

Package ‘BayesianDisaggregation’

June 19, 2026

Title Evidence-Based Bayesian Disaggregation of Aggregate Indices

Version 0.2.1

Depends R (>= 4.1.0)

Description Disaggregates an observed aggregate price index into sectoral components with a Bayesian state-space model in which the aggregate enters as a genuine observation density rather than as a renormalization identity. A random-walk-with-drift transition in log space (with partial pooling on the drift and the innovation scale) and an estimable cross-sectional concentration produce posterior draws of the sectoral indices with credible intervals, suitable as multiple-imputation input for downstream dynamic models. The Hamiltonian Monte Carlo engine follows Stan (Carpenter et al., 2017) <[doi:10.18637/jss.v076.i01](https://doi.org/10.18637/jss.v076.i01)>; model comparison uses Pareto Smoothed Importance Sampling Leave-One-Out cross-validation (Vehtari, Gelman and Gabry, 2017) <[doi:10.1007/s11222-016-9696-4](https://doi.org/10.1007/s11222-016-9696-4)>. A closed-form linear-Gaussian Kalman/RTS smoother provides an exact, MCMC-free Bayesian alternative for the same aggregate evidence.

License MIT + file LICENSE

Encoding UTF-8

Imports readxl, dplyr, tidyr, stringr, magrittr, stats, parallel

Suggests cmdstanr, rstan (>= 2.21.0), posterior, loo (>= 2.5.0), knitr, rmarkdown, ggplot2, readr, testthat (>= 3.0.0)

Additional_repositories <https://mc-stan.org/r-packages/>

VignetteBuilder knitr

NeedsCompilation no

Author José Mauricio Gómez Julián [aut, cre] (ORCID:
<<https://orcid.org/0009-0000-2412-3150>>)

Maintainer José Mauricio Gómez Julián <isadore.nabi@pm.me>

Config/roxygen2/version 8.0.0

Repository CRAN

Date/Publication 2026-06-18 22:10:07 UTC

Contents

align_disagg_inputs	2
disaggregate_conjugate	3
disaggregate_from_files	4
disaggregate_statespace	5
disagg_default_priors	7
disagg_stan_code	7
log_enable	8
read_cpi	8
read_weights_matrix	9
simulate_disagg	10
Index	12

align_disagg_inputs	<i>Align CPI and VAB weights on their common years</i>
---------------------	--

Description

Reads the CPI and the weights matrix, intersects their years, and returns the aligned aggregate vector and weight matrix ready for the disaggregation engines. Both engines (state-space and conjugate) consume this output.

Usage

```
align_disagg_inputs(path_cpi, path_weights)
```

Arguments

- path_cpi Path to the CPI Excel file (see [read_cpi](#)).
- path_weights Path to the VAB-weights Excel file (see [read_weights_matrix](#)).

Value

A list with cpi (numeric, length T), W ($T \times K$, rows sum to 1), years, industries.

See Also

[disaggregate_from_files](#)

disaggregate_conjugate

Conjugate (closed-form) disaggregation baseline

Description

Exact linear-Gaussian state-space posterior (Kalman/RTS smoother) for the sectoral price-index levels $\varphi_{t,k}$ given the aggregate index and the VAB weights. Optionally returns joint posterior draws via the Durbin-Koopman simulation smoother (so the draws can also feed `bayesianOU::fit_ou_nested_mi`), and a pointwise Gaussian log-likelihood.

Usage

```
disaggregate_conjugate(
  cpi,
  W,
  years = NULL,
  industries = NULL,
  q_frac = 0.1,
  r_frac = 0.05,
  p0_frac = 0.3,
  n_draws = 0L,
  seed = 1234L
)
```

Arguments

<code>cpi</code>	Numeric vector (length T); observed aggregate index (levels).
<code>W</code>	Numeric matrix ($T \times K$); VAB weights (rows sum to 1).
<code>years, industries</code>	Optional period and sector labels.
<code>q_frac</code>	Random-walk innovation sd as a fraction of <code>sd(cpi)</code> (state noise). Default 0.10.
<code>r_frac</code>	Observation sd as a fraction of <code>sd(cpi)</code> . Default 0.05.
<code>p0_frac</code>	Initial cross-sectional sd as a fraction of <code>cpi[1]</code> . Default 0.30.
<code>n_draws</code>	Integer; number of joint posterior draws (simulation smoother). 0 (default) returns only the smoothed mean and bands.
<code>seed</code>	Integer RNG seed (used only when <code>n_draws > 0</code>).

Value

An object of class "disagg_conjugate": a list with `phi_summary` (median = smoothed mean, q2.5/q97.5 from the marginal Gaussian), `agg_summary`, `loglik` (total Gaussian log-likelihood of `cpi`), `phi_draws` ($[T, K, n_draws]$ or NULL), `years`, `industries`, `config`.

See Also

[disaggregate_statespace](#) (canonical engine).

Examples

```
sim <- simulate_disagg(T = 25, K = 4, seed = 7)
bl <- disaggregate_conjugate(sim$cpi, sim$W)
dim(bl$phi_summary$median)
```

disaggregate_from_files

Evidence-based disaggregation directly from Excel files

Description

Thin convenience wrapper: reads and aligns the CPI and VAB-weight files ([align_disagg_inputs](#)) and runs the canonical state-space engine ([disaggregate_statespace](#)).

Usage

```
disaggregate_from_files(path_cpi, path_weights, ...)
```

Arguments

path_cpi	Path to the CPI Excel file (index levels, re-indexed to the same base as the production prices; see the package vignette and the data note on <code>convert_to_index</code>).
path_weights	Path to the VAB-weights Excel file.
...	Passed to disaggregate_statespace (sampler controls, priors, student_obs, ...).

Value

A "disagg_statespace" object.

See Also

[disaggregate_statespace](#), [align_disagg_inputs](#)

Examples

```
## Not run:
cpi_file <- system.file("extdata", "CPI.xlsx", package = "BayesianDisaggregation")
w_file <- system.file("extdata", "WEIGHTS.xlsx", package = "BayesianDisaggregation")
fit <- disaggregate_from_files(cpi_file, w_file, chains = 2, iter = 800)

## End(Not run)
```

disaggregate_statespace

Evidence-based Bayesian disaggregation (state-space; canonical engine)

Description

Disaggregates an observed aggregate index (CPI) into K latent sectoral price indices $\varphi_{t,k}$ with a Bayesian state-space model in which the aggregate enters as a genuine observation density (not a renormalization identity). The model couples a random-walk-with-drift transition in $\log \varphi$ (partial pooling on the drift and the innovation scale), an estimable cross-sectional concentration, and a Student-t (or Gaussian) observation $cpi_t \mid \varphi \sim \text{Student-t}(\nu, \sum_k W_{t,k} \varphi_{t,k}, \sigma)$. See `vignette("evidence-based-disaggregation")`.

Usage

```
disaggregate_statespace(
  cpi,
  W,
  years = NULL,
  industries = NULL,
  student_obs = TRUE,
  priors = NULL,
  chains = 4L,
  iter = 2000L,
  warmup = 1000L,
  thin = 1L,
  cores = NULL,
  adapt_delta = 0.95,
  max_treedepth = 12L,
  seed = 1234L,
  init = 0.5,
  keep_fit = TRUE,
  verbose = FALSE
)
```

Arguments

<code>cpi</code>	Numeric vector (length T); the observed aggregate index in levels (e.g. CPI re-indexed to a common base). Strictly positive.
<code>W</code>	Numeric matrix ($T \times K$); the (known) VAB aggregation weights, rows summing to 1 (small deviations are renormalized). Sector columns must align with the desired output ordering.
<code>years</code>	Optional integer vector (length T); period labels.
<code>industries</code>	Optional character vector (length K); sector labels. Defaults to <code>colnames(W)</code> when present.

<code>student_obs</code>	Logical; if TRUE (default) the observation is Student-t (robust to aggregate outliers), otherwise Gaussian.
<code>priors</code>	Optional named list overriding disagg_default_priors .
<code>chains, iter, warmup, thin</code>	Sampler controls (HMC/NUTS). Defaults 4 / 2000 / 1000 / 1.
<code>cores</code>	Integer; parallel chains. Default <code>min(chains, detectCores())</code> .
<code>adapt_delta, max_treedepth</code>	NUTS tuning. Defaults 0.95 / 12.
<code>seed</code>	Integer RNG seed. Default 1234.
<code>init</code>	Sampler init; a numeric scalar is an init radius (<code>cmdstanr</code>) or is translated to <code>init_r</code> (<code>rstan</code>). Default 0.5.
<code>keep_fit</code>	Logical; keep the raw Stan fit object in the result. Default TRUE (needed to draw further quantities or run LOO).
<code>verbose</code>	Logical; print progress. Default FALSE.

Details

The returned posterior draws of φ (a $[T, K, \text{draws}]$ array) are exactly the multiple-imputation input consumed by `bayesianOU::fit_ou_nested_mi()`, propagating the disaggregation uncertainty into the downstream nested-OU analysis (Rubin's rule).

Value

An object of class "disagg_statespace": a list with

phi_draws $[T, K, \text{draws}]$ numeric array of posterior draws of φ (the multiple-imputation input for the nested OU).

phi_summary List of $T \times K$ matrices median, q2.5, q97.5 (credible bands per sector and period).

agg_summary $T \times 3$ matrix: posterior median and 95% band of the fitted aggregate $\sum_k W \varphi$ (against which `cpi` is the evidence).

years, industries Period and sector labels.

diagnostics `rhat_max`, `divergences`.

stan_fit The Stan fit (if `keep_fit`).

config Sampler/prior configuration and T, K .

See Also

[disaggregate_conjugate](#) (closed-form Bayesian baseline), [disaggregate_from_files](#), [simulate_disagg](#).

Examples

```
## Not run:
set.seed(1)
sim <- simulate_disagg(T = 30, K = 4)
fit <- disaggregate_statespace(sim$cpi, sim$W, chains = 2, iter = 800)
dim(fit$phi_draws)      # T x K x draws

## End(Not run)
```

disagg_default_priors *Default prior scales for the state-space disaggregation*

Description

Weakly-informative scales on the raw index level, derived from the observed aggregate so the model is location/scale aware. Override any field by passing a (partial) named list to `disaggregate_statespace(priors = ...)`.

Usage

```
disagg_default_priors(cpi)
```

Arguments

cpi Numeric vector; the observed aggregate index (level).

Value

Named list of prior scales (see [disaggregate_statespace](#)).

disagg_stan_code *Stan code for the canonical disaggregation model*

Description

Returns the complete Stan code of the evidence-based state-space model read from the canonical file (single source of truth; no embedded duplicate).

Usage

```
disagg_stan_code()
```

Value

Character string with the Stan model code.

Examples

```
code <- disagg_stan_code()
cat(substr(code, 1, 200))
```

log_enable	<i>Enable logging at a specific level</i>
------------	---

Description

Sets the package-wide logging verbosity.

Usage

```
log_enable(level = "INFO")
```

Arguments

level	Character scalar. One of "TRACE", "DEBUG", "INFO", "WARN", "ERROR".
-------	---

Value

(Invisibly) the level set.

read_cpi	<i>Read CPI data from an Excel file</i>
----------	---

Description

Loads and normalizes a CPI time series from an Excel worksheet. The function detects the date/year column and the CPI/value column by pattern-matching on lower-cased header names, parses localized numerics (via `to_num_commas()`), collapses duplicate years by averaging, and returns a clean, sorted data frame.

Usage

```
read_cpi(path_cpi)
```

Arguments

path_cpi	Character path to the CPI Excel file.
----------	---------------------------------------

Details

Column detection. Headers are lower-cased and matched with:

- Date/year: patterns "date|fecha|year|anio|ano".
- CPI/value: patterns "cpi|indice|price".

If either column cannot be identified, the function errors.

Cleaning.

- Year is extracted as the first 4 digits of the date-like column.
- CPI is parsed with `to_num_commas()` (handles commas/thousands).
- NA rows are dropped; duplicates in Year are averaged.
- Output is sorted by Year ascending.

Value

A data.frame with two columns:

- Year (integer)
- CPI (numeric)

See Also

[read_weights_matrix](#), [align_disagg_inputs](#)

Examples

```
cpi_file <- system.file("extdata", "CPI.xlsx", package = "BayesianDisaggregation")
if (nzchar(cpi_file)) {
  df <- read_cpi(cpi_file)
  head(df)
}
```

read_weights_matrix	<i>Read a weights matrix from an Excel file</i>
---------------------	---

Description

Loads a sector-by-year weight table, normalizes weights to the simplex per year, and returns a list with the $T \times K$ prior matrix P, the sector names, and the year vector. The first column is assumed to contain sector names (renamed to Industry); all other columns are treated as years.

Usage

```
read_weights_matrix(path_weights)
```

Arguments

`path_weights` Character path to the weights Excel file.

Details

Expected layout. One sheet with:

- First column: sector names (any header; renamed to Industry).
- Remaining columns: years; the function extracts a 4-digit year from each header using `stringr::str_extract(Year, "\\d{4}")`.

Values are parsed with `to_num_commas()`, missing rows are dropped, and weights are normalized within each year to sum to 1. Any absent (sector, year) entry becomes 0 when pivoting wide. Finally, rows are re-normalized with `row_norm1()` for numerical safety.

Safeguards.

- Rows with all-missing/zero after parsing are dropped by the filters.
- If no valid year columns are found, the function errors.

Value

A list with:

`P` $T \times K$ numeric matrix of prior weights (rows sum to 1).

`industries` Character vector of sector names (length K).

`years` Integer vector of years (length T).

See Also

[read_cpi](#), [align_disagg_inputs](#)

Examples

```
w_file <- system.file("extdata", "WEIGHTS.xlsx", package = "BayesianDisaggregation")
if (nzchar(w_file)) {
  w <- read_weights_matrix(w_file)
  stopifnot(is.matrix(w$P), all(abs(rowSums(w$P) - 1) < 1e-8))
  str(w)
}
```

simulate_disagg

Simulate from the state-space disaggregation DGP

Description

Generates a synthetic aggregate index `cpi`, the (known) VAB weights `W`, and the latent sectoral price-index paths `phi_true` from the same data-generating process as [disaggregate_statespace](#). The innovation scale is kept modest so the log random walk stays in a numerically stable region over the simulated horizon (the same care taken in the sibling OU simulator).

Usage

```
simulate_disagg(
  T = 40L,
  K = 5L,
  phi1_center = 100,
  omega_struct = 0.3,
  delta_mu = 0.02,
  delta_sigma = 0.01,
  tau_mu = 0.04,
  tau_sigma = 0.3,
  sigma_cpi = 1,
  nu = Inf,
  seed = 1234L
)
```

Arguments

T	Integer; number of periods.
K	Integer; number of sectors.
phi1_center	Numeric; central level of the initial cross-section.
omega_struct	Numeric; cross-sectional log-level dispersion at $t = 1$.
delta_mu, delta_sigma	Common drift and its cross-sector dispersion.
tau_mu, tau_sigma	Geometric mean innovation scale and log-dispersion (so $\tau_k = \tau_{mu} \exp(\tau_{sigma} z_k)$).
sigma_cpi	Observation noise scale on the aggregate.
nu	Student-t degrees of freedom of the observation (Inf = Gaussian).
seed	Integer RNG seed.

Value

A list with cpi (length T), W ($T \times K$, rows sum to 1), phi_true ($T \times K$), agg_true (length T), and params (the true scalar/vector parameters).

Examples

```
sim <- simulate_disagg(T = 20, K = 3, seed = 42)
str(sim$params)
```

Index

`align_disagg_inputs`, [2](#), [4](#), [9](#), [10](#)

`disagg_default_priors`, [6](#), [7](#)

`disagg_stan_code`, [7](#)

`disaggregate_conjugate`, [3](#), [6](#)

`disaggregate_from_files`, [2](#), [4](#), [6](#)

`disaggregate_statespace`, [4](#), [5](#), [7](#), [10](#)

`log_enable`, [8](#)

`read_cpi`, [2](#), [8](#), [10](#)

`read_weights_matrix`, [2](#), [9](#), [9](#)

`simulate_disagg`, [6](#), [10](#)