Package: pvarife (via r-universe)

June 12, 2026

Type Package

Version 0.1.2

Title Panel VAR Models with Interactive Fixed Effects

Description Implements the estimator of Tugan (2021)
<doi:10.1093/ectj/utaa021> for panel vector autoregression
(VAR) models with interactive fixed effects. Provides joint
estimation of VAR coefficients, latent common factors, and
factor loadings via an iterative algorithm that alternates
between principal component estimation of the factors and least
squares estimation of the VAR coefficients, following the
approach of Bai (2009) <doi:10.3982/ECTA6135>. Supports impulse
response functions under recursive (Cholesky) identification,
parametric confidence bands from the joint asymptotic
distribution of the estimator (Theorem 2.3), and a classical
residual bootstrap for robustness checks.

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 4.1.0)

Imports stats, mvtnorm, ggplot2, rlang

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

VignetteBuilder knitr

URL `https://github.com/Rickchen0910/pvarife`

BugReports `https://github.com/Rickchen0910/pvarife/issues`

RoxygenNote 7.3.3

Repository https://rickchen0910.r-universe.dev

Date/Publication 2026-06-11 18:56:07 UTC

RemoteUrl https://github.com/rickchen0910/pvarife

RemoteRef HEAD

RemoteSha 2f39492ec24f9b871c7dcdcef8b46c8cce165c52

Contents

`asymptotic_var` *Asymptotic variance and bias of the pvarife estimator*

Description

Computes the asymptotic bias and variance-covariance matrix of $\hat{\beta}$ under Theorem 2.3 of Tugan (2021). These quantities are used by `irf_bands` to construct parametric confidence bands.

Usage

```
asymptotic_var(fit)
```

Arguments

`fit` An object of class `"pvarife_result"` returned by `pvarife`.

Details

The function computes three components:

D The Hessian $D_{F,\Lambda}$ (Eq. A.4), which accounts for factor estimation uncertainty via a two-term formula.

Omega The sandwich variance Ω, accumulated unit by unit.

Bias Two bias terms: B_Ψ (from factor loading estimation) and B_γ (HAC serial correction with bandwidth $\bar{G} = \lfloor T^{1/3} \rceil$).

Notes on MATLAB replication: This implementation deviates from the original `Asymptotic_Distribution_of_beta.m` in two places, following the paper rather than the code:

1. B_γ: the MATLAB accumulation (line 189) uses only the final value of the loop variable g rather than summing over $g = 1, \ldots, \bar{G}$ as in Eq. (2.56). Corrected here.
2. Ω: MATLAB uses $\Gamma\,\mathrm{diag}(u)^2\,\Gamma^\top$, which drops the within-period cross-variable terms $u_{t,n}u_{t,m}$ present in Eq. (2.65). This function computes the per-period outer products of Eq. (2.65); in simulations this gives (weakly) better confidence-interval coverage.

Value

A list with:

bias Bias vector for $\hat{\beta}$, of the same length as `fit$beta`.

variance Asymptotic variance-covariance matrix of $\hat{\beta}$.

References

Tugan, M. (2021). Panel VAR models with interactive fixed effects. *Econometrics Journal*, 24, 225–246. doi:10.1093/ectj/utaa021

Examples

```
sim <- sim_pvarife(n_units = 30, n_time = 20, n_vars = 2,
                   n_lags = 1, n_factors = 1, seed = 1)
fit <- pvarife(sim$y, n_lags = 1, n_factors = 1, n_out = 5, n_in = 3)
avar <- asymptotic_var(fit)
cat("Bias:", avar$bias, "\n")
```

`bootstrap_irf_bands` *Recursive residual bootstrap confidence bands for IRFs*

Description

Constructs confidence bands for structural impulse responses via a recursive residual bootstrap. For each bootstrap replicate, the idiosyncratic residuals are resampled (by time index, preserving the contemporaneous correlation across variables), a new panel is generated *recursively* from the estimated VAR dynamics and the (fixed) estimated common component, the full `pvarife` model is re-estimated, and IRFs are computed. This is computationally expensive but does not rely on asymptotic approximations.

Usage

```
bootstrap_irf_bands(
  fit,
  n_periods,
  shock = 1L,
  diff_vars = integer(0),
  identification = c("short_run", "long_run"),
  n_boot = 200L,
```

```
  level = 0.95,
  seed = NULL
)
```

Arguments

`fit`	An object of class "pvarife_result".
`n_periods`	Positive integer. Number of IRF horizons.
`shock`	Positive integer. Index of the structural shock (default 1).
`diff_vars`	Integer vector. Variables to cumulate (default none).
`identification`	Character. "short_run" (default) or "long_run". Passed to `compute_irf` for each bootstrap replicate.
`n_boot`	Positive integer. Number of bootstrap replicates (default 200).
`level`	Numeric in (0, 1). Confidence level (default 0.95).
`seed`	Optional integer seed for reproducibility.

Details

Each bootstrap panel is generated as

$$y^*_{i,t} = c + \sum_{l=1}^{\ell} \Theta_l y^*_{i,t-l} + \hat{F}_t \hat{\lambda}_i + e^*_{i,t},$$

where (c, Θ_l) are the estimated coefficients, the common component $\hat{F}_t \hat{\lambda}_i$ is held fixed at its estimate, the first ℓ periods are initialised at the observed data $y^*_{i,t} = y_{i,t}$, and $e^*_{i,t}$ are residuals resampled with replacement (whole time rows, so the cross-variable correlation is kept). Because the path is generated recursively, the bootstrap correctly propagates the VAR dynamics — unlike a fixed-design scheme that reuses the original lags.

Scope of the bands. Only the idiosyncratic errors are resampled; the common component $\hat{F}_t \hat{\lambda}_i$ and the reduced-form covariance structure are held at their estimates. The resulting bands therefore capture idiosyncratic-error uncertainty only and *under-cover* when the common factors account for a large share of the variation (as they do in the simulation design of Tugan 2021). This routine is intended as a robustness check; for coefficient inference use `asymptotic_var` or `summary.pvarife_result`, and for the paper's IRF bands use `irf_bands`.

Value

An object of class "pvarife_bands" with components `irf`, `lower`, `upper`, `level`, and `method = "bootstrap"`.

See Also

`irf_bands`, `compute_irf`

Examples

```
sim   <- sim_pvarife(n_units = 20, n_time = 15, n_vars = 2,
                     n_lags = 1, n_factors = 1, seed = 1)
fit   <- pvarife(sim$y, n_lags = 1, n_factors = 1, n_out = 5, n_in = 3)
bands <- bootstrap_irf_bands(fit, n_periods = 6, n_boot = 20, seed = 42)
plot(bands)
```

---

coef.pvarife_result *Extract coefficients from a pvarife_result*

Description

Extract coefficients from a pvarife_result

Usage

```
## S3 method for class 'pvarife_result'
coef(object, ...)
```

Arguments

`object`	An object of class "`pvarife_result`".
`...`	Ignored.

Value

Named numeric vector of coefficients.

---

compute_irf *Compute impulse response functions under recursive (Cholesky) identification*

Description

Computes structural impulse responses from an estimated `pvarife_result` using a recursive (lower-triangular Cholesky) identification scheme. The identification follows the short-run restriction approach of Tugan (2021).

Usage

```
compute_irf(
  fit,
  n_periods,
  shock = 1L,
  diff_vars = integer(0),
  identification = c("short_run", "long_run"),
  bias_correct = FALSE
)
```

Arguments

`fit`	An object of class "pvarife_result".
`n_periods`	Positive integer. Number of IRF horizons to compute.
`shock`	Positive integer. Index of the structural shock (1 = first variable in the ordering). Default is 1.
`diff_vars`	Integer vector. Indices of variables for which cumulative IRFs are reported (e.g., for variables entered in first differences). Default is `integer(0)` (no cumulation). For long-run identification `diff_vars = 1L` is common when variable 1 is treated as I(1).
`identification`	Character. Either "short_run" (default, Cholesky of $\hat{\Sigma}$) or "long_run" (Blanchard-Quah: long-run multiplier lower-triangular). See Details.
`bias_correct`	Logical. If `TRUE`, the IRF is evaluated at the bias-corrected coefficient vector $\hat{\beta} - \hat{b}$ from `asymptotic_var`. Default `FALSE` (faster; use `irf_bands` for full bias-corrected inference).

Details

The MA representation is computed recursively:

$$B_0 = I_K, \quad B_h = \sum_{l=1}^{\ell} \Theta_l B_{h-l},$$

with the convention $B_j = 0$ for $j < 0$.

Short-run identification (default): The impact matrix is the lower-triangular Cholesky factor of $\hat{\Sigma}$: $A_0 = \text{chol}(\hat{\Sigma})^\top$.

Long-run identification (Blanchard-Quah type): The long-run multiplier $C(1) = (I - \sum_\ell \Theta_\ell)^{-1} A_0$ is constrained to be lower-triangular. The impact matrix is

$$A_0 = (I - \Theta)\, \text{chol}(D)^\top,$$

where $\Theta = \sum_{\ell=1}^{L} \Theta_\ell$ and $D = (I - \Theta)^{-1} \hat{\Sigma} (I - \Theta)^{-\top}$. This identification restricts shock 1 to have no long-run effect on variable 2 (in a 2-variable system). Faithful to `IRs_to_Shocks_LR_Identification.m` in the Monte Carlo replication code.

Bias correction: When `bias_correct = TRUE`, the impact matrix is evaluated at the bias-corrected coefficient vector $\hat{\beta} - \hat{b}$ from `asymptotic_var`. The uncorrected estimator is used by default (bias_correct = FALSE) for speed; users who need confidence bands can rely on irf_bands, whose median is already implicitly bias-corrected.

The impulse response to shock s at horizon h is $B_h A_0 e_s$ where e_s is the s-th standard basis vector.

Value

A matrix of dimension $K \times n\_periods$. Row k gives the response of variable k to the structural shock at horizons $1, \ldots, n\_periods$. The object has class c("pvarife_irf", "matrix").

References

Tugan, M. (2021). Panel VAR models with interactive fixed effects. *Econometrics Journal*, 24, 225–246. doi:10.1093/ectj/utaa021

See Also

irf_bands, bootstrap_irf_bands

Examples

```
sim <- sim_pvarife(n_units = 30, n_time = 20, n_vars = 2,
                   n_lags = 1, n_factors = 1, seed = 1)
fit <- pvarife(sim$y, n_lags = 1, n_factors = 1, n_out = 5, n_in = 3)
ir  <- compute_irf(fit, n_periods = 8)
dim(ir)   # 2 x 8

# Long-run identification
ir_lr <- compute_irf(fit, n_periods = 8, identification = "long_run",
                     diff_vars = 1L)
```

---

extract_factors — *Extract factors and loadings at an arbitrary coefficient vector*

Description

Given an estimated pvarife_result and an arbitrary coefficient vector beta, runs the inner EM loop to extract common factors and factor loadings. Useful for advanced users (e.g., bootstrap procedures that need factor estimates at a perturbed beta).

Usage

```
extract_factors(beta, fit, n_in = 10L)
```

Arguments

beta	Numeric vector of VAR coefficients (same length as fit$beta).
fit	An object of class "pvarife_result".
n_in	Number of inner iterations (default 10).

Details

Faithful translation of `Inner_Iteration.m` from Tugan (2021).

Value

A list with `ff` (T x r factor matrix) and `loadings` (Kr x I loading matrix).

Examples

```
sim <- sim_pvarife(n_units = 20, n_time = 15, n_vars = 2,
                   n_lags = 1, n_factors = 1, seed = 2)
fit <- pvarife(sim$y, n_lags = 1, n_factors = 1, n_out = 5, n_in = 3)
ef  <- extract_factors(fit$beta * 0.9, fit)
dim(ef$ff)       # T x r
```

---

irf_bands — *Parametric confidence bands for impulse response functions*

Description

Constructs confidence bands for structural impulse responses by drawing from the joint asymptotic distribution of the estimator and the common component (Theorem 2.3 of Tugan 2021). This is a parametric simulation, not a residual bootstrap.

Usage

```
irf_bands(
  fit,
  n_periods,
  shock = 1L,
  diff_vars = integer(0),
  identification = c("short_run", "long_run"),
  n_draw = 500L,
  level = 0.95,
  seed = NULL
)
```

Arguments

`fit`	An object of class `"pvarife_result"`.
`n_periods`	Positive integer. Number of IRF horizons.
`shock`	Positive integer. Index of the structural shock (default 1).
`diff_vars`	Integer vector. Variables to cumulate (default none).
`identification`	Character. `"short_run"` (default) or `"long_run"`. Passed to `compute_irf` for each draw. The median IRF returned is implicitly bias-corrected regardless of this choice (draws are centred on $\hat{\beta} - \hat{b}$).

`n_draw`	Positive integer. Number of simulation draws (default 500).
`level`	Numeric in (0, 1). Confidence level (default 0.95).
`seed`	Optional integer seed for reproducibility.

Details

For each of the `n_draw` repetitions, the function:

1. Draws $\beta^{(b)} \sim N(\hat{\beta} - \hat{b}, \hat{V})$ where $\hat{b}$ and $\hat{V}$ are the estimated bias and variance from `asymptotic_var`.
2. Draws the common component $\tilde{C}_{i,t,k}$ from a normal with mean $\hat{F}_t \hat{\lambda}_i$ and standard deviation
$$\sqrt{\hat{\Xi}^{*}_{i,t,k}/I + \hat{\Xi}^{**}_{i,t,k}/T},$$
capturing cross-sectional and time-series uncertainty in factor estimation.
3. Computes the IRF at $(\beta^{(b)}, \tilde{C}^{(b)})$.

The median and the $(1 - \text{level})/2$ and $(1 + \text{level})/2$ quantiles across draws give the point estimate and confidence bands.

Scope of the bands. This is a faithful implementation of the band construction in Tugan (2021) (`ConfidenceBandforIRs.m`): the bands propagate uncertainty in the VAR coefficients β and in the estimated common component, but the reduced-form covariance $\hat{\Sigma}$ is recomputed deterministically for each draw and therefore contributes little draw-to-draw variation. As a result the bands mainly reflect *coefficient* (dynamic) uncertainty and *under-state* uncertainty in the contemporaneous impact at horizon 0 (which is a function of Σ alone). Impulse responses are conventionally reported normalised by the shock's own horizon-0 response (see `plot` and the `normalise_by_h1` argument), which fixes that element to one. For formal inference on the coefficients themselves use `asymptotic_var` or `summary.pvarife_result`, whose Wald intervals attain nominal coverage in simulations.

Value

An object of class `"pvarife_bands"` with components:

irf Point estimate (median across draws, bias-corrected), $K \times n\_periods$.

lower Lower confidence bound, $K \times n\_periods$.

upper Upper confidence bound, $K \times n\_periods$.

level The confidence level used.

method `"asymptotic"`.

References

Tugan, M. (2021). Panel VAR models with interactive fixed effects. *Econometrics Journal*, 24, 225–246. doi:10.1093/ectj/utaa021

See Also

`bootstrap_irf_bands`, `compute_irf`

Examples

```
sim   <- sim_pvarife(n_units = 20, n_time = 15, n_vars = 2,
                     n_lags = 1, n_factors = 1, seed = 1)
fit   <- pvarife(sim$y, n_lags = 1, n_factors = 1, n_out = 5, n_in = 3)
bands <- irf_bands(fit, n_periods = 6, n_draw = 100, seed = 42)
plot(bands)
```

---

lag_lead_matrix — *Build a matrix of lags and/or leads*

Description

Creates a matrix whose columns are lagged or lead copies of the columns of `x`, for integer offsets `from_lag` through `to_lag`. A positive offset k produces a lag (rows $k+1$ to T of the output receive rows 1 to $T-k$ of `x`); a negative offset k produces a lead. Unfilled positions are set to `NA`.

Usage

```
lag_lead_matrix(x, from_lag, to_lag)
```

Arguments

`x`	A numeric matrix with T rows and K columns.
`from_lag`	Integer. The smallest offset to include (may be negative for leads).
`to_lag`	Integer. The largest offset to include (must be $\geq$ `from_lag`).

Details

Faithful translation of `lagleadmatrix.m` from the replication package of Tugan (2021).

Value

A matrix with T rows and $K \times (to\_lag - from\_lag + 1)$ columns. Column block j (1-indexed) corresponds to offset $from\_lag + j - 1$.

References

Tugan, M. (2021). Panel VAR models with interactive fixed effects. *Econometrics Journal*, 24, 225–246. doi:10.1093/ectj/utaa021

Examples

```
x <- matrix(1:12, nrow = 4)
lag_lead_matrix(x, 1, 2)   # two lags
lag_lead_matrix(x, -1, 1)  # one lead, contemporaneous, one lag
```

plot.pvarife_bands *Plot impulse response bands*

Description

Plot impulse response bands

Usage

```
## S3 method for class 'pvarife_bands'
plot(x, var_names = NULL, normalise_by_h1 = FALSE, ...)
```

Arguments

`x`	An object of class "pvarife_bands".
`var_names`	Character vector of variable names (optional).
`normalise_by_h1`	Logical. If `TRUE`, divide all responses by the first-horizon response of the shock variable (replicates Figure 1 of Tugan 2021). Default `FALSE`.
`...`	Ignored.

Value

A `ggplot2` plot object.

plot.pvarife_irf *Plot impulse response functions*

Description

Plot impulse response functions

Usage

```
## S3 method for class 'pvarife_irf'
plot(x, var_names = NULL, ...)
```

Arguments

`x`	An object of class "pvarife_irf".
`var_names`	Character vector of variable names (optional).
`...`	Ignored.

Value

A `ggplot2` plot object.

`print.pvarife_result` *Print method for pvarife_result*

Description

Print method for pvarife_result

Usage

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

Arguments

`x`	An object of class "`pvarife_result`".
`...`	Ignored.

Value

Invisibly returns `x`.

`pvarife` *Estimate a Panel VAR with Interactive Fixed Effects*

Description

Jointly estimates VAR coefficients β, latent common factors F, and factor loadings Λ for a panel vector autoregression with interactive fixed effects, following the iterative algorithm of Tugan (2021) based on Bai (2009).

Usage

```
pvarife(y, n_lags, n_factors, n_out = 50L, n_in = 10L, balanced_init = TRUE)
```

Arguments

`y`	A numeric array of dimension $I \times T \times K$ (units $\times$ time $\times$ variables). `NA` values are allowed for unbalanced panels. Following the original implementation, if *any* variable is missing for unit i at period t, the whole period is treated as missing for that unit. Missing periods are excluded from the coefficient update and their common component is imputed by the EM step. **Caution:** simulation evidence shows that the point estimator can exhibit noticeable finite-sample bias when the share of missing periods is substantial (roughly above 10–15% at moderate I, T); results under heavy missingness should be interpreted with care and checked for robustness (e.g., on a balanced subsample).

`n_lags` Positive integer. Lag order ℓ.

`n_factors` Positive integer. Number of interactive fixed effects r.

`n_out` Positive integer. Number of outer iterations (default 50). Corresponds to `out_number` in the MATLAB replication code.

`n_in` Positive integer. Number of inner PCA/EM iterations per outer step (default 10). Corresponds to `in_number` in the MATLAB code.

`balanced_init` Logical. If `TRUE` (default), the initial beta estimate is obtained from units that have at least 10 fully observed periods in the last window of the sample — matching the approach of the MATLAB `Initial_Step_in_Iteration.m` for unbalanced real data. Set to `FALSE` for balanced panels (e.g., Monte Carlo simulations) to skip this selection step and use all units directly.

Details

The model is

$$y_{i,t} = \sum_{l=1}^{\ell} \Theta_l y_{i,t-l} + F_t \lambda_i + e_{i,t},$$

where $y_{i,t}$ is a $K \times 1$ vector of endogenous variables for unit i at time t, F_t is an $r \times 1$ vector of unobservable common factors, λ_i is a unit-specific loading vector, and $e_{i,t}$ is an idiosyncratic error.

The algorithm alternates between:

1. An **inner loop** that extracts factors and loadings via PCA (principal components on the residual cross-product matrix) and imputes missing observations (EM step of Bai 2009).
2. An **outer loop** that updates β via least squares after projecting out the estimated factors (using $M_F = I - F(F'F)^{-1}F'$).

Value

An object of class `"pvarife_result"`, which is a list with:

beta Coefficient vector of length $K + K^2\ell$. The first K elements are equation-specific intercepts; the remaining $K^2\ell$ elements are VAR lag coefficients stacked as $[\Theta_1, \Theta_2, \ldots, \Theta_\ell]'$ (column-major).

ff Factor matrix of dimension $T \times r$ (one row per time period; analogous to MATLAB `f`).

factors_mat Block-diagonal factor matrix of dimension $TK \times Kr$, built as $I_K \otimes f_t'$.

loadings Matrix of dimension $Kr \times I$ (factor loadings per unit, stacked by variable).

sigma Reduced-form error covariance matrix ($K \times K$).

u_c Array of residuals $TK \times 1 \times I$ (NA at unobserved positions).

y_arr The original input array $I \times T \times K$ (used, e.g., for initial conditions in `bootstrap_irf_bands`).

y_c, z_c Stacked outcome/regressor arrays ($TK \times 1 \times I$ and $TK \times (K + K^2\ell) \times I$).

y_stack, z_stack Pooled outcome/regressor matrices (complete-case rows only).

i_obs Integer matrix $TK \times I$: 1 = observed, 0 = missing.

n_time_i Integer vector of length I: number of observed time periods per unit (analogous to MATLAB `TC`).

tnc_i Integer vector of length I: $n\_time\_i \times K$ (analogous to MATLAB `TNC`).

n_lags, n_factors, n_vars, n_units, n_time Dimensions.

References

Tugan, M. (2021). Panel VAR models with interactive fixed effects. *Econometrics Journal*, 24, 225–246. doi:10.1093/ectj/utaa021

Bai, J. (2009). Panel data models with interactive fixed effects. *Econometrica*, 77(4), 1229–1279.

Examples

```
sim <- sim_pvarife(n_units = 30, n_time = 20, n_vars = 2,
                   n_lags = 1, n_factors = 1, seed = 1)
fit <- pvarife(sim$y, n_lags = 1, n_factors = 1, n_out = 5, n_in = 3)
print(fit)
```

---

pvarife_sim — *Simulated panel VAR dataset with interactive fixed effects*

Description

A synthetic panel dataset generated from the DGP of Tugan (2021), Section S10, via `sim_pvarife`. Provided as a compact example dataset for vignettes and function examples.

Usage

```
pvarife_sim
```

Format

A list with the following components:

y Numeric array of dimension $50 \times 30 \times 2$ (50 units, 30 time periods, 2 variables).

beta_true True coefficient vector of length 6 (2 intercepts + 4 VAR lag coefficients).

theta_true True VAR coefficient array of dimension $2 \times 2 \times 1$.

sigma_true True reduced-form covariance matrix 2×2.

factors_true True factor matrix of dimension 30×1.

loadings_true True factor loadings matrix of dimension 2×50.

Details

Generated with `sim_pvarife(n_units = 50, n_time = 30, n_vars = 2,n_lags = 1, n_factors = 1, seed = 42)`.

Source

Simulated; see `sim_pvarife`.

References

Tugan, M. (2021). Panel VAR models with interactive fixed effects. *Econometrics Journal*, 24, 225–246. doi:10.1093/ectj/utaa021

`sim_pvarife` *Simulate panel VAR data with interactive fixed effects*

Description

Generates synthetic panel data from the DGP of Tugan (2021), Section S10, with a single common factor following an AR(1) process and factor loadings drawn from $N(1, 1)$.

Usage

```
sim_pvarife(
  n_units = 50L,
  n_time = 30L,
  n_vars = 2L,
  n_lags = 1L,
  n_factors = 1L,
  identification = c("short_run", "long_run"),
  seed = 42L
)
```

Arguments

`n_units`	Positive integer. Number of cross-sectional units I (default 50).
`n_time`	Positive integer. Number of time periods T after burn-in (default 30).
`n_vars`	Positive integer. Number of VAR variables K (default 2).
`n_lags`	Positive integer. VAR lag order (default 1; higher lags use zero coefficient matrices).
`n_factors`	Positive integer. Number of common factors (default 1; each factor has its own AR(1) process and independent loadings).
`identification`	Character. `"short_run"` (default) uses $A_0 = \mathrm{chol}(\Sigma_e)^\top$; `"long_run"` uses the Blanchard-Quah impact matrix $A_0 = (I - \Theta_1)\,\mathrm{chol}(D)^\top$ where $D = (I - \Theta_1)^{-1}\Sigma_e(I - \Theta_1)^{-\top}$. The long-run multiplier $C(1) = (I - \Theta_1)^{-1}A_0 = \mathrm{chol}(D)^\top$ is lower-triangular. Matches the MATLAB `GeneratingSynteticDataSets.m` with `Identification='WithLongRunRestrictions'`.
`seed`	Optional integer seed for reproducibility (default 42).

Details

The data generating process is:

$$y_{i,t} = c + \Theta_1 y_{i,t-1} + f_t \lambda_i + A_0 \varepsilon_{i,t},$$

where

$$\Theta_1 = \begin{pmatrix} 0.65 & 0.30 \\ 0.20 & 0.60 \end{pmatrix}, \quad \Sigma_e = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix},$$

$$f_t = 0.5 f_{t-1} + \eta_t,\ \eta_t \sim N(0, 0.5),$$

$$\lambda_{k,i} \sim N(1, 1) \text{ (independently)},$$

and $A_0 = \text{chol}(\Sigma_e)^\top$ (Cholesky identification). A burn-in of 1000 periods is discarded.

For `n_vars = 2` and `n_lags = 1`, the parameters match the MATLAB `GeneratingSynteticDataSets.m` (Short-Run identification variant).

Value

A list with:

y Array of dimension $I \times T \times K$ (unit $\times$ time $\times$ variable). No missing values.

beta_true True coefficient vector, same format as `fit$beta`.

theta_true True VAR coefficient array $K \times K \times n\_lags$.

sigma_true True reduced-form covariance matrix $K \times K$.

a0_true True structural impact matrix $K \times K$.

factors_true $T \times n\_factors$ matrix of true factors.

loadings_true $K \times n\_units$ matrix of true loadings.

identification The identification scheme used (`"short_run"` or `"long_run"`).

diff_vars_suggested Integer vector: variables that should be cumulated in `compute_irf()`. `integer(0)` for short-run; `1L` for long-run (to display cumulative responses, matching the MATLAB MC code's `DifferencedVariables = [1]`).

Examples

```
sim <- sim_pvarife(n_units = 30, n_time = 20, n_vars = 2,
                   n_lags = 1, n_factors = 1, seed = 1)
dim(sim$y)        # 30 x 20 x 2
sim$beta_true
```

`summary.pvarife_result`
Summary method for pvarife_result

Description

Summary method for pvarife_result

Usage

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

Arguments

`object`	An object of class `"pvarife_result"`.
`...`	Ignored.

Value

Invisibly returns a list with `fit`, `avar`.

Index