Package 'bvarnet'

November 19, 2026

Title Bayesian Estimation of Dynamic VAR Models using Stan

Version 1.0.4

Author Florian Metwaly [aut, cre, cph] (ORCID:
<https://orcid.org/0009-0001-8357-7870>)

Maintainer Florian Metwaly <f.j.metwaly@uva.nl>

Description Bayesian estimation of multilevel Vector Autoregression (VAR) models using Stan. Supports Gaussian, Binary, and Ordinal (adjacent category) outcome variables with random effects and customizable priors.

License GPL (>= 3)

Copyright file inst/COPYRIGHTS

Encoding UTF-8

URL https://flo1met.github.io/bvarnet/,
https://github.com/flo1met/bvarnet

BugReports https://github.com/flo1met/bvarnet/issues

RoxygenNote 7.3.3

Additional_repositories https://stan-dev.r-universe.dev

Imports digest, instantiate, jsonlite, logspline, mvtnorm, posterior

Suggests cmdstanr, codetools, bayesplot, knitr, qgraph, Rcpp, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Config/Needs/website pkgdown

VignetteBuilder knitr

Depends R (>= 4.0.0)

LazyData true

NeedsCompilation yes

Repository CRAN

Date/Publication 2026-11-19 09:40:02 UTC

Contents

bf_table . . . 2
bvar . . . 3
bvarnet_clear_model_cache . . . 7
bvarnet_model_cache_dir . . . 8
bvarnet_setup_models . . . 9
compare_to_truth . . . 10
extract_draws . . . 11
extract_network_matrix . . . 12
extract_param . . . 13
extract_random_effects . . . 13
extract_temporal . . . 14
format.bvarnet_prior . . . 15
get_default_priors . . . 16
print.bvarnet . . . 16
print.bvarnet_prior . . . 17
print.bvarnet_priors . . . 17
print.summary.bvarnet . . . 18
prior . . . 18
set_priors . . . 19
sim_var . . . 20
studentlife . . . 22
summary.bvarnet . . . 22

Index **24**

`bf_table` *Compute Savege-Dickey Bayes factors*

Description

Computes Savage-Dickey density ratio Bayes factors for each (requested set of) parameter in the model. By default, all applicable parameters are tested and returned in a tidy data frame. The `type` argument controls which parameter groups are included; the `variable` argument can be used to filter to effects involving specific variables. The `log_BF10` argument allows including the natural log of the Bayes factor in the output, and `round` controls numeric rounding of the results.

Usage

```
bf_table(
  object,
  type = "all",
  lag = 1L,
  null_value = 0,
  variable = NULL,
  log_BF10 = FALSE,
  round = 5L
)
```

Arguments

`object`	A `bvarnet` object returned by `bvar()`.
`type`	Character vector specifying which parameter groups to test. Use `"all"` (default) to include all applicable groups automatically. Available options: `"ar"` Autoregressive effects (self-loops). Per-cell BFs for the lag specified by `lag`, plus a joint BF. `"cl"` Cross-lagged effects. Same structure as `"ar"`. `"intercepts"` Intercept parameters. Skipped automatically for ordinal outcomes. `"fe"` Non-intercept fixed effects (covariates). `"lag_fe"` Joint BFs for lag × covariate interaction terms. Only available when the model was fitted with `fe_interactions` containing lag terms. `"temporal"` Joint BF for the entire temporal structure (all AR + CL parameters across all lags). When lag × covariate interactions are present, additional omnibus rows are included.
`lag`	Integer; which lag block to use (default 1). Applies to `"ar"` and `"cl"` types.
`null_value`	Numeric scalar; the null hypothesis value (default 0).
`variable`	Character vector or `NULL` (default). One or more variable names. When set, only effects involving these variables are included.
`log_BF10`	Logical; if `TRUE`, an additional `log_BF10` column (natural log of `BF10`) is appended to the output. Default is `FALSE`.
`round`	Integer or `NULL`; number of decimal places to round numeric output columns. Default is `5`. Set to `NULL` to disable rounding.

Value

A data frame with columns: `type`, `predictor`, `outcome`, `BF10` (and optionally `log_BF10`).

Which prior the Bayes factor divides by

The Savage-Dickey density ratio is only valid against the prior the model was actually fitted with, so `bf_table()` evaluates the prior density using `object$priors_effective` — resolved per outcome, since Gaussian nodes scale their default priors by the outcome SD (see `set_priors`). For a Gaussian outcome with default priors this makes `intercepts` and `fe` Bayes factors differ from what the unscaled `object$priors` would give, by roughly a factor of `sd(y)`. `ar`, `cl` and `temporal` tests are unaffected: `phi` priors are never scaled.

`bvar` *Fit a Bayesian multilevel VAR network model*

Description

The `bvar` function estimates the posterior distribution of the specified Bayesian (Multilevel) Vector Autoregression.

Usage

```
bvar(
  id_col,
  time_col,
  beep_col = NULL,
  y_cols,
  x_cols = NULL,
  center_x = FALSE,
  fe_interactions = NULL,
  re_interactions = NULL,
  re_cols = NULL,
  re_temporal = FALSE,
  K = 1,
  na_action = c("listwise"),
  data,
  family = c("bernoulli", "ordinal", "gaussian"),
  priors = set_priors(),
  iter = 4000,
  warmup = 1000,
  chains = 4,
  cores = 1,
  seed = NULL,
  adapt_delta = NULL,
  max_treedepth = NULL,
  save_data = TRUE,
  ...
)
```

Arguments

id_col	Character. Name of the subject/group identifier column.
time_col	Character scalar naming the time-index column in `data`. Its meaning depends on `beep_col`: • If `beep_col = NULL` (default), this column contains the scheduled occasion index for one sequence per subject. Values must be finite and integer-valued, with one unit representing one scheduled lag step. Indices must continue across the subject's sequence rather than restart each day. For lag k, the predecessor must have index `time - k`. No separate day or overnight boundary is inferred in this mode. • If `beep_col` names a column, `time_col` identifies the day or session. `beep_col` then contains finite integer-valued beep indices within each day/session, with one unit representing one scheduled lag step. Beep indices may restart in each day/session. For lag k, the predecessor must have index `beep - k` in the same subject and day/session. Preserve the original scheduled labels when observations are missing. Rows without a complete, correctly spaced lag window are excluded as responses from

	the likelihood; their observed outcomes may still be used as predictors for later eligible responses.
beep_col	NULL (default), or a character scalar naming the scheduled occasion index within each day/session, specified in the time_col column. If NULL, time_col itself supplies the occasion index for one sequence per subject, and no day/session splitting is performed. If supplied, time_col must identify the day/session and this column must contain finite integer-valued beep indices, with one unit representing one scheduled lag step. Beep indices may restart in each day/session. For lag k, the predecessor must have index beep - k in the same subject and day/session.
y_cols	Character vector. Names of the outcome columns.
x_cols	Character vector or NULL. Names of the covariate columns.
center_x	Logical. Grand-mean centre covariates before fitting? Default FALSE.
fe_interactions	List or NULL. Fixed-effect interaction terms to add to the design matrix. Each element is a character vector of column names to interact, or c("lag", "x") to interact all lag columns with a covariate.
re_interactions	List or NULL. Random-effect interaction terms.
re_cols	Character vector. Columns from X and/or "Intercept" to include as random slopes.
re_temporal	Logical. Include random slopes on lag predictors? Default FALSE.
K	Integer. AR order. Default 1.
na_action	Character. Missing-data strategy; currently only "listwise".
data	Data frame in long format.
family	Character scalar or vector. Observation model per node. A scalar is recycled to all y_cols. A vector of length length(y_cols) (named or positional) specifies per-node families. Valid values: "bernoulli", "ordinal", "gaussian".
priors	A bvarnet_priors object from set_priors(). Defaults to set_priors() (package defaults).
iter	Integer. Number of post-warmup iterations per chain. Default 4000.
warmup	Integer. Number of warmup iterations per chain. Default 1000.
chains	Integer. Number of MCMC chains. Default 4.
cores	Integer. Number of chains to run in parallel. Default 1.
seed	Integer or NULL. RNG seed.
adapt_delta	Numeric in (0, 1). Target average proposal acceptance probability during warmup adaptation. Higher values (e.g., 0.95–0.99) reduce divergences at the cost of slower sampling. Default NULL (CmdStan default of 0.8).
max_treedepth	Integer. Maximum depth of the NUTS binary tree. Increasing this allows the sampler to take more leapfrog steps per iteration, which can help with difficult posteriors (e.g., funnels in hierarchical logistic models) but increases computation. Default NULL (CmdStan default of 10).

save_data — Logical. Keep the Stan data used for estimation in the returned object's standata slot. Default TRUE. Set to FALSE to remove these data from the fitted model before saving or sharing it.

... — Additional named arguments forwarded to the CmdStanR $sample() method, e.g. init, refresh, thin, step_size, or show_messages. See ?cmdstanr::"model-method-sample" for the full list. Arguments that bvar() sets itself (data, seed, iter_warmup, iter_sampling, chains, parallel_chains, adapt_delta, max_treedepth, and CmdStanR's deprecated aliases for them) are rejected with an error; use the corresponding bvar() argument instead.

Value

A bvarnet object (a named list) with slots: draws, convergence, diagnostics, timing, metadata, return_codes, family, model_info, priors, and priors_effective. The standata slot contains the estimation data when save_data = TRUE (default) and is NULL when save_data = FALSE. The model_info slot retains only dimensions and variable/interaction labels needed for summaries and parameter extraction. No additional data frame is stored.

Scheduled time and missing observations

Index columns identify eligible transitions; they do not automatically add time or day effects as covariates. A daily series with one observation per day can use time_col = "day" alone: consecutive days are ordinary lag steps. With multiple observations per day, supply both day and beep columns to exclude overnight transitions. Day/session labels may be numeric, character, factor or Date; occasion indices must be numeric integers. Retained keys must be unique within subject (and day/session when supplied), with no missing keys.

Only complete lag windows enter the likelihood. For indices 1, 2, 4, 5, a VAR(1) uses 1 -> 2 and 4 -> 5, excluding 2 -> 4. With day/beep indices, day 1 beep 5 never predicts day 2 beep 1, while day 2 beep 1 can predict day 2 beep 2. Subject random effects remain shared across that subject's days. Occasion-level listwise deletion is applied to all selected outcomes and covariates before constructing lags. This is not a general correction for missingness under MAR or MNAR.

The former skip_lag argument and zero-filled gap likelihood were removed because they could bias temporal and non-temporal parameters. Passing the removed argument is an error. Fits may use fewer responses and give different estimates than the former default.

Internal predictor centering

Stan samples intercepts (Gaussian/Bernoulli) or thresholds (ordinal) using centered completed fixed-effect and lag columns. Reported beta, kappa, priors and Bayes-factor hypotheses retain the reference of the public design produced with the requested center_x setting. Random-effect designs and priors are unchanged. This coordinate change can improve sampling, but does not guarantee a speed improvement.

Requested versus effective priors

priors holds the priors as you specified them. priors_effective holds the priors Stan actually sampled under: one bvarnet_priors object per outcome. The two differ for Gaussian outcomes left at their default priors, where intercept, beta and sigma scales are multiplied by the outcome

SD so that unit-scale defaults stay weakly informative on the raw data scale. User-supplied priors are never rescaled.

Bayes factors (bf_table) divide by priors_effective, as the Savage-Dickey density ratio requires. Note that the joint path (family a single value) scales every outcome by the mean SD across outcomes, whereas the nodewise path (mixed family) scales each outcome by its own SD, so the same data can imply slightly different effective priors depending on which path runs.

See Also

bvarnet_setup_models, which must be run once before the first bvar() call to set up the required Stan models (either by downloading precompiled binaries or compiling them locally).

Examples

```
# Scheduled indices retain missing occasions rather than compressing time.
single_index <- data.frame(id = 1, occasion = c(1, 2, 4, 5), y = c(1, 2, 3, 4))
# time_col = "occasion": transitions 1 -> 2 and 4 -> 5 are eligible.
day_beep <- data.frame(id = 1, day = c(1, 1, 1, 2, 2),
                       beep = c(1, 2, 4, 1, 2), y = c(1, 2, 3, 4, 5))
# time_col = "day", beep_col = "beep": only within-day beeps 1 -> 2.
## Not run:
# Run bvar on studentlife data
data(studentlife, package = "bvarnet")
fit <- bvar(
  id_col = "id",
  time_col = "day",
  y_cols = c("anxious", "calm", "conventional", "critical", "dependable"),
  re_temporal = TRUE,
  K = 1,
  data = studentlife,
  family = "ordinal",
  priors = set_priors(),
  seed = 1337)

summary(fit)

## End(Not run)
```

---

bvarnet_clear_model_cache

Remove bvarnet's cached Stan model binaries

Description

Remove bvarnet's cached Stan model binaries

Usage

```
bvarnet_clear_model_cache(all_versions = FALSE, quiet = FALSE)
```

Arguments

`all_versions`	Logical. Remove the entire cache (all versions), not just the entry for the currently installed version.
`quiet`	Logical. Suppress the confirmation message.

Value

Invisibly, the path removed.

See Also

`bvarnet_setup_models()` to set models up again afterward, `bvarnet_model_cache_dir()` to inspect the cache path.

`bvarnet_model_cache_dir` *Path to bvarnet's Stan model cache directory*

Description

Path to bvarnet's Stan model cache directory

Usage

```
bvarnet_model_cache_dir()
```

Value

Character scalar: the cache directory for the currently installed package version (may not exist yet).

See Also

`bvarnet_setup_models()` to populate this directory, `bvarnet_clear_model_cache()` to remove it.

`bvarnet_setup_models` *Set up bvarnet's precompiled Stan models*

Description

`bvar()` requires compiled Stan model executables. When installing from a CRAN/r-universe binary (no CmdStan at install time) they are not built automatically, so this function sets them up: either by downloading precompiled, toolchain-free binaries for your platform, or by compiling them locally with your own CmdStan installation. Either way the result is cached in a per-user cache directory (see `bvarnet_model_cache_dir()`) and persists across sessions.

Usage

```
bvarnet_setup_models(
  method = c("download", "compile"),
  force = FALSE,
  ask = interactive()
)
```

Arguments

`method`	`"download"` or `"compile"`. If omitted, and the session is interactive, you are prompted to choose.
`force`	Logical. If `TRUE`, clear the current cache entry first and set up from scratch.
`ask`	Logical. Whether to prompt for confirmation before downloading or compiling. Defaults to `interactive()`. When `FALSE`, a download is only performed if pre-authorised via `options(bvarnet.allow_download = TRUE)` or the `BVARNET_ALLOW_DOWNLOAD` environment variable.

Details

Downloaded binaries are only ever fetched over HTTPS from this package's official GitHub repository, and are verified against a published SHA-256 checksum before being made executable; a mismatch aborts and deletes the download. The download is also pinned to a hash of your installed package's Stan sources, so a binary can never silently be paired with the wrong model version.

`bvar()` itself never downloads or prompts – all network activity happens here, in this explicitly user-invoked function, so scripts and non-interactive sessions never trigger a download by surprise.

Value

Invisibly, `TRUE` on success, `FALSE` if declined or unavailable.

See Also

`bvarnet_model_cache_dir()` to inspect where models are cached, `bvarnet_clear_model_cache()` to remove a cached entry, and `bvar()` for the function that uses the resulting models.

Examples

```
## Not run:
# Interactive: prompts you to choose download vs. compile
bvarnet_setup_models()

# Non-interactive / scripted, pre-authorised for download:
options(bvarnet.allow_download = TRUE)
bvarnet_setup_models(method = "download", ask = FALSE)

# Force a clean re-setup (e.g. after a corrupted cache)
bvarnet_setup_models(method = "download", force = TRUE)

## End(Not run)
```

---

compare_to_truth *Compare fitted model parameters to simulation truth*

Description

Extracts posterior summaries from a fitted `bvarnet` object and compares them to the true parameter values used for data generation.

Usage

```
compare_to_truth(
  fit,
  truth,
  ci_width = 0.95,
  bayes_factor = FALSE,
  null_value = 0
)
```

Arguments

`fit`	A fitted `bvarnet` object (output from `bvar()`).
`truth`	The `truth` component from `sim_var()` output.
`ci_width`	Numeric scalar strictly between 0 and 1. Mass of the equal-tailed credible interval used for `ci_lower`, `ci_upper`, and the `covered` indicator (default 0.95).
`bayes_factor`	Logical; if `TRUE`, compute Savage-Dickey BFs for beta and phi parameters and append `BF01`, `BF10`, and `bf_correct` columns. `bf_correct` is `TRUE` when BF01 > 1 for true null parameters (true value == `null_value`) and BF10 > 1 for true non-null parameters. Default `FALSE`.
`null_value`	Numeric scalar; the null hypothesis value for Bayes factor computation (default 0). Only used when `bayes_factor = TRUE`.

Value

A data frame with columns: parameter, node, true_value, post_mean, post_sd, ci_lower, ci_upper, covered (logical), and optionally BF01, BF10, bf_correct.

`extract_draws` *Extract raw posterior draws for one or more parameter blocks*

Description

Returns an (`iterations * chains`) by `params` matrix with Stan-indexed column names (e.g. `"beta[1,1]"`, `"phi[2,3]"`). Several blocks can be requested at once; their columns are returned side by side in one matrix.

Usage

```
extract_draws(object, parameter = "all")
```

Arguments

`object`	A `bvarnet` object returned by `bvar`.
`parameter`	Character vector naming one or more of `"beta"`, `"phi"`, `"sd_u"`, `"sigma"`, `"kappa"`, or `"lp__"`; or the single value `"all"` (the default), which returns every block the fitted model actually has draws for.

Details

The blocks are the fixed effects `beta`, the temporal coefficients `phi`, the random-effect SDs `sd_u`, the residual SDs `sigma`, the ordinal thresholds `kappa`, and `lp__`, the sampler's log density.

Value

A numeric matrix with one row per posterior draw and one column per Stan parameter element, named with the Stan index (e.g. `"phi[1,2]"`). When several blocks are requested, their columns appear in the order the blocks were named. Use `extract_param` for a labelled summary table instead of raw draws.

Examples

```
## Not run:
# One block
phi <- extract_draws(fit, "phi")

# Several blocks, side by side in one matrix
d <- extract_draws(fit, c("phi", "kappa"))
colnames(d)

# Everything this model has
d <- extract_draws(fit)
```

```
## End(Not run)
```

`extract_network_matrix` *Extract a network matrix of temporal coefficients*

Description

Returns a named `p` x `p` matrix of posterior summary statistics for the VAR lag coefficients at a chosen lag, suitable for network visualisation (e.g., with **igraph** or **qgraph**).

Usage

```
extract_network_matrix(
  object,
  lag = 1L,
  stat = c("mean", "median", "ci_lower", "ci_upper"),
  ci_level = 0.95
)
```

Arguments

`object`	A `bvarnet` object returned by `bvar`.
`lag`	Integer. Which lag block. Default 1.
`stat`	Character. Summary statistic to fill the matrix with: `"mean"` (default), `"median"`, `"ci_lower"`, or `"ci_upper"`.
`ci_level`	Numeric scalar strictly between 0 and 1; the mass of the equal-tailed credible interval. Default `0.95`. Only used when `stat` is `"ci_lower"` or `"ci_upper"`.

Value

A named `p` x `p` numeric matrix. Element `[i, j]` gives the effect of variable `i` (lagged) on variable `j` (outcome). Row and column names are the outcome variable names.

extract_param *Extract labelled parameter summaries from a fitted bvarnet model*

Description

Returns a single flat data frame with posterior summaries (mean, median, and an equal-tailed credible interval) and convergence diagnostics (Rhat, ESS) for all model parameters.

Usage

```
extract_param(
  object,
  bayes_factor = FALSE,
  null_value = 0,
  type = NULL,
  ci_level = 0.95
)
```

Arguments

object	A bvarnet object returned by bvar().
bayes_factor	Logical; if TRUE, append BF01 and BF10 columns computed via the Savage-Dickey density ratio for beta and phi parameters. Default FALSE.
null_value	Numeric scalar; the null hypothesis value for Bayes factor computation (default 0). Only used when bayes_factor = TRUE.
type	Character vector or NULL (default). If supplied, only rows matching the given type(s) are returned. Valid values are: "Intercept", "Fixed Effect", "Autoregressive", "Cross-lagged", "Random Effect SD", "Residual SD", "Threshold".
ci_level	Numeric scalar strictly between 0 and 1; the mass of the equal-tailed credible interval reported in ci_lower and ci_upper. Default 0.95.

Value

A data frame with columns: type, predictor, outcome, mean, median, ci_lower, ci_upper, rhat, ess_bulk, ess_tail, and optionally BF01, BF10.

extract_random_effects *Extract random-effect summaries*

Description

Returns random-effect standard deviations (group-level variance), subject-level posterior means, or the full posterior draws of the subject-level random effects u.

Usage

```
extract_random_effects(
  object,
  what = c("sd", "mean_u", "draws_u"),
  ci_level = 0.95
)
```

Arguments

object	A bvarnet object returned by bvar.
what	Character. What to extract: "sd" Data frame of random-effect SD summaries (from extract_param). "mean_u" 3D array [node, subject, re] of posterior means of subject-level effects. "draws_u" 4D array [draw, node, subject, re] of full posterior draws.
ci_level	Numeric scalar strictly between 0 and 1; the mass of the equal-tailed credible interval reported in ci_lower and ci_upper. Default 0.95. Only used when what = "sd".

Value

Depends on what; see above.

Array layout

The "mean_u" and "draws_u" arrays are indexed [node, subject, re] (with a leading draw dimension for "draws_u"), matching the Stan declaration array[p] matrix[J, n_re] u. node is named after the outcome columns, subject after the subject index 1..J, and re after the random-effect design columns. Elements are placed by parsing the u[node, subject, re] indices from the draws, so the layout is the same whether the model was fitted through the joint or the nodewise path.

---

extract_temporal *Extract temporal (VAR lag) effects*

Description

Returns a data frame of autoregressive and/or cross-lagged parameter summaries with convergence diagnostics, filtered by lag and effect type.

Usage

```
extract_temporal(
  object,
  lag = NULL,
  effect = c("all", "ar", "cl"),
  bayes_factor = FALSE,
```

```
  null_value = 0,
  ci_level = 0.95
)
```

Arguments

object	A bvarnet object returned by bvar.
lag	Integer or NULL. If specified, only effects from this lag are returned. Default NULL (all lags).
effect	Character. One of "all" (default), "ar" (autoregressive only), or "cl" (cross-lagged only).
bayes_factor	Logical; if TRUE, append BF columns. Default FALSE.
null_value	Numeric; null hypothesis for BF. Default 0.
ci_level	Numeric scalar strictly between 0 and 1; the mass of the equal-tailed credible interval reported in ci_lower and ci_upper. Default 0.95.

Value

A data frame with columns type, predictor, outcome, mean, median, ci_lower, ci_upper, rhat, ess_bulk, ess_tail, and optionally BF01, BF10.

---

format.bvarnet_prior *Format a bvarnet_prior for printing*

Description

Format a bvarnet_prior for printing

Usage

```
## S3 method for class 'bvarnet_prior'
format(x, half = FALSE, ...)
```

Arguments

x	A bvarnet_prior object.
half	Logical; if TRUE prepends "Half-" to indicate a half-prior (used for positive-constrained parameters like sd_u and sigma).
...	Ignored.

Value

A character string.

get_default_priors *Get the default prior specification for a given model family*

Description

Returns a bvarnet_priors object showing the default priors that apply to a particular model configuration. Parameters irrelevant to the chosen family or model structure are omitted, so the returned object reflects what the sampler will actually use.

Usage

```
get_default_priors(family = NULL, has_re = TRUE)
```

Arguments

family	Character (optional). One of "bernoulli", "ordinal", "gaussian". When NULL (the default), all parameter priors are shown.
has_re	Logical. Does the model include random effects? Default TRUE. When FALSE, the sd_u prior is omitted.

Value

A bvarnet_priors object.

print.bvarnet *Print a bvarnet model object*

Description

Displays a brief summary of the fitted model: family, dimensions, Rhat, divergences, chain return codes, priors, and total sampling time.

Usage

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

Arguments

x	A bvarnet object.
...	Ignored.

Value

x invisibly.

print.bvarnet_prior *Print a bvarnet_prior*

Description

Print a bvarnet_prior

Usage

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

Arguments

x	A bvarnet_prior object.
...	Passed to format.bvarnet_prior().

Value

x invisibly.

print.bvarnet_priors *Print a bvarnet_priors specification*

Description

Shows only the priors explicitly set by the user. When no priors have been overridden (all defaults), a compact note is printed instead.

Usage

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

Arguments

x	A bvarnet_priors object.
...	Ignored.

Value

x invisibly.

print.summary.bvarnet *Print a bvarnet summary*

Description

Pretty-prints the output of `summary.bvarnet`, grouping parameters by type and displaying convergence information. Each group is truncated to `max_rows` rows; use `extract_param()` or dedicated extractors to see full output.

Usage

```
## S3 method for class 'summary.bvarnet'
print(x, digits = 3, max_rows = 10, ...)
```

Arguments

`x`	A `summary.bvarnet` object.
`digits`	Number of decimal digits for numeric columns. Default 3.
`max_rows`	Maximum number of rows to print per parameter group. Default 10.
`...`	Ignored.

Value

`x` invisibly.

prior *Construct a single prior distribution*

Description

Builds a `bvarnet_prior` object specifying the prior family and its parameters. Supported families in Phase 1 are `"normal"`, `"student_t"`, and `"cauchy"`.

Usage

```
prior(family, loc = 0, scale = 1, df = 7)
```

Arguments

`family`	Character. One of `"normal"`, `"student_t"`, `"cauchy"`.
`loc`	Location parameter (default 0).
`scale`	Scale parameter (default 1). Must be > 0.
`df`	Degrees of freedom for `"student_t"` (default 7). Must be > 0 when `family = "student_t"`.

Value

A bvarnet_prior S3 object.

---

set_priors *Build a prior specification object for* bvar()

Description

Returns a bvarnet_priors object containing a bvarnet_prior for every model parameter type. Any argument left as NULL uses the package default. Available prior distributions are:

- normal(loc, scale)
- student_t(loc, scale, df)
- cauchy(loc, scale) For standart deviations and random effects, the prior is automatically converted to a half-prior (truncated at loc) in the Stan code, so the printed format reflects this.

Usage

```
set_priors(
  intercept = NULL,
  beta = NULL,
  phi = NULL,
  sd_u = NULL,
  kappa = NULL,
  sigma = NULL
)
```

Arguments

intercept	Prior for the intercept. Only applies to gaussian and bernoulli models; for ordinal models the intercept is absorbed into the kappa (threshold parameter).
beta	Prior for fixed-effect regression coefficients (slopes).
phi	Prior for lag coefficients.
sd_u	Prior for random-effect standard deviations (half-prior).
kappa	Prior for the adjacent-category thresholds (ordinal models only).
sigma	Prior for residual standard deviation (gaussian models only; half-prior).

Value

A bvarnet_priors S3 object.

Automatic scaling of Gaussian defaults

Gaussian outcomes are modelled on their raw scale, so `bvar()` widens the default `intercept`, `beta` and `sigma` scales by the outcome SD before passing them to Stan. A default `beta ~ Normal(0, 1)` on data with `sd(y) = 17.8` therefore becomes `Normal(0, 17.8)`. This keeps the unit-scale defaults weakly informative whatever the units of `y`, but it means the default you see here is not the prior that was used.

A prior you pass explicitly is taken at face value and never rescaled. The priors actually used are recorded on the fitted object as `priors_effective` (see `bvar`), are reported by `print()`, and are what `bf_table` divides by.

`sim_var` *Simulate data from a multilevel VAR model*

Description

Generates data from the generative model implied by each Stan model family. Useful for testing parameter recovery and model validation.

Usage

```
sim_var(
  N,
  T_obs,
  p,
  K = 1L,
  family = c("bernoulli", "ordinal", "gaussian"),
  alpha = NULL,
  gamma = NULL,
  Phi = NULL,
  sigma = NULL,
  kappa = NULL,
  q = 0L,
  x_gen = NULL,
  sd_alpha = 0.5,
  sd_phi = 0.2,
  sd_gamma = NULL,
  re_temporal = FALSE,
  C = 5L,
  burnin = 500L,
  seed = NULL
)
```

Arguments

`N`	Integer. Number of subjects (groups).
`T_obs`	Integer. Number of time points per subject.

p	Integer. Number of outcome nodes.
K	Integer. AR order (default 1).
family	Character. One of "bernoulli", "ordinal", "gaussian".
alpha	Numeric vector of length p. Population intercepts (on logit scale for bernoulli, identity for gaussian). For ordinal, this is absorbed into kappa and should be left NULL. Generated if NULL.
gamma	Matrix q x p. Population covariate effects. Generated if NULL and q > 0.
Phi	Matrix (p*K) x p. Population lag coefficients. Generated if NULL.
sigma	Numeric vector of length p. Residual SD per node (gaussian only). Generated if NULL.
kappa	List of p numeric vectors, each of length C-1. Adjacent-category thresholds per node (ordinal only). Generated if NULL.
q	Integer. Number of covariates (default 0).
x_gen	Function f(N, T_obs) returning an N x T_obs x q array of covariates. If NULL, default generation is used.
sd_alpha	Numeric. SD of random intercepts (scalar or p-vector). Default 0.5. Set to 0 to simulate a fixed-effects-only model with no between-person variation in intercepts.
sd_phi	Numeric. SD of random lag coefficients (scalar or matrix). Default 0.2.
sd_gamma	Numeric or NULL. SD of random covariate slopes. NULL means no random slopes on covariates.
re_temporal	Logical. Include random slopes on lag predictors? Default FALSE.
C	Integer. Number of ordinal categories (ordinal only, default 5).
burnin	Integer. Number of time points to discard as warmup before recording data (default 500). The VAR process is simulated for burnin + T_obs time points per subject, and the first burnin are discarded. This allows the process to reach its stationary distribution before data collection begins.
seed	Integer or NULL. RNG seed.

Details

To simulate a VAR without any random effects (i.e. all subjects share identical parameters), set sd_alpha = 0, re_temporal = FALSE (the default), and sd_gamma = NULL (the default).

Value

A list with two components:

data A long-format data frame with columns id, t, y_1, ..., y_p, and optionally x_1, ..., x_q.

truth A list of true generating parameters.

studentlife *StudentLife Data*

Description

Assessing mental health, academic performance and behavioral trends of college students using smartphones. Wang, R., Chen, F., Chen, Z., Li, T., Harari, G., Tignor, S., Zhou, X., Ben-Zeev, D., & Campbell, A. T. (2014). StudentLife: assessing mental health, academic performance and behavioral trends of college students using smartphones. Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, 3–14. https://doi.org/10.1145/2632048.2632054

Usage

```
studentlife
```

Format

`studentlife`**:**
A data frame with 1912 rows and 72 columns:

Source

https://openesmdata.org/datasets/0004_wang/

https://doi.org/10.1145/2632048.2632054

summary.bvarnet *Summary method for bvarnet objects*

Description

Returns a labelled posterior summary table grouped by parameter type, with convergence diagnostics and optional Bayes factors. Wraps `extract_param`.

Usage

```
## S3 method for class 'bvarnet'
summary(object, bayes_factor = FALSE, null_value = 0, ..., ci_level = 0.95)
```

Arguments

`object`	A `bvarnet` object returned by `bvar`.
`bayes_factor`	Logical; if `TRUE`, append Savage-Dickey BF columns. Default `FALSE`.
`null_value`	Numeric scalar; null hypothesis value for BF computation. Default 0.
`...`	Ignored.
`ci_level`	Numeric scalar strictly between 0 and 1; the mass of the equal-tailed credible interval reported in `ci_lower` and `ci_upper`. Default `0.95`. Must be supplied by name.

Value

An object of class `"summary.bvarnet"` (a list) with elements:

table Data frame from `extract_param()`.

family Model family.

p Number of outcome variables.

K AR order.

n Number of observations.

rhat_max Maximum Rhat across all parameters.

n_divergences Total divergent transitions.

ci_level Credible-interval mass used for the table.

Index

∗ **datasets**
studentlife, 22

bf_table, 2, *7*, *20*
bvar, 3, *11, 12*, *14, 15*, *20*, *22*
bvar(), *9*
bvarnet_clear_model_cache, 7
bvarnet_clear_model_cache(), *8, 9*
bvarnet_model_cache_dir, 8
bvarnet_model_cache_dir(), *8, 9*
bvarnet_setup_models, *7*, 9
bvarnet_setup_models(), *8*

compare_to_truth, 10

extract_draws, 11
extract_network_matrix, 12
extract_param, *11*, 13, *22*
extract_random_effects, 13
extract_temporal, 14

format.bvarnet_prior, 15

get_default_priors, 16

print.bvarnet, 16
print.bvarnet_prior, 17
print.bvarnet_priors, 17
print.summary.bvarnet, 18
prior, 18

set_priors, *3*, 19
sim_var, 20
studentlife, 22
summary.bvarnet, *18*, 22