

# Package ‘fnets’

December 2, 2022

**Type** Package

**Title** Factor-adjusted Network Estimation and Forecasting for High-dimensional Time Series

**Version** 0.1.1

**Maintainer** Haeran Cho <haeran.cho@bristol.ac.uk>

**Description** Implements methods for network estimation and forecasting of high-dimensional time series exhibiting strong serial and cross-sectional correlations under a factor-adjusted vector autoregressive model.

**Depends** R (>= 4.1.0)

**Imports** lpSolve,  
parallel,  
doParallel,  
foreach,  
MASS,  
fields,  
igraph,  
RColorBrewer

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.2

## R topics documented:

common.predict . . . . .	2
factor.number . . . . .	3
fnets . . . . .	5
fnets.factor.model . . . . .	9
fnets.var . . . . .	11
idio.predict . . . . .	13
par.lrpc . . . . .	14
plot.fnets . . . . .	16
predict.fm . . . . .	17

predict.fnets . . . . .	18
sim.restricted . . . . .	19
sim.unrestricted . . . . .	20
sim.var . . . . .	21
threshold . . . . .	22
<b>Index</b>	<b>24</b>

---

<code>common.predict</code>	<i>Forecasting the factor-driven common component</i>
-----------------------------	---

---

**Description**

Produces forecasts of the common component for a given forecasting horizon by estimating the best linear predictors

**Usage**

```
common.predict(object, x, h = 1, forecast.restricted = TRUE, r = c("ic", "er"))
```

**Arguments**

<code>object</code>	fnets object
<code>x</code>	input time series matrix, with each row representing a variable
<code>h</code>	forecasting horizon
<code>forecast.restricted</code>	whether to forecast using a restricted or unrestricted, blockwise VAR representation of the common component
<code>r</code>	number of restricted factors, or a string specifying the factor number selection method when <code>forecast.restricted = TRUE</code> ; possible values are: <ul style="list-style-type: none"><li>• "ic" information criteria of Bai and Ng (2002)</li><li>• "er" eigenvalue ratio</li></ul>

**Value**

a list containing	
<code>is</code>	in-sample estimator of the common component
<code>fc</code>	forecasts of the common component for a given forecasting horizon <code>h</code>
<code>r</code>	restricted factor number
<code>h</code>	forecast horizon

## References

- Ahn, S. C. & Horenstein, A. R. (2013) Eigenvalue ratio test for the number of factors. *Econometrica*, 81(3), 1203–1227.
- Barigozzi, M., Cho, H. & Owens, D. (2022) FNETS: Factor-adjusted network estimation and forecasting for high-dimensional time series. *arXiv preprint arXiv:2201.06110*.
- Forni, M., Hallin, M., Lippi, M. & Reichlin, L. (2005). The generalized dynamic factor model: one-sided estimation and forecasting. *Journal of the American Statistical Association*, 100(471), 830–840.
- Forni, M., Hallin, M., Lippi, M. & Zaffaroni, P. (2017). Dynamic factor models with infinite-dimensional factor space: Asymptotic analysis. *Journal of Econometrics*, 199(1), 74–92.
- Owens, D., Cho, H. & Barigozzi, M. (2022)

## Examples

```
set.seed(123)
n <- 500
p <- 50
common <- sim.unrestricted(n, p)
idio <- sim.var(n, p)
x <- common$data + idio$data
out <- fnets(x, q = NULL, var.order = 1, var.method = "lasso", do.lrpc = FALSE)
cpre <- common.predict(out, x, h = 1, r = NULL)
ipre <- idio.predict(out, x, cpre, h = 1)
```

---

factor.number

*Factor number selection methods*

---

## Description

Methods to estimate the number of factor. Either maximises the ratio of successive eigenvalues, or minimises an information criterion over sub-samples of the data. For `restricted = FALSE`, the three information criteria proposed in Hallin and Liška (2007) (`pen.op = 1, 2 or 3`) and their variations with logarithm taken on the cost (`pen.op = 4, 5 or 6`) are implemented, with `pen.op = 5` recommended as a default choice based on numerical experiments. For `restricted = TRUE`, the three information criteria in Owens, Cho, and Barigozzi (2022) are implemented, with `pen.op = 2` recommended by default.

## Usage

```
factor.number(
  x,
  fm.restricted = FALSE,
  method = c("ic", "er"),
  q.max = NULL,
  mm = NULL,
  w = NULL,
```

```

do.plot = FALSE,
center = TRUE
)

```

### Arguments

x	input time series matrix, with each row representing a variable
fm.restricted	whether to estimate the number of restricted or unrestricted factors
method	A string specifying the factor number selection method; possible values are: <ul style="list-style-type: none"> <li>• "ic" information criteria-based methods of Alessi, Barigozzi &amp; Capasso (2010) when fm.restricted = TRUE or Hallin and Liška (2007) when fm.restricted = FALSE modifying Bai and Ng (2002)</li> <li>• "er" eigenvalue ratio of Ahn and Horenstein (2013)</li> </ul>
q.max	maximum number of factors; if q.max = NULL, a default value is selected as $\min(50, \text{floor}(\sqrt{\min(\text{dim}(x)[2] - 1, \text{dim}(x)[1])}))$
mm	bandwidth; defaults to $\text{floor}(4 * (\text{dim}(x)[2] / \log(\text{dim}(x)[2]))^{(1/3)})$
w	vector of length $2 * \text{mm} + 1$ containing symmetric weights; if w = NULL, default weights are generated using the Bartlett kernel and mm
do.plot	whether to plot the information criteria values
center	whether to de-mean the input x row-wise

### Details

For further details, see references.

### Value

if method = "ic", a list containing

q.hat	a vector containing minimisers of the six information criteria
sv	a list containing the singular value decomposition of $\Sigma_x$

and if restricted = FALSE

Gamma_x	an array containing the estimates of the autocovariance matrices of x at $2 * \text{mm} + 1$ lags
Sigma_x	an array containing the estimates of the spectral density matrices of x at $2 * \text{mm} + 1$ Fourier frequencies

otherwise

q.hat	the maximiser of the eigenvalue ratio
pca	dynamic or static pca output

## References

- Ahn, S. C. & Horenstein, A. R. (2013) Eigenvalue ratio test for the number of factors. *Econometrica*, 81(3), 1203–1227.
- Alessi, L., Barigozzi, M., and Capasso, M. (2010) Improved penalization for determining the number of factors in approximate factor models. *Statistics & Probability Letters*, 80(23-24):1806–1813.
- Bai, J. & Ng, S. (2002) Determining the number of factors in approximate factor models. *Econometrica*. 70: 191-221.
- Hallin, M. & Liška, R. (2007) Determining the number of factors in the general dynamic factor model. *Journal of the American Statistical Association*, 102(478), 603–617.
- Owens, D., Cho, H. & Barigozzi, M. (2022)

## Examples

```
library(fnets)

set.seed(123)
n <- 500
p <- 50
common <- sim.unrestricted(n, p)
idio <- sim.var(n, p)
x <- common$data * apply(idio$data, 1, sd) / apply(common$data, 1, sd) + idio$data

hl <- factor.number(x, fm.restricted = FALSE, do.plot = TRUE)
hl$q.hat
library(fnets)

set.seed(123)
n <- 500
p <- 50
common <- sim.restricted(n, p)
idio <- sim.var(n, p)
x <- common$data * apply(idio$data, 1, sd) / apply(common$data, 1, sd) + idio$data

abc <- factor.number(x, fm.restricted = TRUE, do.plot = TRUE)
abc$q.hat

er <- factor.number(x, method = "er", fm.restricted = TRUE, do.plot = TRUE)
er$q.hat
```

---

fnets

---

*Factor-adjusted network estimation*


---

## Description

Operating under a factor-adjusted vector autoregressive (VAR) model, the function estimates the spectral density and autocovariance matrices of the factor-driven common component and the idiosyncratic VAR process, the impulse response functions and common shocks for the common component, and VAR parameters, innovation covariance matrix and long-run partial correlations for the idiosyncratic component.

**Usage**

```
fnets(
  x,
  center = TRUE,
  fm.restricted = FALSE,
  q = c("ic", "er"),
  ic.op = NULL,
  kern.bw = NULL,
  common.args = list(factor.var.order = NULL, max.var.order = NULL, trunc.lags = 20,
    n.perm = 10),
  var.order = 1,
  var.method = c("lasso", "ds"),
  var.args = list(tuning = c("cv", "bic"), n.iter = 100, tol = 0, n.cores =
    min(parallel::detectCores() - 1, 3)),
  do.threshold = FALSE,
  do.lrpc = TRUE,
  lrpc.adaptive = FALSE,
  tuning.args = list(n.folds = 1, penalty = NULL, path.length = 10, do.plot = FALSE)
)
```

**Arguments**

<code>x</code>	input time series matrix, with each row representing a variable
<code>center</code>	whether to de-mean the input <code>x</code> row-wise
<code>fm.restricted</code>	whether to estimate a restricted factor model using static PCA
<code>q</code>	<p>Either the number of factors or a string specifying the factor number selection method; possible values are:</p> <ul style="list-style-type: none"> <li>• "ic" information criteria-based methods of Alessi, Barigozzi &amp; Capasso (2010) when <code>fm.restricted = TRUE</code> or Hallin and Liška (2007) when <code>fm.restricted = FALSE</code> modifying Bai and Ng (2002)</li> <li>• "er" eigenvalue ratio of Ahn and Horenstein (2013)</li> </ul> <p>; see <a href="#">factor.number</a>.</p>
<code>ic.op</code>	choice of the information criterion penalty, see <a href="#">factor.number</a> for further details
<code>kern.bw</code>	kernel bandwidth for dynamic PCA; defaults to $\text{floor}(4 * (\text{dim}(x)[2] / \log(\text{dim}(x)[2]))^{(1/3)})$
<code>common.args</code>	<p>a list specifying the tuning parameters required for estimating the impulse response functions and common shocks. It contains:</p> <ul style="list-style-type: none"> <li>• <code>factor.var.order</code> order of the blockwise VAR representation of the common component. If <code>factor.var.order = NULL</code>, it is selected blockwise by Schwarz criterion</li> <li>• <code>max.var.order</code> maximum blockwise VAR order for the Schwarz criterion</li> <li>• <code>trunc.lags</code> truncation lag for impulse response function estimation</li> <li>• <code>n.perm</code> number of cross-sectional permutations involved in impulse response function estimation</li> </ul>
<code>var.order</code>	order of the idiosyncratic VAR process; if a vector of integers is supplied, the order is chosen via tuning

<code>var.method</code>	a string specifying the method to be adopted for idiosyncratic VAR process estimation; possible values are: <ul style="list-style-type: none"> <li>• "lasso" Lasso-type l1-regularised M-estimation</li> <li>• "ds" Dantzig Selector-type constrained l1-minimisation</li> </ul>
<code>var.args</code>	a list specifying the tuning parameters required for estimating the idiosyncratic VAR process. It contains: <ul style="list-style-type: none"> <li>• <code>tuning</code> a string specifying the selection procedure for <code>var.order</code> and <code>lambda</code>; possible values are: <ul style="list-style-type: none"> <li>– "cv" cross validation</li> <li>– "bic" information criterion</li> </ul> </li> <li>• <code>n.iter</code> maximum number of descent steps; applicable when <code>var.method = "lasso"</code></li> <li>• <code>tol</code> numerical tolerance for increases in the loss function; applicable when <code>var.method = "lasso"</code></li> <li>• <code>n.cores</code> number of cores to use for parallel computing, see <a href="#">makePSOCK-cluster</a>; applicable when <code>var.method = "ds"</code></li> </ul>
<code>do.threshold</code>	whether to perform adaptive thresholding of all parameter estimators with <a href="#">thresh-old</a>
<code>do.lrpc</code>	whether to estimate the long-run partial correlation
<code>lrpc.adaptive</code>	whether to use the adaptive estimation procedure
<code>tuning.args</code>	a list specifying arguments for tuning for selecting the tuning parameters involved in VAR parameter and (long-run) partial correlation matrix estimation. It contains: <ul style="list-style-type: none"> <li>• <code>n.folds</code> if <code>tuning = "cv"</code>, number of folds</li> <li>• <code>penalty</code> if <code>tuning = "bic"</code>, penalty multiplier between 0 and 1; if <code>penalty = NULL</code>, defaults to <math>1/(1+\exp(\dim(x)[1]/\dim(x)[2]))</math></li> <li>• <code>path.length</code> number of regularisation parameter values to consider; a sequence is generated automatically based in this value</li> <li>• <code>do.plot</code> whether to plot the output of the cross validation step</li> </ul>

## Details

See Barigozzi, Cho and Owens (2022) and Owens, Cho and Barigozzi (2022) for further details. List arguments do not need to be specified with all list components; any missing entries will be filled in with the default argument.

## Value

an S3 object of class `fnets`, which contains the following fields:

<code>q</code>	number of factors
<code>spec</code>	if <code>fm.restricted = FALSE</code> a list containing estimates of the spectral density matrices for <code>x</code> , common and idiosyncratic components
<code>acv</code>	a list containing estimates of the autocovariance matrices for <code>x</code> , common and idiosyncratic components

<code>loadings</code>	if <code>fm.restricted = TRUE</code> , factor loadings; if <code>fm.restricted = FALSE</code> and <code>q &gt;= 1</code> , a list containing estimators of the impulse response functions (as an array of dimension $(p, q, \text{trunc.lags} + 2)$ )
<code>factors</code>	if <code>fm.restricted = TRUE</code> , factor series; else, common shocks (an array of dimension $(q, n)$ )
<code>idio.var</code>	a list containing the following fields: <ul style="list-style-type: none"> <li>• beta estimate of VAR parameter matrix; each column contains parameter estimates for the regression model for a given variable</li> <li>• Gamma estimate of the innovation covariance matrix</li> <li>• lambda regularisation parameter</li> <li>• convergence returned when <code>var.method = "lasso"</code>; indicates whether a convergence criterion is met</li> <li>• <code>var.order</code> VAR order</li> </ul>
<code>lrpc</code>	see the output of <a href="#">par.lrpc</a>
<code>mean.x</code>	if <code>center = TRUE</code> , returns a vector containing row-wise sample means of <code>x</code> ; if <code>center = FALSE</code> , returns a vector of zeros
<code>var.method</code>	input parameter
<code>do.lrpc</code>	input parameter
<code>kern.bw</code>	input parameter

## References

- Ahn, S. C. & Horenstein, A. R. (2013) Eigenvalue ratio test for the number of factors. *Econometrica*, 81(3), 1203–1227.
- Alessi, L., Barigozzi, M., & Capasso, M. (2010) Improved penalization for determining the number of factors in approximate factor models. *Statistics & Probability Letters*, 80(23-24):1806–1813.
- Bai, J. & Ng, S. (2002) Determining the number of factors in approximate factor models. *Econometrica*. 70: 191-221.
- Barigozzi, M., Cho, H. & Owens, D. (2022) FNETS: Factor-adjusted network estimation and forecasting for high-dimensional time series. *arXiv preprint arXiv:2201.06110*.
- Hallin, M. & Liška, R. (2007) Determining the number of factors in the general dynamic factor model. *Journal of the American Statistical Association*, 102(478), 603–617.
- Owens, D., Cho, H. & Barigozzi, M. (2022)

## See Also

[predict.fnets](#), [plot.fnets](#)

## Examples

```
## Not run:
set.seed(123)
n <- 500
p <- 50
common <- sim.unrestricted(n, p)
```



```

idio <- sim.var(n, p)
x <- common$data + idio$data
out <- fnets(x,
  q = NULL, var.order = 1, var.method = "lasso", do.threshold = TRUE,
  do.lrpc = TRUE, tuning.args = list(n.folds = 1, path.length = 10, do.plot = TRUE)
)
pre <- predict(out, x, h = 1, common.method = "unrestricted")
plot(out, type = "granger", display = "network")
plot(out, type = "lrpc", display = "heatmap")

## End(Not run)

```

---

fnets.factor.model      *Factor model estimation*


---

## Description

Unrestricted and restricted factor model estimation

## Usage

```

fnets.factor.model(
  x,
  center = TRUE,
  fm.restricted = FALSE,
  q = c("ic", "er"),
  ic.op = NULL,
  kern.bw = NULL,
  common.args = list(factor.var.order = NULL, max.var.order = NULL, trunc.lags = 20,
    n.perm = 10)
)

```

## Arguments

x	input time series matrix, with each row representing a variable
center	whether to de-mean the input x row-wise
fm.restricted	whether to estimate a restricted factor model using static PCA
q	Either a string specifying the factor number selection method when fm.restricted = TRUE; possible values are: <ul style="list-style-type: none"> <li>"ic" information criteria of Hallin and Liška (2007) or Bai and Ng (2002), see <a href="#">factor.number</a></li> <li>"er" eigenvalue ratio</li> </ul> ; or the number of unrestricted factors.
ic.op	choice of the information criterion penalty, see <a href="#">hl.factor.number</a> or <a href="#">abc.factor.number</a> for further details
kern.bw	kernel bandwidth for dynamic PCA; defaults to $4 * \text{floor}((\text{dim}(x)[2]/\log(\text{dim}(x)[2]))^{(1/3)})$

<code>common.args</code>	<p>a list specifying the tuning parameters required for estimating the impulse response functions and common shocks. It contains:</p> <ul style="list-style-type: none"> <li>• <code>factor.var.order</code> order of the blockwise VAR representation of the common component. If <code>factor.var.order = NULL</code>, it is selected blockwise by Schwarz criterion</li> <li>• <code>max.var.order</code> maximum blockwise VAR order for the Schwarz criterion</li> <li>• <code>trunc.lags</code> truncation lag for impulse response function estimation</li> <li>• <code>n.perm</code> number of cross-sectional permutations involved in impulse response function estimation</li> </ul>
--------------------------	---

### Details

See Barigozzi, Cho and Owens (2022) for further details.

### Value

an S3 object of class `fm`, which contains the following fields:

<code>q</code>	number of factors
<code>spec</code>	if <code>fm.restricted = FALSE</code> a list containing estimates of the spectral density matrices for <code>x</code> , common and idiosyncratic components
<code>acv</code>	a list containing estimates of the autocovariance matrices for <code>x</code> , common and idiosyncratic components
<code>loadings</code>	if <code>fm.restricted = TRUE</code> , factor loadings; if <code>fm.restricted = FALSE</code> and <code>q &gt;= 1</code> , a list containing estimators of the impulse response functions (as an array of dimension $(p, q, \text{trunc.lags} + 2)$ )
<code>factors</code>	if <code>fm.restricted = TRUE</code> , factor series; else, common shocks (an array of dimension $(q, n)$ )
<code>mean.x</code>	if <code>center = TRUE</code> , returns a vector containing row-wise sample means of <code>x</code> ; if <code>center = FALSE</code> , returns a vector of zeros

### References

- Alessi, L., Barigozzi, M., & Capasso, M. (2010) Improved penalization for determining the number of factors in approximate factor models. *Statistics & Probability Letters*, 80(23-24):1806–1813.
- Bai, J. & Ng, S. (2002) Determining the number of factors in approximate factor models. *Econometrica*. 70: 191-221.
- Barigozzi, M., Cho, H. & Owens, D. (2022) Factor-adjusted network estimation and forecasting for high-dimensional time series. *arXiv preprint arXiv:2201.06110*.
- Hallin, M. & Liška, R. (2007) Determining the number of factors in the general dynamic factor model. *Journal of the American Statistical Association*, 102(478), 603–617.
- Owens, D., Cho, H. & Barigozzi, M. (2022)

**Examples**

```
## Not run:
set.seed(123)
n <- 500
p <- 50
common <- sim.restricted(n, p)
x <- common$data
out <- fnets.factor.model(x, fm.restricted = TRUE)

## End(Not run)
```

fnets.var

*l1-regularised Yule-Walker estimation for VAR processes***Description**

Estimates the VAR parameter matrices via l1-regularised Yule-Walker estimation and innovation covariance matrix via constrained l1-minimisation.

**Usage**

```
fnets.var(
  x,
  center = TRUE,
  method = c("lasso", "ds"),
  lambda = NULL,
  var.order = 1,
  tuning.args = list(tuning = c("cv", "bic"), n.folds = 1, path.length = 10, do.plot =
    FALSE),
  do.threshold = FALSE,
  n.iter = 100,
  tol = 0,
  n.cores = min(parallel::detectCores() - 1, 3)
)
```

**Arguments**

x	input time series matrix, with each row representing a variable
center	whether to de-mean the input x row-wise
method	a string specifying the method to be adopted for VAR process estimation; possible values are: <ul style="list-style-type: none"> <li>• "lasso" Lasso-type l1-regularised M-estimation</li> <li>• "ds" Dantzig Selector-type constrained l1-minimisation</li> </ul>
lambda	regularisation parameter; if lambda = NULL, tuning is employed to select the parameter

<code>var.order</code>	order of the VAR process; if a vector of integers is supplied, the order is chosen via tuning
<code>tuning.args</code>	a list specifying arguments for tuning for selecting the regularisation parameter (and VAR order). It contains: <ul style="list-style-type: none"> <li>• <code>tuning</code> a string specifying the selection procedure for <code>var.order</code> and <code>lambda</code>; possible values are: <ul style="list-style-type: none"> <li>– <code>"cv"</code> cross validation</li> <li>– <code>"bic"</code> information criterion</li> </ul> </li> <li>• <code>n.folds</code> if <code>tuning = "cv"</code>, number of folds</li> <li>• <code>penalty</code> if <code>tuning = "bic"</code>, penalty multiplier between 0 and 1; if <code>penalty = NULL</code>, defaults to <math>1/(1+\exp(\dim(x)[1]/\dim(x)[2]))</math></li> <li>• <code>path.length</code> number of regularisation parameter values to consider; a sequence is generated automatically based in this value</li> <li>• <code>do.plot</code> whether to plot the output of the cross validation step</li> </ul>
<code>do.threshold</code>	whether to perform adaptive thresholding of VAR parameter estimator with <a href="#">threshold</a>
<code>n.iter</code>	maximum number of descent steps; applicable when <code>method = "lasso"</code>
<code>tol</code>	numerical tolerance for increases in the loss function; applicable when <code>method = "lasso"</code>
<code>n.cores</code>	number of cores to use for parallel computing, see <a href="#">makePSOCKcluster</a> ; applicable when <code>method = "ds"</code>

## Details

Further information can be found in Barigozzi, Cho and Owens (2022).

## Value

a list which contains the following fields:

<code>beta</code>	estimate of VAR parameter matrix; each column contains parameter estimates for the regression model for a given variable
<code>Gamma</code>	estimate of the innovation covariance matrix
<code>lambda</code>	regularisation parameter
<code>convergence</code>	returned when <code>method = "lasso"</code> ; indicates whether a convergence criterion is met
<code>var.order</code>	VAR order
<code>mean.x</code>	if <code>center = TRUE</code> , returns a vector containing row-wise sample means of <code>x</code> ; if <code>center = FALSE</code> , returns a vector of zeros

## References

- Barigozzi, M., Cho, H. & Owens, D. (2022) FNETS: Factor-adjusted network estimation and forecasting for high-dimensional time series. arXiv preprint arXiv:2201.06110.
- Owens, D., Cho, H. & Barigozzi, M. (2022)

**Examples**

```
library(fnets)

set.seed(123)
n <- 500
p <- 50
idio <- sim.var(n, p)
x <- idio$data

fv <- fnets.var(x,
  center = TRUE, method = "lasso", var.order = 1,
  tuning.args = list(tuning = "cv", n.folds = 1, path.length = 10, do.plot = TRUE)
)
norm(fv$beta - t(idio$A), "F") / norm(t(idio$A), "F")
```

---

idio.predict	<i>Forecasting idiosyncratic VAR process</i>
--------------	--

---

**Description**

Produces forecasts of the idiosyncratic VAR process for a given forecasting horizon by estimating the best linear predictors

**Usage**

```
idio.predict(object, x, cpre, h = 1)
```

**Arguments**

object	fnets object
x	input time series matrix, with each row representing a variable
cpre	output of <a href="#">common.predict</a>
h	forecast horizon

**Value**

a list containing	
is	in-sample estimator of the idiosyncratic component
fc	forecasts of the idiosyncratic component for a given forecasting horizon h
h	forecast horizon

**References**

Barigozzi, M., Cho, H. & Owens, D. (2022) FNETS: Factor-adjusted network estimation and forecasting for high-dimensional time series. arXiv preprint arXiv:2201.06110.

Owens, D., Cho, H. & Barigozzi, M. (2022)

**Examples**

```

set.seed(123)
n <- 500
p <- 50
common <- sim.unrestricted(n, p)
idio <- sim.var(n, p)
x <- common$data + idio$data
out <- fnets(x, q = NULL, var.order = 1, var.method = "lasso", do.lrpc = FALSE)
cpre <- common.predict(out, x, h = 1, r = NULL)
ipre <- idio.predict(out, x, cpre, h = 1)

```

---

par.lrpc	<i>Parametric estimation of long-run partial correlations of factor-adjusted VAR processes</i>
----------	--

---

**Description**

Returns a parametric estimate of long-run partial correlations of the VAR process from the VAR parameter estimates and the inverse of innovation covariance matrix obtained via constrained l1-minimisation.

**Usage**

```

par.lrpc(
  object,
  x,
  eta = NULL,
  tuning.args = list(n.folds = 1, path.length = 10, do.plot = FALSE),
  lrpc.adaptive = FALSE,
  eta.adaptive = NULL,
  do.correct = TRUE,
  do.threshold = FALSE,
  do.plot = FALSE,
  n.cores = min(parallel::detectCores() - 1, 3)
)

```

**Arguments**

object	fnets object
x	input time series matrix; with each row representing a variable
eta	regularisation parameter; if eta = NULL, it is selected by cross validation
tuning.args	a list specifying arguments for the cross validation procedure for selecting the tuning parameter involved in long-run partial correlation matrix estimation. It contains: <ul style="list-style-type: none"> <li>• n.folds number of folds</li> </ul>

	<ul style="list-style-type: none"> <li>• <code>path.length</code> number of regularisation parameter values to consider; a sequence is generated automatically based in this value</li> <li>• <code>do.plot</code> whether to plot the output of the cross validation step</li> </ul>
<code>lrpc.adaptive</code>	whether to use the adaptive estimation procedure
<code>eta.adaptive</code>	regularisation parameter for Step 1 of the adaptive estimation procedure; if <code>eta.adaptive = NULL</code> , defaults to $2 * \sqrt{\log(\dim(x)[1])/\dim(x)[2]}$
<code>do.correct</code>	whether to correct for any negative entries in the diagonals of the inverse of long-run covariance matrix
<code>do.threshold</code>	whether to perform adaptive thresholding of Delta and Omega parameter estimators with <a href="#">threshold</a>
<code>do.plot</code>	whether to plot thresholding output
<code>n.cores</code>	number of cores to use for parallel computing, see <a href="#">makePSOCKcluster</a>

### Details

See Barigozzi, Cho and Owens (2022) for further details, and Cai, Liu and Zhou (2016) for further details on the adaptive estimation procedure.

### Value

a list containing

<code>Delta</code>	estimated inverse of the innovation covariance matrix
<code>Omega</code>	estimated inverse of the long-run covariance matrix
<code>pc</code>	estimated innovation partial correlation matrix
<code>lrpc</code>	estimated long-run partial correlation matrix
<code>eta</code>	regularisation parameter
<code>lrpc.adaptive</code>	input argument

### References

- Barigozzi, M., Cho, H. & Owens, D. (2022) FNETS: Factor-adjusted network estimation and forecasting for high-dimensional time series.
- Cai, T. T., Liu, W., & Zhou, H. H. (2016). Estimating sparse precision matrix: Optimal rates of convergence and adaptive estimation. *The Annals of Statistics*, 44(2), 455-488.
- Owens, D., Cho, H. & Barigozzi, M. (2022)

### Examples

```
## Not run:
set.seed(123)
n <- 500
p <- 50
common <- sim.unrestricted(n, p)
idio <- sim.var(n, p)
x <- common$data + idio$data
```

```

out <- fnets(x, q = NULL, var.method = "lasso", do.lrpc = FALSE)
plrpc <- par.lrpc(out, x, tuning.args = list(n.folds = 1, path.length = 10, do.plot = TRUE))
out$lrpc <- plrpc
out$do.lrpc <- TRUE
plot(out, type = "pc", display = "network", threshold = .05)
plot(out, type = "lrpc", display = "heatmap", threshold = .05)

## End(Not run)

```

---

plot.fnets

---

*Plotting the networks estimated by fnets*


---

## Description

Plotting method for S3 objects of class `fnets`. Produces a plot visualising three networks underlying factor-adjusted VAR processes: (i) directed network representing Granger causal linkages, as given by estimated VAR transition matrices summed across the lags, (ii) undirected network representing contemporaneous linkages after accounting for lead-lag dependence, as given by partial correlations of VAR innovations, (iii) undirected network summarising (i) and (ii) as given by long-run partial correlations of VAR processes.

## Usage

```

## S3 method for class 'fnets'
plot(
  x,
  type = c("granger", "pc", "lrpc"),
  display = c("network", "heatmap"),
  names = NA,
  groups = NA,
  threshold = 0,
  ...
)

```

## Arguments

<code>x</code>	<code>fnets</code> object
<code>type</code>	a string specifying which of the above three networks (i)–(iii) to visualise; possible values are <ul style="list-style-type: none"> <li>• <code>"granger"</code> directed network representing Granger causal linkages</li> <li>• <code>"pc"</code> undirected network representing contemporaneous linkages; available when <code>x\$do.lrpc = TRUE</code></li> <li>• <code>"lrpc"</code> undirected network summarising Granger causal and contemporaneous linkages; available when <code>x\$do.lrpc = TRUE</code></li> </ul>
<code>display</code>	a string specifying how to visualise the network; possible values are: <ul style="list-style-type: none"> <li>• <code>"network"</code> as an <code>igraph</code> object, see <a href="#">plot.igraph</a></li> </ul>



	<ul style="list-style-type: none"> <li>• "heatmap" as a heatmap, see <a href="#">imagePlot</a></li> </ul>
names	a character vector containing the names of the vertices
groups	an integer vector denoting any group structure of the vertices
threshold	if $\text{threshold} > 0$ , hard thresholding is performed on the matrix giving rise to the network of interest
...	additional arguments

### Details

See Barigozzi, Cho and Owens (2022) for further details.

### References

Barigozzi, M., Cho, H. & Owens, D. (2022) FNETS: Factor-adjusted network estimation and forecasting for high-dimensional time series. arXiv preprint arXiv:2201.06110.

Owens, D., Cho, H. & Barigozzi, M. (2022)

### See Also

[fnets](#)

---

predict.fm	<i>Forecasting for factor models</i>
------------	--------------------------------------

---

### Description

Produces forecasts of the data for a given forecasting horizon by estimating the best linear predictors of the common component

### Usage

```
## S3 method for class 'fm'
predict(object, x, h = 1, forecast.restricted = TRUE, r = c("ic", "er"), ...)
```

### Arguments

object	fm object
x	input time series matrix, with each row representing a variable
h	forecasting horizon
forecast.restricted	whether to forecast using a restricted or unrestricted, blockwise VAR representation of the common component
r	number of restricted factors, or a string specifying the factor number selection method when <code>forecast.restricted = TRUE</code> ; possible values are: <ul style="list-style-type: none"> <li>• "ic" information criteria of Alessi, Barigozzi &amp; Capasso (2010)</li> <li>• "er" eigenvalue ratio</li> </ul>
...	not used

**Value**

a list containing

is	in-sample predictions
forecast	forecasts for the given forecasting horizon
r	factor number

**References**

Ahn, S. C. & Horenstein, A. R. (2013) Eigenvalue ratio test for the number of factors. *Econometrica*, 81(3), 1203–1227.

Alessi, L., Barigozzi, M., & Capasso, M. (2010) Improved penalization for determining the number of factors in approximate factor models. *Statistics & Probability Letters*, 80(23-24):1806–1813.

Barigozzi, M., Cho, H. & Owens, D. (2022) Factor-adjusted network estimation and forecasting for high-dimensional time series. *arXiv preprint arXiv:2201.06110*.

Owens, D., Cho, H. & Barigozzi, M. (2022)

**See Also**

[fnets.factor.model](#), [common.predict](#)

---

predict.fnets	<i>Forecasting by fnets</i>
---------------	-----------------------------

---

**Description**

Produces forecasts of the data for a given forecasting horizon by separately estimating the best linear predictors of common and idiosyncratic components

**Usage**

```
## S3 method for class 'fnets'
predict(object, x, h = 1, forecast.restricted = TRUE, r = c("ic", "er"), ...)
```

**Arguments**

object	fnets object
x	input time series matrix, with each row representing a variable
h	forecasting horizon
forecast.restricted	whether to forecast using a restricted or unrestricted, blockwise VAR representation of the common component
r	number of restricted factors, or a string specifying the factor number selection method when forecast.restricted = TRUE; possible values are: <ul style="list-style-type: none"> <li>• "ic" information criteria of Bai and Ng (2002)</li> <li>• "er" eigenvalue ratio</li> </ul>
...	not used

**Value**

a list containing

forecast	forecasts for the given forecasting horizon
common.pred	a list containing forecasting results for the common component
idio.pred	a list containing forecasting results for the idiosyncratic component
mean.x	mean.x argument from object

**References**

Ahn, S. C. & Horenstein, A. R. (2013) Eigenvalue ratio test for the number of factors. *Econometrica*, 81(3), 1203–1227.

Barigozzi, M., Cho, H. & Owens, D. (2022) FNETS: Factor-adjusted network estimation and forecasting for high-dimensional time series. *arXiv preprint arXiv:2201.06110*.

Owens, D., Cho, H. & Barigozzi, M. (2022)

**See Also**

[fnets](#), [common.predict](#), [idio.predict](#)

**Examples**

```
set.seed(123)
n <- 500
p <- 50
common <- sim.restricted(n, p)
idio <- sim.var(n, p)
x <- common$data + idio$data
out <- fnets(x, q = 2, var.order = 1, var.method = "lasso", do.lrpc = FALSE)
cpre.unr <- common.predict(out, x, h = 1, forecast.restricted = FALSE, r = NULL)
cpre.res <- common.predict(out, x, h = 1, forecast.restricted = TRUE, r = NULL)
ipre <- idio.predict(out, x, cpre.res, h = 1)
```

---

sim.restricted

---

*Simulate data from a restricted factor model*


---

**Description**

Simulate the common component following an unrestricted factor model that admits a restricted representation; see the model (C2) in the reference.

**Usage**

```
sim.restricted(n, p, q = 2, heavy = FALSE)
```

**Arguments**

n	sample size
p	dimension
q	number of unrestricted factors; number of restricted factors is given by $2 * q$
heavy	if heavy = FALSE, common shocks are generated from <code>rnorm</code> whereas if heavy = TRUE, from <code>rt</code> with <code>df = 5</code> and then scaled by <code>sqrt(3 / 5)</code>

**Value**

a list containing	
data	generated series
q	number of factors
r	number of restricted factors

**References**

Barigozzi, M., Cho, H. & Owens, D. (2022) FNETS: Factor-adjusted network estimation and forecasting for high-dimensional time series.

Owens, D., Cho, H. & Barigozzi, M. (2022)

**Examples**

```
common <- sim.restricted(500, 50)
```

---

sim.unrestricted	<i>Simulate data from an unrestricted factor model</i>
------------------	--

---

**Description**

Simulate the common component following an unrestricted factor model that does not admit a restricted representation; see the model (C1) in the reference.

**Usage**

```
sim.unrestricted(n, p, q = 2, heavy = FALSE)
```

**Arguments**

n	sample size
p	dimension
q	number of unrestricted factors
heavy	if heavy = FALSE, common shocks are generated from <code>rnorm</code> whereas if heavy = TRUE, from <code>rt</code> with <code>df = 5</code> and then scaled by <code>sqrt(3 / 5)</code>

**Value**

a list containing

data	generated series
q	number of factors

**References**

Barigozzi, M., Cho, H. & Owens, D. (2022) FNETS: Factor-adjusted network estimation and forecasting for high-dimensional time series. arXiv preprint arXiv:2201.06110

Owens, D., Cho, H. & Barigozzi, M. (2022)

**Examples**

```
common <- sim.unrestricted(500, 50)
```

---

sim.var	<i>Simulate a VAR(1) process</i>
---------	----------------------------------

---

**Description**

Simulate a VAR(1) process; see the reference for the generation of the transition matrix.

**Usage**

```
sim.var(n, p, Gamma = diag(1, p), heavy = FALSE)
```

**Arguments**

n	sample size
p	dimension
Gamma	innovation covariance matrix; ignored if heavy = TRUE
heavy	if heavy = FALSE, common shocks are generated from rnorm whereas if heavy = TRUE, from rt with df = 5 and then scaled by sqrt(3 / 5)

**Value**

a list containing

data	generated series
A	transition matrix
Gamma	innovation covariance matrix

## References

Barigozzi, M., Cho, H. & Owens, D. (2022) FNETS: Factor-adjusted network estimation and forecasting for high-dimensional time series.

Owens, D., Cho, H. & Barigozzi, M. (2022)

## Examples

```
idio <- sim.var(500, 50)
```

---

threshold	<i>Edge selection for VAR parameter, inverse innovation covariance, and long-run partial correlation matrices</i>
-----------	---

---

## Description

Threshold the entries of the input matrix at a data-driven level to perform edge selection

## Usage

```
threshold(mat, path.length = 500, do.plot = FALSE)
```

## Arguments

mat	input parameter matrix
path.length	number of candidate thresholds
do.plot	whether to plot thresholding output

## Details

See Liu, Zhang, and Liu (2021) for more information on the threshold selection process

## Value

a list which contains the following fields:

threshold	data-driven threshold
thr.mat	thresholded input matrix

## References

Barigozzi, M., Cho, H. & Owens, D. (2022) FNETS: Factor-adjusted network analysis for high-dimensional time series. arXiv preprint arXiv:2201.06110.

Liu, B., Zhang, X. & Liu, Y. (2021) Simultaneous Change Point Inference and Structure Recovery for High Dimensional Gaussian Graphical Models. *Journal of Machine Learning Research*, 22(274), 1–62.

Owens, D., Cho, H. & Barigozzi, M. (2022)

**Examples**

```
## Not run:  
set.seed(123)  
A <- diag(.7, 50) + rnorm(50^2, 0, .1)  
threshold.A <- threshold(A)  
  
## End(Not run)
```

# Index

abc.factor.number, [9](#)

common.predict, [2](#), [13](#), [18](#), [19](#)

factor.number, [3](#), [6](#), [9](#)

fnets, [5](#), [17](#), [19](#)

fnets.factor.model, [9](#), [18](#)

fnets.var, [11](#)

hl.factor.number, [9](#)

idio.predict, [13](#), [19](#)

imagePlot, [17](#)

makePSOCKcluster, [7](#), [12](#), [15](#)

par.lrpc, [8](#), [14](#)

plot.fnets, [8](#), [16](#)

plot.igraph, [16](#)

predict.fm, [17](#)

predict.fnets, [8](#), [18](#)

sim.restricted, [19](#)

sim.unrestricted, [20](#)

sim.var, [21](#)

threshold, [7](#), [12](#), [15](#), [22](#)