Package: factorselect (via r-universe)

October 20, 2026

Title Eigenvalue-Based Estimation of the Number of Factors in Approximate Factor Models

Version 0.1.2

Description Eigenvalue-based estimation of the number of factors in approximate factor models. Designed to work when either N or T is large, without requiring both dimensions to grow simultaneously. Implements the eigenvalue ratio estimator of Ahn and Horenstein (2013) <doi:10.3982/ECTA8968>, the information criteria of Bai and Ng (2002) <doi:10.1111/1468-0262.00273>, the tuned penalty of Alessi, Barigozzi and Capasso (2010) <doi:10.1016/j.spl.2010.08.005>, the auto-covariance ratio estimator of Lam and Yao (2012) <doi:10.1214/12-AOS970>, and the edge distribution estimators of Onatski (2009) <doi:10.3982/ECTA6964> and Onatski (2010) <doi:10.1162/REST_a_00043>.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Suggests RSpectra, testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

URL https://github.com/penny4nonsense/factorselect

BugReports https://github.com/penny4nonsense/factorselect/issues

Repository https://penny4nonsense.r-universe.dev

Date/Publication 2026-04-22 16:45:48 UTC

RemoteUrl https://github.com/penny4nonsense/factorselect

RemoteRef HEAD

RemoteSha f0f08d5953488bf2c0db0aa8b341b16e5f28dc7c

Contents

.abc . . . 2
.ahn_horenstein . . . 4
.bai_ng . . . 5
.extract_eigenvalues . . . 7
.lam_yao . . . 8
.onatski_2009 . . . 9
.onatski_2010 . . . 10
.prepare_matrix . . . 11
plot.factor_select . . . 13
print.factor_select . . . 13
select_factors . . . 14
simulate_factor_model . . . 16
summary.factor_select . . . 17

Index **18**

`.abc` *Alessi, Barigozzi and Capasso (2010) Tuned Information Criteria*

Description

Estimates the number of factors using the tuning-stability procedure of Alessi, Barigozzi and Capasso (2010) applied to the three IC penalty functions of Bai and Ng (2002). For each penalty function, a grid of tuning constants is used and the most stable estimate across the grid is selected as the final estimate.

Usage

```
.abc(eigenvalues, V0, kmax, N, TT, c_grid = seq(0, 1, by = 0.01))
```

Arguments

`eigenvalues`	Numeric vector of eigenvalues in descending order of length kmax + 1, typically obtained from `.extract_eigenvalues`.
`V0`	Numeric scalar. Total mean squared value of the panel, `sum(X^2) / (N * TT)`, computed from unstandardized demeaned data.
`kmax`	Integer. Maximum number of factors to consider.
`N`	Integer. Number of cross-sectional units.
`TT`	Integer. Number of time periods.
`c_grid`	Numeric vector. Grid of tuning constants over which to evaluate stability. Defaults to `seq(0, 1, by = 0.01)`.

Details

The ABC estimator applies the tuning-stability procedure of Hallin and Liska (2007) to the IC criteria of Bai and Ng (2002). For each tuning constant c in the grid, a modified criterion is minimized:

$$IC_j(k, c) = \ln(V(k)) + k \cdot c \cdot g_j(N, T)$$

where g_j is the penalty function from IC_{pj} of Bai and Ng (2002), for j = 1, 2, 3. The final estimate is the modal value of $\hat{k}(c)$ across the grid — the value of k that is selected most frequently as c varies.

As with `.bai_ng`, this estimator requires unstandardized data. The argument `V0` should be computed from demeaned but unstandardized data.

The ABC estimator generally outperforms the raw Bai & Ng IC criteria in finite samples, particularly when errors are cross-sectionally correlated.

Value

A named list with the following elements:

k_abc1 Integer. Selected number of factors using ABC with IC1 penalty.

k_abc2 Integer. Selected number of factors using ABC with IC2 penalty.

k_abc3 Integer. Selected number of factors using ABC with IC3 penalty.

k_grid_abc1 Integer vector of length `length(c_grid)`. Selected k for each value of c using IC1 penalty.

k_grid_abc2 Integer vector of length `length(c_grid)`. Selected k for each value of c using IC2 penalty.

k_grid_abc3 Integer vector of length `length(c_grid)`. Selected k for each value of c using IC3 penalty.

c_grid Numeric vector. The tuning constant grid used.

References

Alessi, L., Barigozzi, M. and Capasso, M. (2010). Improved Penalization for Determining the Number of Factors in Approximate Factor Models. *Statistics and Probability Letters*, 80, 1806-1813.

Bai, J. and Ng, S. (2002). Determining the Number of Factors in Approximate Factor Models. *Econometrica*, 70(1), 191-221.

Hallin, M. and Liska, R. (2007). Determining the Number of Factors in the Generalized Dynamic Factor Model. *Journal of the American Statistical Association*, 102, 603-617.

See Also

`.bai_ng`, `.extract_eigenvalues`, `select_factors`

`.ahn_horenstein` *Ahn-Horenstein Eigenvalue Ratio Estimator*

Description

Estimates the number of factors using the eigenvalue ratio (ER) and growth ratio (GR) statistics of Ahn and Horenstein (2013). The ratio approach provides robustness to perturbations in the eigenvalue spectrum and performs well when only one dimension (N or T) is large.

Usage

```
.ahn_horenstein(eigenvalues, kmax, n)
```

Arguments

`eigenvalues`	Numeric vector of eigenvalues in descending order of length kmax + 1, typically obtained from `.extract_eigenvalues`. Must be positive.
`kmax`	Integer. Maximum number of factors to consider. The function evaluates the ratio statistics for k = 1, ..., kmax.
`n`	Integer. The value of min(N, T), used to compute the mock eigenvalue boundary term following Ahn and Horenstein (2013) Corollary 1.

Details

The ER statistic is defined as the ratio of successive eigenvalue differences:

$$ER(k) = \delta_k / \delta_{k+1}$$

where δ_k is the k-th successive difference in the eigenvalue sequence. The GR statistic replaces raw differences with log growth rates:

$$GR(k) = \log(1 + \delta_k/\lambda_k) / \log(1 + \delta_{k+1}/\lambda_{k+1})$$

The boundary case k = 0 is handled by assigning $\lambda_1 / \log(n)$ as the initial difference term, following Ahn and Horenstein (2013).

The number of factors is selected as the argmax of each statistic over k = 1, ..., kmax.

Value

A named list with the following elements:

k_er Integer. Selected number of factors based on the ER statistic.

k_gr Integer. Selected number of factors based on the GR statistic.

er Numeric vector of length kmax. Full ER statistic sequence.

gr Numeric vector of length kmax. Full GR statistic sequence.

References

Ahn, S.C. and Horenstein, A.R. (2013). Eigenvalue Ratio Test for the Number of Factors. *Econometrica*, 81(3), 1203-1227.

See Also

`.extract_eigenvalues`, `select_factors`

`.bai_ng` *Bai and Ng (2002) Information Criteria for Number of Factors*

Description

Estimates the number of factors using the six penalty-based criteria of Bai and Ng (2002). Includes three PC criteria (minimize penalized residual variance) and three IC criteria (minimize penalized log residual variance).

Usage

```
.bai_ng(eigenvalues, V0, kmax, N, TT)
```

Arguments

`eigenvalues`	Numeric vector of eigenvalues in descending order of length kmax + 1, typically obtained from `.extract_eigenvalues`.
`V0`	Numeric scalar. Total mean squared value of the panel, `sum(X^2) / (N * TT)`, computed from unstandardized demeaned data.
`kmax`	Integer. Maximum number of factors to consider.
`N`	Integer. Number of cross-sectional units.
`TT`	Integer. Number of time periods.

Details

The six criteria are defined as follows. Let $V(k)$ denote the residual variance from a k-factor model, $m = \min(N, T)$, and $\hat{\sigma}^2 = V(k_{max})$.

PC criteria (minimize penalized residual variance):

$$PC_{p1}(k) = V(k) + k\hat{\sigma}^2 \frac{N+T}{NT} \ln\left(\frac{NT}{N+T}\right)$$

$$PC_{p2}(k) = V(k) + k\hat{\sigma}^2 \frac{N+T}{NT} \ln(m)$$

$$PC_{p3}(k) = V(k) + k\hat{\sigma}^2 \frac{\ln(m)}{m}$$

IC criteria (minimize penalized log residual variance):

$$IC_{p1}(k) = \ln(V(k)) + k\frac{N+T}{NT}\ln\left(\frac{NT}{N+T}\right)$$

$$IC_{p2}(k) = \ln(V(k)) + k\frac{N+T}{NT}\ln(m)$$

$$IC_{p3}(k) = \ln(V(k)) + k\frac{\ln(m)}{m}$$

$V(k)$ is computed from the eigenvalues of $XX'/(NT)$ as:

$$V(k) = \frac{1}{NT}\sum_{j=k+1}^{m}\lambda_j$$

which is the mean residual variance after removing the first k factors.

All six criteria are minimized over $k = 0, 1, \ldots, k_{max}$. Note that $k = 0$ is included to allow for the possibility of no factors.

These estimators require both N and T to be large for consistent estimation. They may perform poorly when either dimension is small. For more robust estimation, consider `.ahn_horenstein`.

Value

A named list with the following elements:

k_pc1 Integer. Selected number of factors by PC_p1.

k_pc2 Integer. Selected number of factors by PC_p2.

k_pc3 Integer. Selected number of factors by PC_p3.

k_ic1 Integer. Selected number of factors by IC_p1.

k_ic2 Integer. Selected number of factors by IC_p2.

k_ic3 Integer. Selected number of factors by IC_p3.

pc1 Numeric vector of length kmax. Full PC_p1 criterion sequence.

pc2 Numeric vector of length kmax. Full PC_p2 criterion sequence.

pc3 Numeric vector of length kmax. Full PC_p3 criterion sequence.

ic1 Numeric vector of length kmax. Full IC_p1 criterion sequence.

ic2 Numeric vector of length kmax. Full IC_p2 criterion sequence.

ic3 Numeric vector of length kmax. Full IC_p3 criterion sequence.

References

Bai, J. and Ng, S. (2002). Determining the Number of Factors in Approximate Factor Models. *Econometrica*, 70(1), 191-221.

See Also

`.ahn_horenstein`, `.extract_eigenvalues`, `select_factors`

`.extract_eigenvalues` *Extract Leading Eigenvalues from a Panel Data Matrix*

Description

Computes the leading eigenvalues of the sample covariance matrix using a truncated eigendecomposition. Automatically selects the smaller of the N x N or T x T covariance matrix for efficiency. Uses RSpectra when available for large matrices, falling back to base R otherwise.

Usage

```
.extract_eigenvalues(X, kmax)
```

Arguments

`X`	Numeric matrix of dimensions T x N, typically preprocessed by `.prepare_matrix`.
`kmax`	Integer. Number of leading eigenvalues to compute. Should be set generously (e.g., 8-15) to allow estimators to evaluate the full candidate range.

Details

When N <= T, decomposes the N x N matrix XX'/T. When N > T, decomposes the T x T matrix $X'X/N$. This ensures the cheaper decomposition is always used.

RSpectra's `eigs_sym()` is used when available and when `min(N, T)` > `100`, as the truncated decomposition only provides meaningful speedup at larger scales.

Value

A named list with the following elements:

values Numeric vector of length `kmax + 1` containing the leading eigenvalues in descending order. The extra eigenvalue is required by ratio-based estimators.

vectors Numeric matrix of corresponding eigenvectors.

orientation Character string, either `"N"` or `"T"`, indicating which covariance matrix was decomposed.

References

Ahn, S.C. and Horenstein, A.R. (2013). Eigenvalue Ratio Test for the Number of Factors. *Econometrica*, 81(3), 1203-1227.

See Also

`.prepare_matrix`, `.ahn_horenstein`

`.lam_yao` *Lam and Yao (2012) Eigenvalue Ratio Estimator*

Description

Estimates the number of factors using the eigenvalue ratio estimator of Lam and Yao (2012). Unlike estimators based on the contemporaneous covariance matrix, this estimator uses lagged auto-covariance matrices, exploiting the fact that the factor loading space is spanned by the eigenvectors of the summed lagged auto-covariance matrix M corresponding to its nonzero eigenvalues.

Usage

```
.lam_yao(X, kmax, h = 1)
```

Arguments

`X`	Numeric matrix of dimensions T x N, typically preprocessed by `.prepare_matrix`.
`kmax`	Integer. Maximum number of factors to consider.
`h`	Integer. Number of lags to use in constructing the auto-covariance matrix M. Defaults to `1`. The paper suggests small values are sufficient; increasing h may improve performance when factors have strong serial correlation.

Details

The estimator constructs the N x N matrix:

$$M = \sum_{k=1}^{h} \hat{\Sigma}_k \hat{\Sigma}_k'$$

where $\hat{\Sigma}_k = T^{-1} \sum_{t=k+1}^{T} x_t x_{t-k}'$ is the lag-k sample auto-covariance matrix.

The factor loading space is spanned by the eigenvectors of M corresponding to its nonzero eigenvalues, and the number of nonzero eigenvalues equals the number of factors r (Lam and Yao, 2012, Proposition 1). In finite samples, the ratio of adjacent eigenvalues of M spikes at r because eigenvalues r+1 onward are theoretically zero.

The number of factors is estimated as:

$$\hat{r} = \arg \max_{1 \le k \le k_{max}} \frac{\lambda_k(M)}{\lambda_{k+1}(M)}$$

Value

A named list with the following elements:

k Integer. Selected number of factors.

ratios Numeric vector of length kmax. Full eigenvalue ratio sequence of M.

eigenvalues Numeric vector of length kmax + 1. Leading eigenvalues of M in descending order.

References

Lam, C. and Yao, Q. (2012). Factor Modelling for High-Dimensional Time Series: Inference for the Number of Factors. *The Annals of Statistics*, 40(2), 694-726.

See Also

`.ahn_horenstein`, `select_factors`

`.onatski_2009` *Onatski (2009) Test for the Number of Factors*

Description

Estimates the number of factors using the sequential hypothesis testing procedure of Onatski (2009), applied to the static approximate factor model version described in Section 4 of that paper. The test statistic is based on ratios of differences of adjacent eigenvalues of a complex-valued transformation of the data.

Usage

```
.onatski_2009(X, kmax, alpha = 0.05)
```

Arguments

`X`	Numeric matrix of dimensions T x N, typically preprocessed by `.prepare_matrix`. Must have an even number of rows.
`kmax`	Integer. Maximum number of factors to consider. Defines the upper bound k1 in the sequential testing procedure.
`alpha`	Numeric. Significance level for the sequential test. Defaults to `0.05`. Must be one of `0.01`, `0.02`, `0.03`, `0.04`, `0.05`, `0.06`, `0.07`, `0.08`, `0.09`, `0.10`, `0.15`.

Details

The static approximate factor model version of the Onatski (2009) test (Section 4) proceeds as follows:

1. Split the T x N data matrix into two halves of length T/2.
2. Form complex-valued vectors $\tilde{X}_j = X_j + iX_{j+T/2}$ for $j = 1, \ldots, T/2$.
3. Compute eigenvalues $\tilde{\gamma}_i$ of $\frac{2}{T}\sum_{j=1}^{T/2} \tilde{X}_j \tilde{X}_j^*$.
4. Sequentially test $H_0 : r = k_0$ versus $H_1 : k_0 < r \leq k_{max}$ for $k_0 = 0, 1, \ldots$ using the statistic $\tilde{R} = \max_{k_0 < i \leq k_{max}} (\tilde{\gamma}_i - \tilde{\gamma}_{i+1})/(\tilde{\gamma}_{i+1} - \tilde{\gamma}_{i+2})$.
5. Stop when H_0 is not rejected. The estimate is the current k_0.

Critical values are taken from Table I of Onatski (2009) and depend on the significance level `alpha` and the number of factors tested under the alternative $k_1 - k_0 = k_{max} - k_0$.

If T is odd, the last observation is dropped to ensure equal-length halves.

Value

A named list with the following elements:

k Integer. Estimated number of factors from the sequential testing procedure.

ratios Numeric vector of length kmax. The ratio statistic $(\tilde{\gamma}_i - \tilde{\gamma}_{i+1})/(\tilde{\gamma}_{i+1} - \tilde{\gamma}_{i+2})$ for each i.

eigenvalues Numeric vector of length kmax + 2. Leading eigenvalues of the complex covariance matrix in descending order.

critical_value Numeric. Critical value used for the test at the specified significance level.

alpha Numeric. The significance level used.

References

Onatski, A. (2009). Testing Hypotheses About the Number of Factors in Large Factor Models. *Econometrica*, 77(5), 1447-1479.

See Also

`.ahn_horenstein`, `select_factors`

.onatski_2010 *Onatski (2010) Edge Distribution Estimator*

Description

Estimates the number of factors using the Edge Distribution (ED) estimator of Onatski (2010). The estimator exploits the fact that idiosyncratic eigenvalues of the sample covariance matrix cluster around a single point, while systematic eigenvalues diverge to infinity. The threshold separating the two groups is estimated iteratively using the square root shape of the edge of the eigenvalue distribution.

Usage

```
.onatski_2010(eigenvalues, kmax, n_iter = 4L)
```

Arguments

`eigenvalues`	Numeric vector of eigenvalues in descending order, typically obtained from `.extract_eigenvalues`. Must contain at least `kmax + 5` elements to allow the OLS regression in the calibration step.
`kmax`	Integer. Maximum number of factors to consider.
`n_iter`	Integer. Maximum number of iterations for the calibration procedure. Defaults to 4 as recommended by Onatski (2010).

Details

The ED estimator of Onatski (2010) is based on the theoretical result that idiosyncratic eigenvalues cluster around the upper edge $u(\mathcal{F}^{c,A,B})$ of the limiting spectral distribution, while systematic eigenvalues diverge. Near the edge, the density of the limiting spectral distribution behaves like a square root function, implying that eigenvalue differences $\lambda_i - \lambda_{i+1}$ for idiosyncratic eigenvalues behave approximately as $(an)^{-2/3}$.

The calibration procedure estimates $\hat{\beta} = (an)^{-2/3}$ by regressing five consecutive eigenvalues $\lambda_j, \ldots, \lambda_{j+4}$ on a constant and $(j-1)^{2/3}, \ldots, (j+3)^{2/3}$, where j is initialized at $r_{max} + 1$ and updated iteratively.

The estimator requires `eigenvalues` to contain at least `kmax + 5` elements so that the OLS window $j, \ldots, j+4$ is always available.

Value

A named list with the following elements:

k Integer. Estimated number of factors.

delta Numeric. The estimated threshold $\delta = 2|\hat{\beta}|$.

beta Numeric. The estimated slope coefficient $\hat{\beta}$ from the OLS regression in the final iteration.

differences Numeric vector of length kmax. Successive eigenvalue differences $\lambda_i - \lambda_{i+1}$.

n_iter Integer. Number of iterations performed.

References

Onatski, A. (2010). Determining the Number of Factors From Empirical Distribution of Eigenvalues. *The Review of Economics and Statistics*, 92(4), 1004-1016.

See Also

`.extract_eigenvalues`, `select_factors`

.prepare_matrix *Demean and Scale a Matrix for Factor Analysis*

Description

Removes individual means, time means, or both from a numeric matrix, and optionally scales to unit variance. This is the standard preprocessing step required before eigendecomposition in factor number estimation.

Usage

```
.prepare_matrix(
  X,
  demean = c("both", "individual", "time", "none"),
  standardize = TRUE
)
```

Arguments

X	Numeric matrix of dimensions T x N (time periods x units).
demean	Character string specifying the demeaning method. One of:
	"both" Remove both individual (column) and time (row) means. This is the recommended default for macro panels.
	"individual" Remove individual (column) means only.
	"time" Remove time (row) means only.
	"none" No demeaning applied.
standardize	Logical. If TRUE (default), scale each column to unit variance after demeaning.

Details

When demean = "both", the function iterates individual and time demeaning to convergence (two passes is sufficient for practical purposes). This follows the within-transformation used in panel data models.

Value

A demeaned (and optionally scaled) numeric matrix of the same dimensions as X.

References

Bai, J. and Ng, S. (2002). Determining the Number of Factors in Approximate Factor Models. *Econometrica*, 70(1), 191-221.

See Also

.extract_eigenvalues, select_factors

Examples

```
## Not run:
set.seed(42)
X <- matrix(rnorm(200 * 100, mean = 5), 200, 100)
X_clean <- .prepare_matrix(X, demean = "both", standardize = TRUE)

## End(Not run)
```

plot.factor_select *Plot Method for factor_select Objects*

Description

Produces a scree plot of the leading eigenvalues with the selected number of factors marked.

Usage

```
## S3 method for class 'factor_select'
plot(x, main = "Scree Plot", ...)
```

Arguments

x	A factor_select object.
main	Character string. Plot title. Defaults to "Scree Plot".
...	Further arguments passed to plot().

Value

Invisibly returns x, called for its side effect of producing a scree plot.

print.factor_select *Print Method for factor_select Objects*

Description

Print Method for factor_select Objects

Usage

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

Arguments

x	A factor_select object.
...	Further arguments passed to or from other methods.

Value

Invisibly returns x, called for its side effect of printing a summary of the factor selection results to the console.

`select_factors` *Select the Number of Factors in an Approximate Factor Model*

Description

A unified interface for estimating the number of factors in a large dimensional approximate factor model. Preprocesses the data and dispatches to one or more factor number estimators.

Usage

```
select_factors(
  X,
  method = "ahn_horenstein",
  kmax = NULL,
  demean = c("both", "individual", "time", "none"),
  standardize = TRUE,
  h = 1L,
  alpha = 0.05
)
```

Arguments

`X`	A numeric matrix of dimensions T x N (time periods x units), or an object coercible to a numeric matrix. Must be a balanced panel with no missing values.
`method`	Character vector specifying which estimator(s) to use. One or more of `"ahn_horenstein"`, `"bai_ng"`, `"onatski_2009"`, `"onatski_2010"`, `"abc"`, `"lam_yao"`. Defaults to `"ahn_horenstein"`.
`kmax`	Integer. Maximum number of factors to consider. Defaults to `NULL`, in which case it is set to `min(floor(sqrt(min(N, T))), 8)`.
`demean`	Character string passed to `.prepare_matrix()`. One of `"both"`, `"individual"`, `"time"`, `"none"`. Defaults to `"both"` as recommended by Ahn and Horenstein (2013).
`standardize`	Logical. Whether to standardize columns to unit variance before estimation. Defaults to `TRUE`. Note that `bai_ng`, `abc`, and `lam_yao` always use unstandardized data regardless of this setting.
`h`	Integer. Number of lags to use for the `lam_yao` estimator. Defaults to `1`. Ignored for all other methods.
`alpha`	Numeric. Significance level for the `onatski_2009` sequential test. Defaults to `0.05`. Must be one of `0.01`, `0.02`, `0.03`, `0.04`, `0.05`, `0.06`, `0.07`, `0.08`, `0.09`, `0.10`, `0.15`. Ignored for all other methods.

Details

The data are first preprocessed via `.prepare_matrix()` and then a single eigendecomposition is performed via `.extract_eigenvalues()`, which is shared across all requested estimators for efficiency.

The default method is `"ahn_horenstein"`, which is recommended for most applications. It is robust to perturbations in the eigenvalue spectrum and performs well when only one of N or T is large.

The `"bai_ng"`, `"abc"`, and `"lam_yao"` methods always use unstandardized data because their penalty terms and auto-covariance structure depend on the actual scale of the data.

Value

An object of class `"factor_select"`, which is a named list with the following elements:

k Named integer vector of selected factor numbers, one per method requested.

method Character vector of methods used.

kmax Integer. Maximum number of factors considered.

eigenvalues Numeric vector of leading eigenvalues.

details Named list of full output from each estimator.

call The matched call.

References

Ahn, S.C. and Horenstein, A.R. (2013). Eigenvalue Ratio Test for the Number of Factors. *Econometrica*, 81(3), 1203-1227.

Bai, J. and Ng, S. (2002). Determining the Number of Factors in Approximate Factor Models. *Econometrica*, 70(1), 191-221.

Alessi, L., Barigozzi, M. and Capasso, M. (2010). Improved Penalization for Determining the Number of Factors in Approximate Factor Models. *Statistics and Probability Letters*, 80, 1806-1813.

Lam, C. and Yao, Q. (2012). Factor Modelling for High-Dimensional Time Series: Inference for the Number of Factors. *The Annals of Statistics*, 40(2), 694-726.

See Also

`.ahn_horenstein`, `.bai_ng`, `.abc`, `.lam_yao`, `.prepare_matrix`, `.extract_eigenvalues`

Examples

```
set.seed(42)
N <- 100; T <- 200; k_true <- 3
Lambda <- matrix(rnorm(N * k_true), N, k_true)
F_mat  <- matrix(rnorm(T * k_true), T, k_true)
E      <- matrix(rnorm(T * N, sd = 0.5), T, N)
X      <- F_mat %*% t(Lambda) + E
select_factors(X)
```

`simulate_factor_model` *Simulate Data from an Approximate Factor Model*

Description

Generates a simulated panel data matrix from a static approximate factor model. Useful for testing and benchmarking factor number estimators.

Usage

```
simulate_factor_model(N, TT, k, sd = 1, seed = NULL)
```

Arguments

`N`	Integer. Number of cross-sectional units.
`TT`	Integer. Number of time periods. Named `TT` to avoid conflict with the base R function `T` (which evaluates to `TRUE`).
`k`	Integer. True number of factors.
`sd`	Numeric. Standard deviation of the idiosyncratic error term. Defaults to `1`. Lower values produce stronger signal relative to noise.
`seed`	Integer or `NULL`. Random seed for reproducibility. Defaults to `NULL` (no seed set).

Details

The data generating process follows the standard approximate factor model of Chamberlain and Rothschild (1983) as used in the simulation exercises of Ahn and Horenstein (2013). Factors and loadings are independent standard normal draws. Errors are i.i.d. normal with mean zero and standard deviation `sd`.

The signal-to-noise ratio is controlled by `sd` — smaller values produce a cleaner factor structure that is easier for estimators to recover. The default `sd = 1` matches the baseline simulation design of Ahn and Horenstein (2013) with `theta = 1`.

Value

A numeric matrix of dimensions `TT x N` generated from:

$$X = F\Lambda' + E$$

where F is a `TT x k` matrix of factors drawn from $N(0,1)$, Λ is an `N x k` matrix of loadings drawn from $N(0,1)$, and E is a `TT x N` matrix of idiosyncratic errors drawn from $N(0, sd^2)$.

References

Ahn, S.C. and Horenstein, A.R. (2013). Eigenvalue Ratio Test for the Number of Factors. *Econometrica*, 81(3), 1203-1227.

Chamberlain, G. and Rothschild, M. (1983). Arbitrage, Factor Structure, and Mean-Variance Analysis on Large Asset Markets. *Econometrica*, 51(5), 1281-1304.

See Also

select_factors

Examples

```
# Simulate a factor model with 3 factors
X <- simulate_factor_model(N = 100, TT = 200, k = 3, sd = 0.5, seed = 42)
dim(X)

# Pass directly to select_factors
result <- select_factors(X)
result$k
```

---

summary.factor_select *Summary Method for factor_select Objects*

Description

Summary Method for factor_select Objects

Usage

```
## S3 method for class 'factor_select'
summary(object, ...)
```

Arguments

object	A factor_select object.
...	Further arguments passed to or from other methods.

Value

Invisibly returns object, called for its side effect of printing a summary including leading eigenvalues to the console.

Index

.abc, 2, *15*
.ahn_horenstein, 4, *6, 7*, *9, 10*, *15*
.bai_ng, *3*, 5, *15*
.extract_eigenvalues, *2–6*, 7, *10–12*, *15*
.lam_yao, 8, *15*
.onatski_2009, 9
.onatski_2010, 10
.prepare_matrix, *7–9*, 11, *15*

plot.factor_select, 13
print.factor_select, 13

select_factors, *3*, *5, 6*, *9–12*, 14, *17*
simulate_factor_model, 16
summary.factor_select, 17