Package 'acpbvar'

October 31, 2026

Type Package

Title Bayesian Structural VARs with Asymmetric Conjugate Priors

Version 0.2.0

Description Implements Bayesian estimation of structural Vector Autoregressions in the asymmetric A-form of Chan (2022) <doi:10.3982/QE1381>, with an asymmetric Minnesota prior whose own-lag and cross-lag hyperparameters are selected by empirical Bayes via the closed-form marginal likelihood. Exact equation-by-equation posterior simulation in the structural form (independent Normal-Inverse-Gamma draws) provides posterior draws of the reduced-form parameters; structural identification through sign restrictions follows the algorithm of Rubio-Ramirez, Waggoner and Zha (2010) <doi:10.1111/j.1467-937X.2009.00578.x>.
Impulse responses, posterior bands, and acceptance diagnostics are computed from the accepted draws.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.0)

Imports stats, utils, graphics, grDevices

Suggests bsvarSIGNs, bsvars, knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

URL https://github.com/ursillofederica/acpbvar

BugReports https://github.com/ursillofederica/acpbvar/issues

Config/roxygen2/version 8.1.1

NeedsCompilation no

Author Federica Ursillo [aut, cre]

Maintainer Federica Ursillo <federica.ursillo@outlook.com>

Repository CRAN

Date/Publication 2026-10-31 11:20:08 UTC

Contents

acpbvar-methods . . . 2
build_s2 . . . 3
build_yX . . . 4
collect_irfs . . . 5
compute_irf . . . 6
optim_hyper . . . 7
plot_irfs . . . 8
post_sim . . . 10
recover_rf . . . 11
sign_restrict . . . 12
simulate_var_example . . . 14

Index **16**

`acpbvar-methods` *Print, summary and plot methods for acpbvar objects*

Description

`post_sim()` returns an object of class `"acpbvar_post"`, `sign_restrict()` an object of class `"acpbvar_signs"` and `collect_irfs()` an object of class `"acpbvar_irf"`. All three remain the plain list or array documented in the respective function (subsetting an `"acpbvar_irf"` with `[` gives a plain array); the classes only add the methods below.

Usage

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

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

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

## S3 method for class 'acpbvar_irf'
summary(
  object,
  horizons = NULL,
  probs = c(0.16, 0.5, 0.84),
  vars = NULL,
  shocks = NULL,
  ...
)

## S3 method for class 'summary.acpbvar_irf'
```

```
print(x, digits = 3, ...)

## S3 method for class 'acpbvar_irf'
plot(x, vars = NULL, shocks = NULL, probs = c(0.16, 0.84), ...)
```

Arguments

x, object	An object of the corresponding class.
...	For plot, further arguments passed to plot_irfs() (for example ref, col_band, xlab, mfrow); ignored otherwise.
horizons	Integer vector of horizons to summarise (default all).
probs	Quantiles to report (summary, default c(0.16, 0.5, 0.84)) or the two band quantiles (plot, default c(0.16, 0.84)).
vars, shocks	Variables and shocks to summarise or plot: integer indices or names. Default NULL means all.
digits	Number of digits when printing a summary.

Value

print methods return their argument invisibly. summary returns an object of class "summary.acpbvar_irf": a list, one element per selected shock, each a list of probs x horizons matrices, one per selected variable. plot returns NULL invisibly.

Examples

```
sim  <- simulate_var_example(T = 120, seed = 1)
mcmc <- post_sim(sim$Y, p = 1L, R = 200L, var_names = sim$var_names)
mcmc
S    <- rbind(supply = c(1, -1, 0), demand = c(1, 1, 0), monetary = c(-1, -1, 1))
acc  <- sign_restrict(mcmc, S, max_tries = 500L, verbose = FALSE)
acc
irfs <- collect_irfs(acc, n = 3L, p = 1L, H = 8L)
irfs
summary(irfs, vars = "output", shocks = "monetary", horizons = c(0, 4, 8))
plot(irfs, shocks = "monetary", mfrow = c(1, 3))
```

---

build_s2 *Estimate residual variances from univariate AR fits*

Description

For each variable, fits an AR(ar.lags) with intercept by ordinary least squares and returns the mean of the squared residuals, as in the replication code of Chan (2022). The resulting variances enter the asymmetric Minnesota prior as scaling factors. Series too short for the fit (fewer than ar.lags + 3 observations) get their sample variance.

Usage

```
build_s2(y_list, ar.lags = 4)
```

Arguments

y_list	A list of n response vectors, as produced by build_yX().
ar.lags	Integer; order of the univariate autoregression. Default 4.

Value

A list with elements:

s2 Numeric vector of length n; residual variances.

S Diagonal matrix diag(s2).

Examples

```
sim <- simulate_var_example(T = 120, seed = 1)
dat <- build_yX(sim$Y, p = 1L)
s2  <- build_s2(dat$y_list)$s2
s2                          # AR(4) residual variances, one per variable;
                            # they scale the Minnesota prior
```

---

build_yX *Build response and regressor matrices for the A-form VAR*

Description

Constructs the equation-by-equation response vectors and regressor matrices required by the asymmetric A-form representation of a structural VAR. For equation i, the regressors include the lagged values of all variables, an intercept, and (for i > 1) the contemporaneous values of variables 1 through i - 1 entered with a negative sign, so that the impact matrix is lower triangular by construction.

Usage

```
build_yX(Y, p)
```

Arguments

Y	A T x n numeric matrix of observations on the n endogenous variables, with rows ordered chronologically.
p	The VAR lag order (positive integer).

Value

A list with the following elements:

`y_list` A list of length `n`; `y_list[[i]]` is the post-lag response vector for equation `i`, of length `TT = T - p`.

`X_list` A list of length `n`; `X_list[[i]]` is the regressor matrix for equation `i`, with `TT` rows and `(i - 1) + 1 + n * p` columns (contemporaneous block, intercept, lagged variables).

`n` Number of endogenous variables.

`p` VAR lag order.

`TT` Effective sample size after lag truncation.

Examples

```
sim <- simulate_var_example(T = 120, seed = 1)
dat <- build_yX(sim$Y, p = 1L)

length(dat$y_list)          # one response vector per equation
sapply(dat$X_list, ncol)    # 4, 5, 6: intercept + n*p lags, plus the
                            # contemporaneous variables ordered before
                            # the equation (entered with a minus sign)
```

`collect_irfs` *Collect impulse responses across accepted draws*

Description

Loops `compute_irf()` over the list of accepted rotations returned by `sign_restrict()` and stacks the resulting impulse-response arrays along a fourth dimension indexing the draw.

Usage

```
collect_irfs(accepted, n, p, H)
```

Arguments

`accepted`	The list of accepted draws returned by `sign_restrict()`.
`n, p, H`	As in `compute_irf()`.

Value

An object of class `"acpbvar_irf"`: a four-dimensional array of dimension `n x (H + 1) x n x n_accepted` (variable x horizon x shock x draw), with variable and shock names as dimnames, and `print`, `summary` and `plot` methods. Subsetting with `[` returns a plain array.

Examples

```
sim  <- simulate_var_example(T = 120, seed = 1)
mcmc <- post_sim(sim$Y, p = 1L, R = 200L)
S    <- rbind(c(1, -1, 0), c(1, 1, 0), c(-1, -1, 1))
acc  <- sign_restrict(mcmc, S, max_tries = 500L, verbose = FALSE)

irfs <- collect_irfs(acc, n = 3L, p = 1L, H = 8L)
dim(irfs)                  # variable x horizon x shock x accepted draw

# Posterior median and 68% band of variable 1 to shock 3, horizons 0..8
apply(irfs[1, , 3, ], 1, quantile, probs = c(0.16, 0.5, 0.84))
```

`compute_irf` *Compute structural impulse responses for a single draw*

Description

Computes the structural impulse responses at horizons `0, ..., H` for the given impact matrix and reduced-form coefficient matrix, through the VMA recursion on the `n x n` lag matrices.

Usage

```
compute_irf(impact, B_rf, n, p, H)
```

Arguments

`impact`	The `n x n` impact matrix; columns index shocks.
`B_rf`	Reduced-form coefficient matrix as returned by `recover_rf()`.
`n, p`	Number of variables and VAR lag order.
`H`	Maximum impulse-response horizon.

Value

A three-dimensional array of dimension `n x (H + 1) x n`, indexed by `[variable, horizon, shock]`.

Examples

```
sim  <- simulate_var_example(T = 120, seed = 1)
mcmc <- post_sim(sim$Y, p = 1L, R = 50L)
S    <- rbind(c(1, -1, 0), c(1, 1, 0), c(-1, -1, 1))
acc  <- sign_restrict(mcmc, S, max_tries = 500L, verbose = FALSE)

irf1 <- compute_irf(acc[[1]]$impact, acc[[1]]$B_rf, n = 3L, p = 1L, H = 8L)
dim(irf1)                  # variable x horizon (0..H) x shock
irf1[1, , 3]               # response of variable 1 to shock 3 over horizons
```

optim_hyper *Empirical Bayes selection of Minnesota prior tightness*

Description

Selects the own-lag and cross-lag tightness parameters (κ_1, κ_2) by maximising the closed-form log marginal likelihood. Optimisation is performed by Nelder-Mead in log-space, with the transformation `kappa = exp(log_kappa)` ensuring positivity. The marginal likelihood is evaluated under the prior selected by `elicit`, so that the tightness is optimised for the prior actually used in estimation.

Usage

```
optim_hyper(
  y_list,
  X_list,
  p,
  s2,
  kappa3 = 100,
  unit_root_mean = FALSE,
  elicit = c("structural", "reduced"),
  init = c(0.04, 0.004)
)
```

Arguments

`y_list`, `p`, `s2`	As produced by `build_yX()` and `build_s2()`.
`X_list`	Per-equation regressor matrices as produced by `build_yX()`.
`kappa3`	Intercept variance (positive scalar). Default `100` is the diffuse choice.
`unit_root_mean`	Logical; if `TRUE`, sets the prior mean of the coefficient on the first own lag to `1` (random-walk prior); otherwise zero. Default `FALSE`.
`elicit`	Where the Minnesota prior on the lag coefficients is elicited: `"structural"` (default) places it directly on the structural coefficients; `"reduced"` places it on the reduced-form coefficients and derives the implied structural hyperparameters (Chan 2022, Online Supplement, Section 2): prior means are unchanged and the prior variance of equation `i` is augmented by the sum, over the equations ordered before it, of their prior variances plus their squared prior means divided by the residual variance.
`init`	Numeric vector of length 2; starting values for `(kappa1, kappa2)`. Default `c(0.04, 0.004)`.

Value

A list with `kappa1`, `kappa2`, the optimised `log_ml`, and the `convergence` code returned by `stats::optim()`.

Examples

```
sim <- simulate_var_example(T = 120, seed = 1)
dat <- build_yX(sim$Y, p = 1L)
s2  <- build_s2(dat$y_list)$s2

opt <- optim_hyper(dat$y_list, dat$X_list, p = 1L, s2 = s2)
opt$kappa1                  # own-lag tightness (larger = looser)
opt$kappa2                  # cross-lag tightness, usually much smaller
opt$convergence             # 0 means the optimiser converged
```

`plot_irfs` *Plot posterior impulse responses with credible bands*

Description

Produces a grid of base-R plots displaying the posterior median impulse response of each selected variable to each selected shock, together with a shaded credible band defined by the quantiles in `probs`. A second set of responses (`ref`), for example from another package or the true responses of a simulated process, can be overlaid for comparison.

Usage

```
plot_irfs(
  irfs,
  vars = NULL,
  shocks = NULL,
  var_names = NULL,
  shock_names = NULL,
  probs = c(0.16, 0.84),
  ref = NULL,
  col_med = "black",
  col_band = "steelblue",
  col_ref = "black",
  alpha_band = 0.3,
  xlab = "",
  ylab = "",
  main_cex = 0.9,
  mfrow = NULL
)
```

Arguments

`irfs`	A four-dimensional array of impulse responses as returned by `collect_irfs()` (variable x horizon x shock x draw).
`vars, shocks`	Which variables and shocks to plot: integer indices or names matching the dimnames of `irfs`. Default `NULL` plots all.

var_names	Optional character vector of variable names of length n; defaults to the dimnames of irfs, or "Var i".
shock_names	Optional character vector of shock names of length n_shocks; defaults to the dimnames of irfs, or "Shock k".
probs	Two-element numeric vector of quantiles for the credible band. Default c(0.16, 0.84).
ref	Optional responses to overlay (dashed median, dotted band at the same probs): either a four-dimensional array with the same variables and horizons as irfs (subset with the same vars and shocks), or a three-dimensional array (variable x horizon x draw) when a single shock is plotted.
col_med	Colour of the posterior median line. Default "black".
col_band	Colour of the credible band. Default "steelblue".
col_ref	Colour of the overlaid responses. Default "black".
alpha_band	Transparency of the credible band. Default 0.3.
xlab, ylab	Axis labels. Defaults "".
main_cex	Scaling factor for panel titles. Default 0.9.
mfrow	Panel layout passed to graphics::par(); default one row per variable and one column per shock.

Value

Called for its side effect (a grid of base-R plots). Invisibly returns NULL.

Examples

```
sim  <- simulate_var_example(T = 120, seed = 1)
mcmc <- post_sim(sim$Y, p = 1L, R = 200L)
S    <- rbind(supply = c(1, -1, 0), demand = c(1, 1, 0), monetary = c(-1, -1, 1))
acc  <- sign_restrict(mcmc, S, max_tries = 500L, verbose = FALSE)
irfs <- collect_irfs(acc, n = 3L, p = 1L, H = 8L)

plot_irfs(irfs, probs = c(0.16, 0.84))              # all panels, 68% band
plot_irfs(irfs, shocks = "monetary", mfrow = c(1, 3)) # one shock, one row

# overlay the true responses of the simulated process
irf_true <- compute_irf(sim$impact_true, cbind(sim$intercept_true, sim$B_true),
                        n = 3L, p = 1L, H = 8L)
plot_irfs(irfs, ref = array(irf_true, c(dim(irf_true), 1)))
```

post_sim *Posterior simulation for the asymmetric A-form BVAR*

Description

End-to-end wrapper: prepares the equation-by-equation responses and regressors, fits univariate AR variance estimates, optimises the Minnesota tightness via Empirical Bayes, constructs the asymmetric prior, and runs draws `R` independent posterior samples, equation by equation.

Usage

```
post_sim(
  Y,
  p,
  R,
  kappa3 = 100,
  unit_root_mean = FALSE,
  elicit = c("structural", "reduced"),
  var_names = NULL,
  seed = NULL
)
```

Arguments

`Y`	A `T x n` numeric matrix of observations on the endogenous variables.
`p`	VAR lag order.
`R`	Number of posterior draws to retain.
`kappa3`	Intercept prior variance, passed to `build_prior_asym()`.
`unit_root_mean`	Logical; whether to centre the first own-lag prior at one. Default `FALSE`.
`elicit`	Where the Minnesota prior on the lag coefficients is elicited: `"structural"` (default) places it directly on the structural coefficients; `"reduced"` places it on the reduced-form coefficients and derives the implied structural hyperparameters (Chan 2022, Online Supplement, Section 2), as in the replication code of Chan's empirical application. Passed to `build_prior_asym()` and `optim_hyper()`.
`var_names`	Optional character vector of variable names of length `n`. Stored in the returned spec.
`seed`	Optional integer seed passed to `set.seed()` before drawing. Default `NULL`: the random number generator is left untouched, so set the seed yourself before the call if you need reproducibility.

Value

An object of class `"acpbvar_post"` (a list with `print` method) with `samples` (the posterior draws of `theta` and `sigma2`), `spec` (sampler configuration, optimised tightness, log marginal likelihood, the `convergence` code returned by `stats::optim()` for the hyperparameter search, residual variances, variable names), and `dat` (the prepared response/regressor objects).

Examples

```
sim  <- simulate_var_example(T = 120, seed = 1)
mcmc <- post_sim(sim$Y, p = 1L, R = 200L)

mcmc$spec$kappa             # tightness chosen by Empirical Bayes
mcmc$spec$convergence       # 0 = optimiser converged
dim(mcmc$samples$sigma2)    # R draws x n equations
length(mcmc$samples$theta)  # R draws, each a list of n coefficient vectors
```

`recover_rf` *Recover reduced-form quantities from a structural A-form draw*

Description

Given a posterior draw of the structural coefficients and variances, reconstructs the contemporaneous impact matrix `A`, its inverse, the reduced-form coefficient matrix `B_rf = A^{-1} B_struct`, and the reduced- form innovation covariance `Sigma_rf = A^{-1} diag(sigma2) A^{-1}'`.

Usage

```
recover_rf(theta_draw, sigma2_draw)
```

Arguments

`theta_draw`	A list of length `n` containing the equation-wise structural coefficient draws produced by `draw_all()`.
`sigma2_draw`	Numeric vector of length `n` of structural variance draws.

Value

A list with `A`, `A_inv`, `B_struct`, `B_rf`, and `Sigma_rf`.

Examples

```
sim  <- simulate_var_example(T = 120, seed = 1)
mcmc <- post_sim(sim$Y, p = 1L, R = 50L)

rf <- recover_rf(mcmc$samples$theta[[1]], mcmc$samples$sigma2[1, ])
rf$A                        # lower triangular, ones on the diagonal
rf$B_rf                     # reduced-form intercept and lag coefficients
rf$Sigma_rf                 # reduced-form innovation covariance
```

`sign_restrict` *Structural identification via sign restrictions (RRWZ algorithm)*

Description

Implements the Rubio-Ramirez, Waggoner and Zha (2010, Theorem 9) rotation algorithm for sign-restriction identification. For each retained posterior draw of the reduced form, candidate rotations `Q` are drawn from the Haar measure (via QR decomposition of a standard normal matrix, with sign normalisation on the diagonal of `R`); for each candidate, all `2^n_id` sign flips on the columns are checked against the restrictions, and the first accepted rotation is stored.

Usage

```
sign_restrict(
  mcmc,
  S,
  n_id = nrow(S),
  horizons = 0L,
  method = c("per_draw", "joint"),
  max_tries = 10000,
  verbose = TRUE
)
```

Arguments

`mcmc`	A posterior object produced by `post_sim()`.
`S`	Sign restrictions, with entries `+1`, `-1` or `0` (unrestricted). Either an `n_id x n_var` matrix (rows: identified shocks, columns: variables), imposed on the impulse responses at every horizon `0, ..., horizons`; or an `n_id x n_var x (horizons + 1)` array with one matrix per horizon, so that different signs can be imposed at different horizons (a slice of zeros leaves that horizon unrestricted). Every identified shock must carry at least one non-zero restriction at some horizon. A matrix with `horizons = 0` restricts the impact only.
`n_id`	Number of identified shocks (defaults to `nrow(S)`). Columns of the rotation beyond `n_id` are left unrestricted and unlabelled.
`horizons`	Non-negative integer. Sign restrictions are imposed on the impulse responses at horizons `0, 1, ..., horizons`; the default `0L` restricts the impact matrix only. Responses at later horizons are computed with `compute_irf()` from the reduced form of each draw, so the cost per candidate rotation grows with `horizons`, and draws whose identified set is empty cost `max_tries` candidates each.
`method`	`"per_draw"` (default) or `"joint"`. With `"per_draw"`, up to `max_tries` rotations are tried for each posterior draw and the draw is kept with the first admissible one: the posterior of the reduced form is left unchanged (draws whose

identified set is empty are dropped), and the rotation is uniform within the identified set of each draw. With `"joint"`, a single rotation is drawn for each posterior draw and the draw is discarded if it fails, as in the replication code of Chan (2022): reduced-form draws are then reweighted by the size of their identified set. Chan (2022, footnote 13) describes both implementations. Under `"joint"` the number of accepted draws is random and typically much smaller than `R`; `max_tries` is ignored. Other R packages (BVAR, bsvarSIGNs) use the per-draw search.

`max_tries` Maximum number of `Q` candidates per posterior draw before moving on. Default `10000`. Draws for which no rotation is found are dropped from the output and listed in the `failed_draws` attribute. This is the intended behaviour when the identified set of a draw is empty, but it also removes draws whose identified set is small relative to `max_tries`. Check that `max(attr(x, "n_tries"))` is well below `max_tries`: if it is close to it, the limit is binding and `max_tries` should be increased.

`verbose` Logical; print a progress bar and acceptance summary.

Value

A list of length equal to the number of accepted draws; each entry contains `draw_id`, `impact`, `Q`, `B_rf`, `Sigma_rf`. The result carries two attributes: `"n_tries"`, the number of `Q` candidates needed for each accepted draw, and `"failed_draws"`, the indices of the posterior draws dropped after `max_tries`.

Examples

```
sim  <- simulate_var_example(T = 120, seed = 1)
mcmc <- post_sim(sim$Y, p = 1L, R = 100L)

# Rows: identified shocks. Columns: variables (output, inflation, rate).
# +1 / -1 restrict the sign of the impact response, 0 leaves it free.
S <- rbind(supply   = c( 1, -1,  0),
           demand   = c( 1,  1,  0),
           monetary = c(-1, -1,  1))

acc <- sign_restrict(mcmc, S, max_tries = 500L, verbose = FALSE)
length(acc)                          # accepted draws (one rotation each)
acc[[1]]$impact                      # impact matrix of the first draw
summary(attr(acc, "n_tries"))        # rotations screened per acceptance;
                                     # compare the max with max_tries
attr(acc, "failed_draws")            # draws dropped after max_tries

# Partial identification: two shocks, the third column is left free
S2  <- S[1:2, ]
acc2 <- sign_restrict(mcmc, S2, max_tries = 500L, verbose = FALSE)

# Same signs imposed on impact and on the next horizon (Uhlig 2005 uses
# several): more demanding, so more rotations are screened per draw
acc3 <- sign_restrict(mcmc, S, horizons = 1L, max_tries = 1000L, verbose = FALSE)
summary(attr(acc3, "n_tries"))
```

```
# Different signs at different horizons: one matrix per horizon
A <- array(0, c(3, 3, 3))
A[, , 1] <- S            # impact: full pattern
A[3, 3, 2] <- 1          # h = 1: interest rate still up after a monetary shock
A[1, 1, 3] <- 1          # h = 2: output still up after a supply shock
accA <- sign_restrict(mcmc, A, horizons = 2L, max_tries = 2000L, verbose = FALSE)
```

`simulate_var_example` *Simulate a small structural VAR with known sign restrictions*

Description

Generates a three-variable structural VAR(1) calibrated to deliver impulse responses with the canonical macroeconomic sign pattern: a positive supply shock raises output and lowers inflation; a positive demand shock raises both output and inflation; a positive monetary-policy shock lowers output and inflation while raising the interest rate. The function returns the simulated sample together with the true structural impact matrix and the true reduced-form coefficients, so that recovery can be validated against the data-generating process.

Usage

```
simulate_var_example(T = 200, seed = NULL)
```

Arguments

`T`	Integer; number of post-burn-in observations to return. Default `200`.
`seed`	Optional integer seed passed to `set.seed()`. Default `NULL` (generator left untouched).

Details

The simulation discards a burn-in of 100 observations to wash out the influence of the zero initial condition.

Value

A list with the following elements:

`Y` `T x 3` matrix of simulated observations on output, inflation, and the interest rate.

`impact_true` The `3 x 3` structural impact matrix used in the DGP; columns index shocks (supply, demand, monetary).

`B_true` The `3 x 3` reduced-form coefficient matrix on the first lag.

`intercept_true` The 3-vector of intercepts in the DGP.

`var_names` Variable names.

`shock_names` Shock names corresponding to the columns of `impact_true`.

Examples

```
sim <- simulate_var_example(T = 200, seed = 42)
head(sim$Y)
sim$impact_true
```

Index

acpbvar-methods, 2

build_prior_asym(), *10*
build_s2, 3
build_s2(), *7*
build_yX, 4
build_yX(), *4*, *7*

collect_irfs, 5
collect_irfs(), *2*, *8*
compute_irf, 6
compute_irf(), *5*, *12*

draw_all(), *11*

graphics::par(), *9*

optim_hyper, 7
optim_hyper(), *10*

plot.acpbvar_irf *(*acpbvar-methods*)*, 2
plot_irfs, 8
plot_irfs(), *3*
post_sim, 10
post_sim(), *2*, *12*
print.acpbvar_irf *(*acpbvar-methods*)*, 2
print.acpbvar_post *(*acpbvar-methods*)*, 2
print.acpbvar_signs *(*acpbvar-methods*)*, 2
print.summary.acpbvar_irf
        *(*acpbvar-methods*)*, 2

recover_rf, 11
recover_rf(), *6*

set.seed(), *10*, *14*
sign_restrict, 12
sign_restrict(), *2*, *5*
simulate_var_example, 14
stats::optim(), *7*, *10*
summary.acpbvar_irf *(*acpbvar-methods*)*, 2