

# Package ‘fnets’

June 8, 2023

**Type** Package

**Title** Factor-Adjusted Network Estimation and Forecasting for High-Dimensional Time Series

**Version** 0.1.3

**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.

See Barigozzi, Cho and Owens (2022) <[arXiv:2201.06110](#)> for further descriptions of FNETS methodology and

Owens, Cho and Barigozzi (2023) <[arXiv:2301.11675](#)> accompanying the R package.

**Depends** R (>= 4.1.0)

**Imports** lpSolve,

parallel,

doParallel,

foreach,

MASS,

fields,

igraph,

RColorBrewer

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** false

**RoxygenNote** 7.1.2

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

## R topics documented:

factor.number . . . . .	2
fnets . . . . .	4
fnets.factor.model . . . . .	8

fnets.var . . . . .	10
network . . . . .	12
network.fnets . . . . .	12
par.lrpc . . . . .	14
plot.factor.number . . . . .	16
plot.fnets . . . . .	17
plot.threshold . . . . .	19
predict.fm . . . . .	20
predict.fnets . . . . .	21
print.factor.number . . . . .	22
print.fm . . . . .	23
print.fnets . . . . .	24
print.threshold . . . . .	25
sim.restricted . . . . .	26
sim.unrestricted . . . . .	27
sim.var . . . . .	28
threshold . . . . .	29

## Index 31

---

factor.number	<i>Factor number selection methods</i>
---------------	--

---

### Description

Methods to estimate the number of factor. When method = 'er', the factor number is estimated by maximising the ration of successive eigenvalues. When method = 'ic', the information criterion-methods discussed in Hallin and Liška (2007) (when fm.restricted = FALSE) and Alessi, Barigozzi and Capasso (2010) (when fm.restricted = TRUE) are implemented. The information criterion called by ic.op = 5 (as an argument to fnets or fnets.factor.model) is recommended by default.

### Usage

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

### Arguments

x	input time series matrix, with each row representing a variable and each column containing the observations at a given time
fm.restricted	whether to estimate the number of restricted or unrestricted factors
method	A string specifying the factor number selection method; possible values are:

- "ic" information criteria-based methods of Alessi, Barigozzi & Capasso (2010) when `fm.restricted = TRUE` or Hallin and Liška (2007) when `fm.restricted = FALSE`
  - "er" eigenvalue ratio of Ahn and Horenstein (2013) when `fm.restricted = TRUE` or Avarucci et al. (2022) when `fm.restricted = FALSE`
- `q.max`            maximum number of factors; if `q.max = NULL`, a default value is selected as `min(50, floor(sqrt(min(dim(x)[2] - 1, dim(x)[1])))`
- `center`            whether to de-mean the input `x` row-wise

## Details

For further details, see references.

## Value

S3 object of class `factor.number`. If `method = "ic"`, a vector containing minimisers of the six information criteria, otherwise, the maximiser of the eigenvalue ratio

## 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.
- Avarucci, M., Cavicchioli, M., Forni, M., & Zaffaroni, P. (2022) The main business cycle shock(s): Frequency-band estimation of the number of dynamic factors.
- 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) *fnets: An R Package for Network Estimation and Forecasting via Factor-Adjusted VAR Modelling*. arXiv preprint arXiv:2301.11675.

## See Also

[plot.factor.number](#), [print.factor.number](#)

## Examples

```
library(fnets)

## Alessi, Barigozzi, and Capasso method for restricted models
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)
```

```

print(abc)
plot(abc)

## Eigenvalue ratio method
er <- factor.number(x, method = "er", fm.restricted = TRUE)
print(er)
plot(er)

## Hallin and Liška method for unrestricted models
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)
print(hl)
plot(hl)

```

---

fnets

---

*Factor-adjusted network estimation*


---

## Description

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(n.iter = NULL, n.cores = min(parallel::detectCores() - 1, 3)),
  do.threshold = FALSE,
  do.lrpc = TRUE,
  lrpc.adaptive = FALSE,

```

```
tuning.args = list(tuning = c("cv", "bic"), n.folds = 1, penalty = NULL, path.length
= 10)
)
```

## Arguments

<code>x</code>	input time series matrix, with each row representing a variable and each column containing the observations at a given time
<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>	Either the number of factors or 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 <code>fm.restricted = TRUE</code> or Hallin and Liška (2007) when <code>fm.restricted = FALSE</code></li> <li>• "er" eigenvalue ratio of Ahn and Horenstein (2013) when <code>fm.restricted = TRUE</code> or Avarucci et al. (2022) when <code>fm.restricted = FALSE</code></li> </ul> see <a href="#">factor.number</a> .
<code>ic.op</code>	choice of the information criterion penalty, see <a href="#">factor.number</a> for further details
<code>kern.bw</code>	a positive integer specifying the kernel bandwidth for dynamic PCA; by default, it is set to $\text{floor}(4 * (\text{dim}(x)[2] / \log(\text{dim}(x)[2]))^{(1/3)})$ . When <code>fm.restricted = TRUE</code> , it is used to compute the number of lags for which autocovariance matrices are estimated
<code>common.args</code>	a list specifying the tuning parameters required for estimating the impulse response functions and common shocks. It contains: <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>n.iter</code> maximum number of descent steps, by default depends on <code>var.order</code>; 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="#">threshold</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>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>, positive integer number of folds</li> <li>• <code>penalty</code> if <code>tuning = "bic"</code>, penalty multiplier between 0 and 1; if <code>penalty = NULL</code>, it is set to <math>1/(1+\exp(\dim(x)[1]/\dim(x)[2]))</math> by default</li> <li>• <code>path.length</code> positive integer number of regularisation parameter values to consider; a sequence is generated automatically based in this value</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>• <code>beta</code> estimate of VAR parameter matrix; each column contains parameter estimates for the regression model for a given variable</li> <li>• <code>Gamma</code> estimate of the innovation covariance matrix</li> <li>• <code>lambda</code> regularisation parameter</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.
- Avarucci, M., Cavicchioli, M., Forni, M., & Zaffaroni, P. (2022) The main business cycle shock(s): Frequency-band estimation of the number of dynamic factors.
- 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) fnets: An R Package for Network Estimation and Forecasting via Factor-Adjusted VAR Modelling. *arXiv preprint arXiv:2301.11675*.

## See Also

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

## 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,
  do.threshold = TRUE,
  var.args = list(n.cores = 2)
)
pre <- predict(out, common.method = "unrestricted")
plot(out, type = "granger", display = "network")
plot(out, type = "lrpc", display = "heatmap")
```

---

fnets.factor.model      *Factor model estimation*


---

## Description

Performs factor modelling under either restricted (static) or unrestricted (dynamic) factor models

## 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 and each column containing the observations at a given time
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-based methods of Alessi, Barigozzi &amp; Capasso (2010) when fm.restricted = TRUE or Hallin and Liška (2007) when fm.restricted = FALSE</li> <li>"er" eigenvalue ratio of Ahn and Horenstein (2013) when fm.restricted = TRUE or Avarucci et al. (2022) when fm.restricted = FALSE</li> </ul> or the number of unrestricted factors, see <a href="#">factor.number</a>
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	a positive integer specifying the kernel bandwidth for dynamic PCA; by default, it is set to $\text{floor}(4 * (\text{dim}(x)[2] / \log(\text{dim}(x)[2]))^{(1/3)})$ . When fm.restricted = TRUE, it is used to compute the number of lags for which autocovariance matrices are estimated
common.args	a list specifying the tuning parameters required for estimating the impulse response functions and common shocks. It contains: <ul style="list-style-type: none"> <li>factor.var.order order of the blockwise VAR representation of the common component. If factor.var.order = NULL, it is selected blockwise by Schwarz criterion</li> </ul>



- `max.var.order` maximum blockwise VAR order for the Schwarz criterion
- `trunc.lags` truncation lag for impulse response function estimation
- `n.perm` number of cross-sectional permutations involved in impulse response function estimation

## 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

- 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.
- Avarucci, M., Cavicchioli, M., Forni, M., & Zaffaroni, P. (2022) The main business cycle shock(s): Frequency-band estimation of the number of dynamic factors.
- 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) *fnets: An R Package for Network Estimation and Forecasting via Factor-Adjusted VAR Modelling*. *arXiv preprint arXiv:2301.11675*.

## See Also

[print.fm](#), [predict.fm](#)

## Examples

```
set.seed(1234)
n <- 500
p <- 50
common <- sim.restricted(n, p)
x <- common$data + matrix(rnorm(n * p), nrow = p)
out <- fnets.factor.model(x, fm.restricted = FALSE)
```

---

fnets.var

11-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, penalty = NULL, path.length
    = 10),
  do.threshold = FALSE,
  n.iter = NULL,
  tol = 0,
  n.cores = min(parallel::detectCores() - 1, 3)
)
```

## Arguments

x	input time series matrix, with each row representing a variable and each column containing the observations at a given time
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	l1-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>, positive integer number of folds</li> <li>• <code>penalty</code> if <code>tuning = "bic"</code>, penalty multiplier between 0 and 1; if <code>penalty = NULL</code>, it is set to <math>1/(1+\exp(\dim(x)[1]/\dim(x)[2]))</math> by default</li> <li>• <code>path.length</code> positive integer number of regularisation parameter values to consider; a sequence is generated automatically based in this value</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, by default depends on <code>var.order</code> ; 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>	l1-regularisation parameter
<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) fnets: An R Package for Network Estimation and Forecasting via Factor-Adjusted VAR Modelling. arXiv preprint arXiv:2301.11675.

Examples

```
library(fnets)

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

fv <- fnets.var(x,
  n.cores = 2
)
```

---

network	<i>Convert networks into igraph objects</i>
---------	---

---

Description

Convert networks into igraph objects

Usage

```
network(object, ...)
```

Arguments

object	object
...	additional arguments

See Also

[network.fnets](#)

---

network.fnets	<i>Convert networks estimated by fnets into igraph objects</i>
---------------	--

---

Description

Converts S3 objects of class `fnets` into a network. Produces an `igraph` object for the 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. When plotting the network, note that the edge weights may be negative since they correspond to the entries of the estimators of VAR parameters and (long-run) partial correlations.

**Usage**

```
## S3 method for class 'fnets'
network(
  object,
  type = c("granger", "pc", "lrpc"),
  names = NA,
  groups = NA,
  group.colours = NA,
  ...
)
```

**Arguments**

object	fnets object
type	a string specifying which of the above three networks (i)–(iii) to visualise; possible values are <ul style="list-style-type: none"> <li>• "granger" directed network representing Granger causal linkages</li> <li>• "pc" undirected network representing contemporaneous linkages; available when <code>object\$do.lrpc = TRUE</code></li> <li>• "lrpc" undirected network summarising Granger causal and contemporaneous linkages; available when <code>x\$do.lrpc = TRUE</code></li> </ul>
names	a character vector containing the names of the vertices
groups	an integer vector denoting any group structure of the vertices
group.colours	a vector denoting colours corresponding to groups
...	additional arguments to <code>igraph::graph_from_adjacency_matrix</code>

**Details**

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

**Value**

a list containing

network	igraph object
names	input argument
groups	input argument
grp.col	vector of colours corresponding to each node
...	additional arguments to <code>igraph::graph_from_adjacency_matrix</code>

**See Also**

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

## 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,
  do.threshold = TRUE,
  var.args = list(n.cores = 2)
)
net <- network(out, type = "granger")$network
plot(net, layout = igraph::layout_in_circle(net))
network(out, type = "pc")
network(out, type = "lrpc")
```

---

par.lrpc

*Parametric estimation of long-run partial correlations of factor-adjusted VAR processes*

---

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

## Arguments

object	fnets object
eta	l1-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>• <code>n.folds</code> positive integer number of folds</li> <li>• <code>path.length</code> positive integer number of regularisation parameter values to consider; a sequence is generated automatically based in this value</li> </ul>
<code>lrpc.adaptive</code>	whether to use the adaptive estimation procedure
<code>eta.adaptive</code>	l1-regularisation parameter for Step 1 of the adaptive estimation procedure; if <code>eta.adaptive = NULL</code> , the default choice is $2 * \sqrt{\log(\dim(x)[1])/\dim(x)[2]}$
<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>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>	l1-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. arXiv preprint arXiv:2201.06110.
- 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) fnets: An R Package for Network Estimation and Forecasting via Factor-Adjusted VAR Modelling. arXiv preprint arXiv:2301.11675.

### 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, do.lrpc = FALSE, var.args = list(n.cores = 2))
plrpc <- par.lrpc(out, n.cores = 2)
out$lrpc <- plrpc
out$do.lrpc <- TRUE
plot(out, type = "pc", display = "network")
plot(out, type = "lrpc", display = "heatmap")

```

---

plot.factor.number	<i>Plot factor number</i>
--------------------	---------------------------

---

## Description

Plots the eigenvalue ratio or information criteria from a `factor.number` object

## Usage

```

## S3 method for class 'factor.number'
plot(x, ...)

```

## Arguments

<code>x</code>	<code>factor.number</code> object
<code>...</code>	not used

## Value

NULL, printed to console

## See Also

[factor.number](#)

## Examples

```

library(fnets)

## Alessi, Barigozzi, and Capasso method for restricted models
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)
print(abc)
plot(abc)

```



```
## Eigenvalue ratio method
er <- factor.number(x, method = "er", fm.restricted = TRUE)
print(er)
plot(er)

## Hallin and Liška method for unrestricted models
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)
print(hl)
plot(hl)
```

plot.fnets

*Plotting the networks estimated by fnets***Description**

Plotting method for S3 objects of class `fnets`. When `display = "network"` or `"heatmap"`, it 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. Edge widths are determined by edge weights. When `display = "tuning"`, it produces up to two plots (when `do.larpc = TRUE`) visualising the outcome of CV or IC adopted for selecting the  $l_1$ -regularisation parameters and the VAR order.

**Usage**

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

**Arguments**

`x` `fnets` object

type	a string specifying which of the above three networks (i)–(iii) to visualise when <code>display = "network"</code> or <code>display = "heatmap"</code> ; possible values are <ul style="list-style-type: none"> <li>• "granger" directed network representing Granger causal linkages</li> <li>• "pc" undirected network representing contemporaneous linkages; available when <code>x\$do.lrpc = TRUE</code></li> <li>• "lrpc" undirected network summarising Granger causal and contemporaneous linkages; available when <code>x\$do.lrpc = TRUE</code></li> </ul>
display	a string specifying which plot to produce; possible values are <ul style="list-style-type: none"> <li>• "network" visualise the network as an <code>igraph</code> object, see <a href="#">plot.igraph</a></li> <li>• "heatmap" visualise the network as a heatmap, see <a href="#">imagePlot</a></li> <li>• "tuning" visualise the outcome from CV or IC (specified by <code>tuning.args\$tuning</code> of <a href="#">fnets</a>) for selecting l1-regularisation parameters and the VAR order</li> </ul>
names	a character vector containing the names of the network vertices
groups	an integer vector denoting any group structure of the network vertices
group.colours	a vector denoting colours corresponding to groups
...	additional arguments

### Details

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

### Value

A plot produced as per the input arguments

### See Also

[fnets](#)

### 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,
  do.threshold = TRUE,
  var.args = list(n.cores = 2)
)
plot(out, type = "granger", display = "network",
  groups = rep(c(1,2), p/2), group.colours = c("orange", "blue"))
plot(out, type = "lrpc", display = "heatmap")
plot(out, display = "tuning")
```

---

plot.threshold	<i>Plotting the thresholding procedure</i>
----------------	--

---

### Description

Plotting method for S3 objects of class threshold. Produces a plot visualising three diagnostics for the thresholding procedure, with threshold values  $t_k$  (x axis) against (i) Ratio\_k, the ratio of the number of non-zero to zero entries in the matrix, as the threshold varies (ii) Diff\_k, the first difference of Ratio\_k (iii) ICUSUM\_kl, the absolute scaled cumulative sums of Diff\_k

### Usage

```
## S3 method for class 'threshold'
plot(x, plots = c(TRUE, FALSE, TRUE), ...)
```

### Arguments

x	threshold object
plots	logical vector, which plots to use (Ratio, Diff, CUSUM respectively)
...	additional arguments

### Details

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

### Value

A network plot produced as per the input arguments

### 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) fnets: An R Package for Network Estimation and Forecasting via Factor-Adjusted VAR Modelling. arXiv preprint arXiv:2301.11675.

### See Also

[threshold](#)

### Examples

```
library(fnets)

set.seed(123)
n <- 500
p <- 20
common <- sim.unrestricted(n, p)
```

```

idio <- sim.var(n, p)
x <- common$data + idio$data
out <- fnets(x,
  var.args = list(n.cores = 2)
)
# Granger-causal network
th1 <- threshold(out$idio.var$beta)
plot(th1)
print(th1)
# Partial correlations
th2 <- threshold(out$lrpc$pc)
# Long-run partial correlations
th3 <- threshold(out$lrpc$lrpc)

```

---

predict.fm

*Forecasting for factor models*


---

## Description

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

## Usage

```

## S3 method for class 'fm'
predict(object, n.ahead = 1, fc.restricted = TRUE, r = c("ic", "er"), ...)

```

## Arguments

object	fm object
n.ahead	forecasting horizon
fc.restricted	if fc.restricted = TRUE, the forecast is generated under a restricted factor model
r	number of static factors, or a string specifying the factor number selection method when fc.restricted = TRUE; possible values are: <ul style="list-style-type: none"> <li>"ic" information criteria of Alessi, Barigozzi &amp; Capasso (2010)</li> <li>"er" eigenvalue ratio of Ahn &amp; Horenstein (2013)</li> </ul>
...	not used

## Value

a list containing

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

**See Also**[fnets.factor.model](#)**Examples**

```

set.seed(123)
n <- 500
p <- 50
common <- sim.restricted(n, p)
x <- common$data + rnorm(n*p)
out <- fnets.factor.model(x, fm.restricted = TRUE)
pre <- predict(out)

```

predict.fnets

*Forecasting by fnets***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,
  newdata = NULL,
  n.ahead = 1,
  fc.restricted = TRUE,
  r = c("ic", "er"),
  ...
)

```

**Arguments**

object	fnets object
newdata	input time series matrix, with each row representing a variable; by default, uses input to object. Valid only for the case where newdata is modelled as a VAR process without any factors
n.ahead	forecasting horizon
fc.restricted	whether to forecast using a restricted or unrestricted, blockwise VAR representation of the common component
r	number of static factors, or a string specifying the factor number selection method when fc.restricted = TRUE; possible values are: <ul style="list-style-type: none"> <li>• "ic" information criteria of Alessi, Barigozzi &amp; Capasso (2010)</li> <li>• "er" eigenvalue ratio of Ahn &amp; Horenstein (2013)</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

**See Also**

[fnets](#)

**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,
do.lrpc = FALSE, var.args = list(n.cores = 2))
pre.unr <- predict(out, fc.restricted = FALSE)
pre.res <- predict(out, fc.restricted = TRUE)
```

---

print.factor.number	<i>Print factor number</i>
---------------------	----------------------------

---

**Description**

Prints a summary of a factor.number object

**Usage**

```
## S3 method for class 'factor.number'
print(x, ...)
```

**Arguments**

x	factor.number object
...	not used

**Value**

NULL, printed to console

**See Also**[factor.number](#)**Examples**

```
library(fnets)

## Alessi, Barigozzi, and Capasso method for restricted models
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)
print(abc)
plot(abc)

## Eigenvalue ratio method
er <- factor.number(x, method = "er", fm.restricted = TRUE)
print(er)
plot(er)

## Hallin and Liška method for unrestricted models
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)
print(hl)
plot(hl)
```

---

`print.fm`*Print factor model*

---

**Description**

Prints a summary of a fm object

**Usage**

```
## S3 method for class 'fm'
print(x, ...)
```

**Arguments**

x	fm object
...	not used

**Value**

NULL, printed to console

**See Also**

[fnets.factor.model](#)

**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.factor.model(x, q = "ic")
print(out)
```

---

print.fnets

*Print fnets*

---

**Description**

Prints a summary of a fnets object

**Usage**

```
## S3 method for class 'fnets'
print(x, ...)
```

**Arguments**

x	fnets object
...	not used

**Value**

NULL, printed to console

**See Also**

[fnets](#)



## 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,
do.lrpc = FALSE, var.args = list(n.cores = 2))
print(out)
x <- idio$data
out <- fnets.var(x,
n.cores = 2)
print(out)
```

---

print.threshold	<i>Print threshold</i>
-----------------	------------------------

---

## Description

Prints a summary of a threshold object

## Usage

```
## S3 method for class 'threshold'
print(x, ...)
```

## Arguments

x	threshold object
...	not used

## Value

NULL, printed to console

## 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) fnets: An R Package for Network Estimation and Forecasting via Factor-Adjusted VAR Modelling

## See Also

[threshold](#)

**Examples**

```

library(fnets)

set.seed(123)
n <- 500
p <- 20
common <- sim.unrestricted(n, p)
idio <- sim.var(n, p)
x <- common$data + idio$data
out <- fnets(x,
  var.args = list(n.cores = 2)
)
# Granger-causal network
th1 <- threshold(out$idio.var$beta)
plot(th1)
print(th1)
# Partial correlations
th2 <- threshold(out$lrpc$pc)
# Long-run partial correlations
th3 <- threshold(out$lrpc$lrpc)

```

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	time series matrix with n rows and p columns
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) fnets: An R Package for Network Estimation and Forecasting via Factor-Adjusted VAR Modelling

## 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	time series matrix with n rows and p columns
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) fnets: An R Package for Network Estimation and Forecasting via Factor-Adjusted VAR Modelling. arXiv preprint arXiv:2301.11675.

## 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 <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	time series matrix with n rows and p columns
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) fnets: An R Package for Network Estimation and Forecasting via Factor-Adjusted VAR Modelling

## Examples

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

threshold

*Threshold the entries of the input matrix at a data-driven level***Description**

Threshold the entries of the input matrix at a data-driven level. This can be used to perform edge selection for VAR parameter, inverse innovation covariance, and long-run partial correlation networks.

**Usage**

```
threshold(mat, path.length = 500)
```

**Arguments**

mat	input parameter matrix
path.length	number of candidate thresholds

**Details**

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

**Value**

an S3 object of class threshold, 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) fnets: An R Package for Network Estimation and Forecasting via Factor-Adjusted VAR Modelling. arXiv preprint arXiv:2301.11675.

**See Also**

[plot.threshold](#), [print.threshold](#)

**Examples**

```
library(fnets)

set.seed(123)
n <- 500
p <- 20
common <- sim.unrestricted(n, p)
idio <- sim.var(n, p)
x <- common$data + idio$data
out <- fnets(x,
  var.args = list(n.cores = 2)
)
# Granger-causal network
th1 <- threshold(out$idio.var$beta)
plot(th1)
print(th1)
# Partial correlations
th2 <- threshold(out$lrpc$pc)
# Long-run partial correlations
th3 <- threshold(out$lrpc$lrpc)
```

# Index

abc.factor.number, 8

factor.number, 2, 5, 8, 16, 23  
fnets, 4, 13, 18, 22, 24  
fnets.factor.model, 8, 21, 24  
fnets.var, 10

hl.factor.number, 8

imagePlot, 18

makePSOCKcluster, 5, 11, 15

network, 12  
network.fnets, 12, 12

par.lrpc, 6, 14  
plot.factor.number, 3, 16  
plot.fnets, 7, 13, 17  
plot.igraph, 18  
plot.threshold, 19, 29  
predict.fm, 9, 20  
predict.fnets, 7, 21  
print.factor.number, 3, 22  
print.fm, 9, 23  
print.fnets, 7, 24  
print.threshold, 25, 29

sim.restricted, 26  
sim.unrestricted, 27  
sim.var, 28

threshold, 6, 11, 15, 19, 25, 29