictions

Value

A list of class "jstest" with elements

`test_statistic`	Test statistic
`p_value`	P-value
`R`	Selection matrix
`r`	Vector of restrictions

References

Herwartz, H., 2018. Hodges Lehmann detection of structural shocks - An analysis of macroeconomic dynamics in the Euro Area, Oxford Bulletin of Economics and Statistics

See Also

`mb.boot`, `wild.boot`

Examples

```
# data contains quarterly observations from 1965Q1 to 2008Q3
# x = output gap
# pi = inflation
# i = interest rates
v1 <- vars::VAR(USA, lag.max = 10, ic = "AIC" )
x1 <- id.dc(v1)

# Bootstrapping of SVAR
bb <- wild.boot(x1, nboot = 1000, n.ahead = 30)

# Testing the hypothesis of a lower triangular matrix as
# relation between structural and reduced form errors
R <- rbind(c(0,0,0,1,0,0,0,0,0), c(0,0,0,0,0,0,1,0,0),
           c(0,0,0,0,0,0,0,1,0))
c.test <- js.test(bb, R)
summary(c.test)
```

`LN` *Interaction between monetary policy and the stock market*

Description

A five dimensional time series model which is commonly used to analyze the interaction between monetary policy and the stock market.
Monthly observations from 1970M1 to 2007M6:

q	Linearly detrended log of an industrial production index
pi	Annual change in the log of consumer prices (CPI index) (x100)
c	annual change in the log of the World Bank (non energy) commodity price index (x100)
s	Log of the real S&P500 stock price index deflated by the consumer price index to measure the real stock prices; the series
r	Interest rate on Federal funds

All series, with exception of the commodity price index (c), are taken from the FRED database and transformed as in Luetkepohl & Netsunajev (2017). The commodity price index comes from the World Bank. A more detailed description of the data and a corresponding VAR model implementation can be found in Luetkepohl & Netsunajev (2017).

Usage

```
LN
```

Format

A `data.frame` containing 450 observations on 5 variables.

Source

Luetkepohl H., Netsunajev A., 2017. "Structural vector autoregressions with smooth transition in variances."
Journal of Economic Dynamics and Control, 84, 43 - 57. ISSN 0165-1889.

`mb.boot` *Moving block bootstrap for IRFs of identified SVARs*

Description

Calculating confidence bands for impulse response via moving block bootstrap

Usage

```
mb.boot(
  x,
  design = "recursive",
  b.length = 15,
  n.ahead = 20,
  nboot = 500,
  nc = 1,
  dd = NULL,
  signrest = NULL,
  signcheck = TRUE,
  itermax = 300,
  steptol = 200,
  iter2 = 50
)
```

Arguments

`x`	SVAR object of class "svars"
`design`	character. If design="fixed", a fixed design bootstrap is performed. If design="recursive", a recursive design bootstrap is performed.
`b.length`	Integer. Length of each block
`n.ahead`	Integer specifying the steps
`nboot`	Integer. Number of bootstrap iterations
`nc`	Integer. Number of processor cores
`dd`	Object of class 'indepTestDist'. A simulated independent sample of the same size as the data. If not supplied, it will be calculated by the function
`signrest`	A list with vectors containing 1 and -1, e.g. c(1,-1,1), indicating a sign pattern of specific shocks to be tested with the help of the bootstrap samples.
`signcheck`	Boolean. Whether the sign pattern should be checked for each bootstrap iteration. Note that this procedure is computationally extremely demanding for high dimensional VARs, since the number of possible permutations of B is K!, where K is the number of variables in the VAR.
`itermax`	Integer. Maximum number of iterations for DEoptim
`steptol`	Numeric. Tolerance for steps without improvement for DEoptim
`iter2`	Integer. Number of iterations for the second optimization

Value

A list of class "sboot" with elements

`true`	Point estimate of impulse response functions
`bootstrap`	List of length "nboot" holding bootstrap impulse response functions
`SE`	Bootstrapped standard errors of estimated covariance decomposition (only if "x" has method "Cramer von-Mises", or "Distance covariances")
`nboot`	Number of bootstrap iterations
`design`	character. Whether a fixed design or recursive design bootstrap is performed
`b_length`	Length of each block
`point_estimate`	Point estimate of covariance decomposition
`boot_mean`	Mean of bootstrapped covariance decompositions
`signrest`	Evaluated sign pattern
`sign_complete`	Frequency of appearance of the complete sign pattern in all bootstrapped covariance decompositions
`sign_part`	Frequency of bootstrapped covariance decompositions which conform the complete predetermined sign pattern. If signrest=NULL, the frequency of bootstrapped covariance decompositions that hold the same sign pattern as the point estimate is provided.
`sign_part`	Frequency of single shocks in all bootstrapped covariance decompositions which accord to a specific predetermined sign pattern

cov_bs	Covariance matrix of bootstrapped parameter in impact relations matrix
method	Used bootstrap method
VAR	Estimated input VAR object

References

Brueggemann, R., Jentsch, C., and Trenkler, C., 2016. Inference in VARs with conditional heteroskedasticity of unknown form. Journal of Econometrics 191, 69-85.
Herwartz, H., 2017. Hodges Lehmann detection of structural shocks - An analysis of macroeconomic dynamics in the Euro Area, Oxford Bulletin of Economics and Statistics.

See Also

id.cvm, id.dc, id.ngml, id.garch, id.cv or id.st

Examples

```
# data contains quarterly observations from 1965Q1 to 2008Q3
# x = output gap
# pi = inflation
# i = interest rates
set.seed(23211)
v1 <- vars::VAR(USA, lag.max = 10, ic = "AIC" )
x1 <- id.dc(v1)
summary(x1)

# impulse response analysis with confidence bands
# Checking how often theory based impact relations appear
signrest <- list(demand = c(1,1,1), supply = c(-1,1,1), money = c(-1,-1,1))
bb <- mb.boot(x1, b.length = 15, nboot = 500, n.ahead = 30, nc = 1, signrest = signrest)
summary(bb)

# Plotting IRFs with confidance bands
plot(bb, lowerq = 0.16, upperq = 0.84)

# With different confidence levels
plot(bb, lowerq = c(0.05, 0.1, 0.16), upperq = c(0.95, 0.9, 0.84))

# Halls percentile
plot(bb, lowerq = 0.16, upperq = 0.84, percentile = 'hall')

# Bonferroni bands
plot(bb, lowerq = 0.16, upperq = 0.84, percentile = 'bonferroni')
```

stability *Structural stability of a VAR(p)*

Description

Computes an empirical fluctuation process according to a specified method from the generalized fluctuation test framework. The test utilises the function `efp()` and its methods from package '`strucchange`'. Additionally, the function provides the option to compute a multivariate chow test.

Usage

```
## S3 method for class 'varest'
stability(
  x,
  type = c("OLS-CUSUM", "Rec-CUSUM", "Rec-MOSUM", "OLS-MOSUM", "RE", "ME", "Score-CUSUM",
    "Score-MOSUM", "fluctuation", "mv-chow-test"),
  h = 0.15,
  dynamic = FALSE,
  rescale = TRUE,
  ...
)
```

Arguments

`x`	Object of class '`varest`'; generated by `VAR`().
`type`	Specifies which type of fluctuation process will be computed, the default is '`OLS-CUSUM`'. For details see:`efp` and `chow.test`.
`h`	A numeric from interval (0,1) specifying the bandwidth. Determines the size of the data window relative to sample size (for '`MOSUM`', '`ME`' and '`mv-chow-test`' only).
`dynamic`	Logical. If '`TRUE`' the lagged observations are included as a regressor (not if '`type`' is '`mv-chow-test`').
`rescale`	Logical. If '`TRUE`' the estimates will be standardized by the regressor matrix of the corresponding subsample; if '`FALSE`' the whole regressor matrix will be used. (only if '`type`' is either '`RE`' or '`E`').
`...`	Ellipsis, is passed to `strucchange::sctest()`, as default.

Details

For details, please refer to documentation `efp` and `chow.test`.

Value

A list with either class attribute '`varstabil`' or '`chowpretest`' holding the following elements in case of class '`varstabil`':

stability	A list with objects of class 'efp'; length is equal to the dimension of the VAR.
names	Character vector containing the names of the endogenous variables.
K	An integer of the VAR dimension.

In case of class 'chowpretest' the list consists of the following elements:

teststat_bp	A vector containing the calculated break point test statistics for all considered break points.
teststat_sp	A vector containing the calculated sample split test statistics for all considered sample splits.
from	An integer sepcifying the first observation as possible break date.
to	An integer sepcifying the last observation as possible break date.
var	A list with objects of class 'varest'
break_point	Logical, if the break point test should be the benchmark for later analysis.

Author(s)

Bernhard Pfaff, Alexander Lange, Bernhard Dalheimer, Simone Maxand, Helmut Herwartz

References

Zeileis, A., F. Leisch, K. Hornik and C. Kleiber (2002), strucchange: An R Package for Testing for Structural Change in Linear Regression Models, *Journal of Statistical Software*, **7(2)**: 1-38, doi:10.18637/jss.v007.i02

and see the references provided in the reference section of efp and chow.test, too.

See Also

VAR, plot, efp, chow.test

Examples

```
data(Canada)
var.2c <- VAR(Canada, p = 2, type = "const")
var.2c.stabil <- stability(var.2c, type = "OLS-CUSUM")
var.2c.stabil
plot(var.2c.stabil)

data(USA)
v1 <- VAR(USA, p = 6)
x1 <- stability(v1, type = "mv-chow-test")
plot(x1)
```

svars — *svars: Data-driven identification of structural VAR models*

Description

This package implements data-driven identification methods for structural vector autoregressive (SVAR) models as described in Lange et al. (2021) doi:10.18637/jss.v097.i05. Based on an existing VAR model object, the structural impact matrix B may be obtained via different forms of heteroskedasticity or independent components.

Details

The main functions to retrieve structural impact matrices are:

- `id.cv` Identification via changes in volatility,
- `id.cvm` Independence-based identification of SVAR models based on Cramer-von Mises distance,
- `id.dc` Independence-based identification of SVAR models based on distance covariances,
- `id.garch` Identification through patterns of conditional heteroskedasticity,
- `id.ngml` Identification via Non-Gaussian maximum likelihood,
- `id.st` Identification by means of smooth transition in covariance.

All of these functions require an estimated var object. Currently the classes 'vars' and 'vec2var' from the `vars` package, 'nlVar', which includes both VAR and VECM, from the `tsDyn` package as well as the list from `MTS` package are supported. Besides these core functions, additional tools to calculate confidence bands for impulse response functions using bootstrap techniques as well as the Chow-Test for structural changes are implemented. The `USA` dataset is used to showcase the functionalities in examples throughout the package.

Author(s)

- Alexander Lange <alexander.lange@uni-goettingen.de>
- Bernhard Dalheimer <bernhard.dalheimer@uni-goettingen.de>
- Helmut Herwartz <hherwartz@uni-goettingen.de>
- Simone Maxand <simone.maxand@helsinki.fi>

USA — *US macroeconomic time series*

Description

The time series of output gap (x), inflation (pi) and interest rate (r) are taken from the FRED database and transformed as in Herwartz & Ploedt (2016). The trivariate time series model is commonly used to analyze monetary policy shocks.
Quarterly observations from 1965Q1 to 2008Q3:

x	Percentage log-deviation of real GDP wrt the estimate of potential output by the Congressional Budget Office
pi	Annualized quarter-on-quarter growth of the GDP deflator
i	Interest rate on Federal funds

A more detailed description of the data and a corresponding VAR model implementation can be found in Herwartz & Ploedt (2016).

Usage

```
USA
```

Format

A `data.frame` containing 174 observations on 3 variables.

Source

Herwartz, H. & Ploedt, M., 2016. Simulation Evidence on Theory-based and Statistical Identification under Volatility Breaks, Oxford Bulletin of Economics and Statistics, 78, 94-112.
Data originally from FRED database of the Federal Reserve Bank of St. Louis.

`wild.boot` *Wild bootstrap for IRFs of identified SVARs*

Description

Calculating confidence bands for impulse response functions via wild bootstrap techniques (Goncalves and Kilian, 2004).

Usage

```
wild.boot(
  x,
  design = "fixed",
  distr = "rademacher",
  n.ahead = 20,
  nboot = 500,
  nc = 1,
  dd = NULL,
  signrest = NULL,
  signcheck = TRUE,
  itermax = 300,
  steptol = 200,
  iter2 = 50,
  rademacher = "deprecated"
)
```

Arguments

`x`	SVAR object of class "svars"
`design`	character. If design="fixed", a fixed design bootstrap is performed. If design="recursive", a recursive design bootstrap is performed.
`distr`	character. If distr="rademacher", the Rademacher distribution is used to generate the bootstrap samples. If distr="mammen", the Mammen distribution is used. If distr = "gaussian", the gaussian distribution is used.
`n.ahead`	Integer specifying the steps
`nboot`	Integer. Number of bootstrap iterations
`nc`	Integer. Number of processor cores
`dd`	Object of class 'indepTestDist'. A simulated independent sample of the same size as the data. roxIf not supplied, it will be calculated by the function
`signrest`	A list with vectors containing 1 and -1, e.g. c(1,-1,1), indicating a sign pattern of specific shocks to be tested with the help of the bootstrap samples.
`signcheck`	Boolean. Whether the sign pattern should be checked for each bootstrap iteration. Note that this procedure is computationally extremely demanding for high dimensional VARs, since the number of possible permutations of B is K!, where K is the number of variables in the VAR.

`itermax`	Integer. Maximum number of iterations for DEoptim
`steptol`	Integer. Tolerance for steps without improvement for DEoptim
`iter2`	Integer. Number of iterations for the second optimization
`rademacher`	deprecated, use "design" instead.

Value

A list of class "sboot" with elements

`true`	Point estimate of impulse response functions
`bootstrap`	List of length "nboot" holding bootstrap impulse response functions
`SE`	Bootstrapped standard errors of estimated covariance decomposition (only if "x" has method "Cramer von-Mises", or "Distance covariances")
`nboot`	Number of bootstrap iterations
`distr`	Character, whether the Gaussian, Rademacher or Mammen distribution is used in the bootstrap
`design`	character. Whether a fixed design or recursive design bootstrap is performed
`point_estimate`	Point estimate of covariance decomposition
`boot_mean`	Mean of bootstrapped covariance decompositions
`signrest`	Evaluated sign pattern
`sign_complete`	Frequency of appearance of the complete sign pattern in all bootstrapped covariance decompositions
`sign_part`	Frequency of bootstrapped covariance decompositions which conform the complete predetermined sign pattern. If signrest=NULL, the frequency of bootstrapped covariance decompositions that hold the same sign pattern as the point estimate is provided.
`sign_part`	Frequency of single shocks in all bootstrapped covariance decompositions which accord to a specific predetermined sign pattern
`cov_bs`	Covariance matrix of bootstrapped parameter in impact relations matrix
`method`	Used bootstrap method
`VAR`	Estimated input VAR object

References

Goncalves, S., Kilian, L., 2004. Bootstrapping autoregressions with conditional heteroskedasticity of unknown form. Journal of Econometrics 123, 89-120.
Herwartz, H., 2017. Hodges Lehmann detection of structural shocks - An analysis of macroeconomic dynamics in the Euro Area, Oxford Bulletin of Economics and Statistics

See Also

`id.cvm`, `id.dc`, `id.garch`, `id.ngml`, `id.cv` or `id.st`

Examples

```
# data contains quarterly observations from 1965Q1 to 2008Q3
# x = output gap
# pi = inflation
# i = interest rates
set.seed(23211)
v1 <- vars::VAR(USA, lag.max = 10, ic = "AIC" )
x1 <- id.dc(v1)
summary(x1)

# impulse response analysis with confidence bands
# Checking how often theory based impact relations appear
signrest <- list(demand = c(1,1,1), supply = c(-1,1,1), money = c(-1,-1,1))
bb <- wild.boot(x1, nboot = 500, n.ahead = 30, nc = 1, signrest = signrest)
summary(bb)

# Plotting IRFs with confidance bands
plot(bb, lowerq = 0.16, upperq = 0.84)

# With different confidence levels
plot(bb, lowerq = c(0.05, 0.1, 0.16), upperq = c(0.95, 0.9, 0.84))

# Halls percentile
plot(bb, lowerq = 0.16, upperq = 0.84, percentile = 'hall')

# Bonferroni bands
plot(bb, lowerq = 0.16, upperq = 0.84, percentile = 'bonferroni')
```

Index