Package: DynCount (via r-universe)

July 14, 2026

Title Bayesian Dynamic Models for Poisson and Binomial Time Series

Version 0.1.0

Description Fits Bayesian state-space models for non-Gaussian time series using a latent log-rate (Poisson) or latent logit (binomial) formulation. The latent trajectory follows a first-order random walk or a stationary AR(1) process, sampled by Metropolis-within-Gibbs using the implied Gaussian Markov random field (GMRF) full conditionals. Four innovation structures are supported for the latent increments: constant-variance Gaussian, Student-t, a finite scale mixture of normals, and stochastic volatility. Both families support time-constant zero inflation. The package provides simulation, fitting, forecasting, summary and plotting tools. It implements and extends the methodology of Zens and Bijak (2026) <doi:10.1214/26-AOAS2171>.

License MIT + file LICENSE

Language en-GB

Encoding UTF-8

Depends R (>= 3.5.0)

Imports stats, graphics, grDevices, utils

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

VignetteBuilder knitr

LazyData true

RoxygenNote 7.3.1

Config/testthat/edition 3

NeedsCompilation no

Author Gregor Zens [aut, cre]

Maintainer Gregor Zens <zens@iiasa.ac.at>

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

Date/Publication 2026-07-14 18:33:59 UTC

RemoteUrl https://github.com/cran/DynCount

RemoteRef HEAD

RemoteSha bbeb2e8206c18b0d6684da2806d13da6e223b188

Contents

`dynamic_prior` *Specify priors for a dynamic count / binomial model*

Description

Builds the prior hyperparameters used by `fit_dynamic_model()`. Called with no arguments it returns weakly informative defaults.

Usage

```
dynamic_prior(
  var_shape = 2.5,
  var_rate = 0.5,
  df_min = 3,
  df_mean_excess = 6,
  mix_components = 2,
  mix_concentration = 1,
  sv_prior = NULL,
  zi_open_a = 1,
  zi_open_b = 1,
  ar_rho_mean = 0,
```

```
  ar_rho_sd = 1,
  mu_mean = 0,
  mu_sd = 1,
  init_mean = 0,
  init_var = 100
)
```

Arguments

`var_shape`	Shape of the inverse-gamma prior on the innovation variance/scale. Default `2.5`.
`var_rate`	Rate of the inverse-gamma prior on the innovation variance/scale. Default `0.5`.
`df_min`	Lower bound for the Student-t degrees of freedom. Default `3`.
`df_mean_excess`	Prior mean of $\nu -$ `df_min`. Default `6`.
`mix_components`	Number of components in the scale-mixture innovation structure. Default `2`.
`mix_concentration`	Symmetric Dirichlet concentration for the mixture weights. Default `1`.
`sv_prior`	Optional **stochvol** prior specification for the stochastic-volatility innovation structure. Default `NULL`.
`zi_open_a`, `zi_open_b`	Beta prior parameters for the gate-open probability in zero-inflated models. Default `1` and `1`.
`ar_rho_mean`, `ar_rho_sd`	Mean and standard deviation of the Gaussian prior on the AR(1) coefficient ρ (used only when `latent_dynamics = "ar1"`). The conjugate Gibbs update truncates ρ to the stationary region $\rho \in (-1, 1)$. Defaults `0` and `1`.
`mu_mean`, `mu_sd`	Mean and standard deviation of the Gaussian prior on the drift/intercept μ (used only when `include_mu = TRUE`). Defaults `0` and `1`.
`init_mean`, `init_var`	Mean and **variance** of the Gaussian prior on the initial latent state, used under both random-walk and AR(1) dynamics. Defaults `0` and `100` (a diffuse $N(0, 100)$).

Details

The model places a GMRF latent process on the series z_t (a log-rate for the Poisson family, a logit for the binomial family): either a first-order random walk (`latent_dynamics = "rw"`) or an AR(1) process (`latent_dynamics = "ar1"`). The increments $\varepsilon_t = z_t - \mu - \rho\, z_{t-1}$ (with $\rho = 1$ for the random walk and $\mu = 0$ unless a drift/intercept is included) are given one of four innovation distributions, each governed by some of the priors below.

Innovation variance (all structures). The baseline increment variance σ^2 (or, for the `"t"`/`"mixture"` structures, the overall scale) has an inverse-gamma prior

$$\sigma^2 \sim \mathrm{InvGamma}(\texttt{var\_shape}, \texttt{var\_rate}).$$

Smaller `var_rate` gives a smoother (more strongly shrunk) latent path.

Student-t degrees of freedom (`innovations = "t"`). The degrees of freedom are modelled as $\nu = \texttt{df\_min} + E$, where $E \sim \mathrm{Exp}(\mathrm{rate} = 1/\texttt{df\_mean\_excess})$. Thus $\nu \geq \texttt{df\_min}$ and its prior mean is $\texttt{df\_min} + \texttt{df\_mean\_excess}$. Large ν approaches the Gaussian case.

Scale mixture (`innovations = "mixture"`). The mixture uses `mix_components` variance components, each with an $\mathrm{InvGamma}(\texttt{var\_shape}, \texttt{var\_rate})$ prior, and symmetric Dirichlet weights with concentration `mix_concentration`.

Stochastic volatility (`innovations = "sv"`). The log-variance AR(1) priors are delegated to **stochvol**. Pass a prior specification created with `stochvol::specify_priors()` via `sv_prior`, or leave it `NULL` to use the **stochvol** defaults.

Zero inflation (`zeros = "inflated"`). The probability that the observation "gate" is open (i.e. that a zero is an ordinary sampling zero rather than a structural zero) has a $\mathrm{Beta}(\texttt{zi\_open\_a}, \texttt{zi\_open\_b})$ prior. The default `Beta(1, 1)` is uniform.

AR(1) coefficient (`latent_dynamics = "ar1"`). When the latent state follows $z_t = \mu + \rho\, z_{t-1} + \varepsilon_t$, the coefficient ρ is given a Gaussian prior $\rho \sim \mathrm{N}(\texttt{ar\_rho\_mean}, \texttt{ar\_rho\_sd}^2)$, truncated to the stationary region $\rho \in (-1, 1)$, and is sampled jointly with μ by an exact conjugate Gibbs draw (see DynCount-package). AR(1) always carries an intercept (`include_mu = TRUE`). Under `latent_dynamics = "rw"` the coefficient is fixed at $\rho = 1$ and this prior is unused.

Drift / intercept (`include_mu = TRUE`). A scalar μ in the state equation $z_t = \mu + \rho\, z_{t-1} + \varepsilon_t$: a *drift* under the random walk ($\rho = 1$) and an *intercept* under AR(1) (where it is always enabled). It is given a Gaussian prior $\mu \sim \mathrm{N}(\texttt{mu\_mean}, \texttt{mu\_sd}^2)$. With `include_mu = FALSE` (random walk only), $\mu = 0$ and is not sampled.

Initial state. A proper, fixed $\mathrm{N}(\texttt{init\_mean}, \texttt{init\_var})$ prior (default $N(0, 100)$) anchors the otherwise-improper GMRF on the first latent state, under both `"rw"` and `"ar1"` dynamics; keeping it free of (μ, ρ) is what makes their update conjugate (see DynCount-package). With the diffuse default the initial state is effectively determined by the data.

Value

An object of class `"dynamic_prior"`: a named list of hyperparameters.

See Also

`fit_dynamic_model()`

Examples

```
# Defaults
dynamic_prior()

# Smoother latent path and heavier-tailed t innovations
dynamic_prior(var_rate = 0.1, df_min = 2, df_mean_excess = 3)
```

`fit_dynamic_model` *Fit a Bayesian dynamic count / binomial time-series model*

Description

Fits a GMRF state-space model in which a latent trajectory z_t evolves as either a first-order random walk (`latent_dynamics = "rw"`, the default) or a stationary AR(1) process (`latent_dynamics = "ar1"`), and the observations are linked to it through a Poisson (log link) or binomial (logit link) observation model. See DynCount-package for an overview.

Usage

```
fit_dynamic_model(
  y,
  family = c("poisson", "binomial"),
  trials = NULL,
  innovations = c("gaussian", "t", "mixture", "sv"),
  latent_dynamics = c("rw", "ar1"),
  include_mu = FALSE,
  zeros = c("none", "inflated", "missing"),
  zero_inflation = FALSE,
  prior = dynamic_prior(),
  nsave = 4000,
  nburn = 1000,
  thin = 1,
  horizon = 0L,
  forecast_trials = NULL,
  forecast_offset = NULL,
  offset = NULL,
  verbose = FALSE,
  seed = NULL
)
```

Arguments

`y`	Numeric vector of non-negative integer observations. For the Poisson family these are counts; for the binomial family these are the numbers of successes.
`family`	Observation model, `"poisson"` (default) or `"binomial"`.
`trials`	For `family = "binomial"`, the number of trials. Either a single number (recycled) or a vector the same length as `y`. Each `y` must not exceed its number of trials. Ignored for the Poisson family.
`innovations`	Distribution of the latent increments, one of `"gaussian"` (default), `"t"`, `"mixture"`, `"sv"`. See DynCount-package for details. `"sv"` requires the **stochvol** package.
`latent_dynamics`	Latent state evolution: `"rw"` (default; a first-order random walk, $\rho = 1$ fixed) or `"ar1"` (a stationary AR(1) with ρ sampled on $(-1, 1)$). `"ar1"` always includes an intercept (`include_mu` is forced to `TRUE`).

include_mu — Logical; include a scalar μ in the state equation $z_t = \mu + \rho z_{t-1} + \varepsilon_t$. It is a **drift** under latent_dynamics = "rw" ($\rho = 1$) and an **intercept** under "ar1". With FALSE (default) $\mu = 0$ and is not sampled; with TRUE it has the Gaussian prior in dynamic_prior(). When latent_dynamics = "ar1" this is forced to TRUE regardless of the value supplied: the intercept gives the process a non-zero stationary mean $\mu/(1-\rho)$, without which the zero-mean stationary assumption is rarely appropriate for a log-rate/logit series.

zeros — Zero handling for both families: "none" (default), "inflated" (time-constant zero inflation; see structural_zero_prob()), or "missing" (observed zeros treated as missing data). zero_inflation = TRUE is shorthand for zeros = "inflated". See Details.

zero_inflation — Logical convenience flag. If TRUE, sets zeros = "inflated". Ignored if zeros is supplied explicitly.

prior — A dynamic_prior() object giving the prior hyperparameters.

nsave — Number of posterior draws to keep. Default 4000.

nburn — Number of burn-in iterations. Default 1000.

thin — Thinning interval: one draw is kept every thin iterations, so the sampler runs nburn + nsave * thin iterations in total and retains nsave draws. Default 1.

horizon — Forecast horizon H (a non-negative integer). When H >= 1, forecasts are produced inside the sampler and stored for retrieval with forecast.dynamic_fit(); when H = 0 (default) no forecast is produced.

forecast_trials — For the binomial family, the number of trials in the forecast period (length 1, recycled, or length horizon). If omitted it defaults to the last observed trials count. Ignored when horizon = 0.

forecast_offset — Known per-period offset over the forecast horizon (length 1, recycled, or length horizon). Defaults to 0. Ignored when horizon = 0.

offset — Optional known per-observation offset on the linear-predictor scale (length 1 or n). For the Poisson family this is a log-exposure term, so the mean is $\exp(\text{offset}_t + z_t)$; for the binomial family it shifts the logit. Default NULL (no offset).

verbose — Logical; print a progress bar. Default FALSE.

seed — Optional integer random seed for reproducibility.

Details

Zero handling. Under zeros = "inflated" a latent gate decides, for each observed zero, whether it is structural (gate closed) or an ordinary sampling zero produced by the Poisson/binomial process (gate open); the gate-open probability is a single time-constant parameter. See structural_zero_prob().

Forecasting. Forecasts are produced *during* sampling: set horizon = H (a positive integer) and H extra latent states are appended to the state vector and sampled as missing values inside the MCMC. A response is drawn from the observation model at each sampled forecast state. The stored forecast draws are retrieved with forecast.dynamic_fit(). With the default horizon = 0 no forecast is produced.

Value

An object of class "dynamic_fit": a list with three elements, each holding a distinct kind of content.

`draws` Posterior draws only – a list of matrices/vectors with one row (or element) per stored draw:

- `z`, `sig2` Latent states and increment variances.
- `fitted`, `yrep` **Unconditional** fitted means and posterior predictive replicates; under `zeros = "inflated"` they include the zero-inflation gate.
- `fitted_open`, `yrep_open` Their **conditional-on-gate-open** counterparts: the latent-implied mean, and a replicate drawn straight from the observation model.
- `gate`, `pi_open` Zero-inflation gate indicators and gate-open probability. For fits without zero inflation these are stored as placeholders: `gate` is the constant indicator `y > 0` and `pi_open` is 1.
- `rho`, `mu`, `nu`, `innov_var` AR(1) coefficient, drift/intercept, Student-t degrees of freedom, and a representative innovation variance whose definition depends on `innovations`: the estimated constant increment variance σ^2 for `"gaussian"`; the marginal increment variance for `"t"` ($\sigma^2\nu/(\nu-2)$) and `"mixture"` ($\sigma^2 \sum_h \eta_h \sigma_h^2$); and the average of the per-increment variances $\exp(h_t)$ over the series for `"sv"`. For `"t"`, if a draw has $\nu \leq 2$ (possible only when `df_min <= 2`), the undefined variance factor is replaced by the convention 3.
- `forecast_z`, `forecast_y` Latent and response forecasts when `horizon >= 1` (`NULL` otherwise); `forecast_y` is unconditional, i.e. includes the gate.

Use `yrep`, not `yrep_open`, for posterior predictive checks; without zero inflation the conditional and unconditional pairs are identical. See `predict.dynamic_fit()`.

`data` The observed inputs: `y`, `trials`, the series length `n`, and the resolved per-observation `offset`.

`spec` The model and MCMC specification: `family`, `innovations`, `latent_dynamics`, `include_mu`, `zeros`, `prior`, `nsave`, `nburn`, `thin`, `horizon`, and the fixed forecast-period inputs `forecast_offset` / `forecast_trials` (`NULL` when not applicable).

See Also

`dynamic_prior()`, `forecast.dynamic_fit()`, `plot_fitted()`, `structural_zero_prob()`

Examples

```
set.seed(1)
sim <- simulate_dynamic_poisson(n = 60, sigma = 0.2, log_rate0 = 2)
fit <- fit_dynamic_model(sim$y, family = "poisson", nsave = 300, nburn = 200)
summary(fit)
```

forecast *Generic forecast function*

Description

Generic for extracting forecasts from a fitted model. See `forecast.dynamic_fit()` for the method for `"dynamic_fit"` objects.

Usage

```
forecast(object, ...)
```

Arguments

`object`	A fitted model object.
`...`	Passed to methods.

Value

The value returned by the method dispatched on `object`; for `"dynamic_fit"` objects a `"dynamic_forecast"` object (see `forecast.dynamic_fit()`).

forecast.dynamic_fit *Forecast a fitted dynamic model*

Description

Returns the forecast draws that were produced during MCMC sampling. Forecasts are generated inside the sampler when the model is fitted with `horizon = H >= 1`. This method extracts and summarises those stored draws.

Usage

```
## S3 method for class 'dynamic_fit'
forecast(object, probs = c(0.025, 0.5, 0.975), ...)
```

Arguments

`object`	A `"dynamic_fit"` object that was fitted with `horizon >= 1`.
`probs`	Quantile probabilities for the summary. Default `c(0.025, 0.5, 0.975)`.
`...`	Unused.

Details

Zero inflation. For fits with `zeros = "inflated"` the response forecast is **unconditional**: each draw includes the zero-inflation gate, so structural zeros are reproduced and the draws are directly comparable to future observations (like the in-sample `yrep`, unlike `yrep_open`). The latent forecast summary in `$latent` is gate-free by construction.

Value

An object of class `"dynamic_forecast"`: a list with `summary` (one row per horizon $1, \ldots, H$, on the response scale), `latent` (summary of the forecast latent path), `draws` (the predictive draws, draws x `H`), `final` (the single-row summary of the final `H`-step-ahead forecast), `final_draws` (the predictive draws at horizon `H`), and the forecast horizon `horizon`.

Examples

```
sim <- simulate_dynamic_poisson(60, 0.2, 2, seed = 1)
fit <- fit_dynamic_model(sim$y, nsave = 300, nburn = 200, horizon = 8)
fc <- forecast(fit)
fc$summary
```

---

med_weekly *Weekly Mediterranean crossings (Mediterranean example)*

Description

A longer weekly count series of irregular sea crossings on the Mediterranean route, beginning in autumn 2015. The counts are larger in magnitude with fewer zeros than uk_weekly, which makes the series useful for the plain Poisson model and the robust (Student-t) and stochastic-volatility innovation structures. The series is used in the irregular-migration application of Zens and Bijak (2026).

Usage

```
med_weekly
```

Format

A data frame with 494 rows and 2 variables:

week ISO week label, e.g. `"2015-W40"`.

count Non-negative integer count of weekly crossings.

Source

Weekly aggregates of detected irregular Mediterranean crossings, compiled from operational/agency records as described in Zens and Bijak (2026), doi:10.1214/26AOAS2171.

References

Zens, G. and Bijak, J. (2026). Dynamic Count Models with Flexible Innovation Processes for Irregular Maritime Migration. *The Annals of Applied Statistics*. doi:10.1214/26AOAS2171.

Examples

```
data(med_weekly)
summary(med_weekly$count)
```

`plot.dynamic_fit` *Plot method for fitted dynamic models*

Description

A convenience wrapper that dispatches to the dedicated plotting functions.

Usage

```
## S3 method for class 'dynamic_fit'
plot(x, which = c("fitted", "latent", "forecast", "zeros"), ...)
```

Arguments

`x`	A `"dynamic_fit"` object.
`which`	One of `"fitted"` (default), `"latent"`, `"forecast"`, `"zeros"`.
`...`	Passed to the underlying plotting function.

Value

Invisibly, the result of the underlying plotting function.

`plot_fitted` *Plot observed versus fitted values*

Description

Observed series with the posterior median fitted mean and a credible band on the response scale.

Usage

```
plot_fitted(object, level = 0.95, ...)
```

Arguments

object	A "dynamic_fit" object.
level	Credible level. Default 0.95.
...	Passed to graphics::plot().

Value

Invisibly, the fitted summary.

Examples

```
sim <- simulate_dynamic_poisson(60, 0.2, 2, seed = 1)
fit <- fit_dynamic_model(sim$y, nsave = 300, nburn = 200)
plot_fitted(fit)
```

---

plot_forecast *Plot observed history, fitted values and forecast with uncertainty*

Description

Shows the observed series together with the in-sample fitted mean (median and credible band) and the forecast (median and credible band) over the forecast horizon. The forecast draws are those produced during sampling, so the model must have been fitted with horizon >= 1 (see forecast.dynamic_fit()).

Usage

```
plot_forecast(object, level = 0.95, ...)
```

Arguments

object	A "dynamic_fit" object fitted with horizon >= 1.
level	Credible level. Default 0.95.
...	Passed to graphics::plot().

Value

Invisibly, the "dynamic_forecast" object.

Examples

```
sim <- simulate_dynamic_poisson(60, 0.2, 2, seed = 1)
fit <- fit_dynamic_model(sim$y, nsave = 300, nburn = 200, horizon = 12)
plot_forecast(fit)
```

plot_latent *Plot the fitted latent trajectory*

Description

Shows the posterior median of the latent state (log-rate for the Poisson family, logit for the binomial family) with a credible band.

Usage

```
plot_latent(object, level = 0.95, ...)
```

Arguments

object	A "dynamic_fit" object.
level	Credible level for the band. Default 0.95.
...	Passed to graphics::plot().

Value

Invisibly, the summary data frame used for plotting.

Examples

```
sim <- simulate_dynamic_poisson(60, 0.2, 2, seed = 1)
fit <- fit_dynamic_model(sim$y, nsave = 300, nburn = 200)
plot_latent(fit)
```

plot_zero_inflation *Plot zero-inflation diagnostics*

Description

Bar chart of the posterior probability that each observed zero is a structural zero. Requires a model fitted with zeros = "inflated".

Usage

```
plot_zero_inflation(object, ...)
```

Arguments

object	A "dynamic_fit" object.
...	Passed to graphics::barplot().

Value

Invisibly, the `structural_zero_prob()` data frame.

Examples

```
sim <- simulate_dynamic_poisson(80, 0.2, 2, zero_inflation = 0.3, seed = 1)
fit <- fit_dynamic_model(sim$y, zero_inflation = TRUE, nsave = 300, nburn = 200)
plot_zero_inflation(fit)
```

`predict.dynamic_fit` *In-sample fitted values and posterior predictive replicates*

Description

In-sample fitted values and posterior predictive replicates

Usage

```
## S3 method for class 'dynamic_fit'
predict(
  object,
  type = c("mean", "response"),
  conditional = FALSE,
  probs = c(0.025, 0.5, 0.975),
  ...
)
```

Arguments

`object`	A `"dynamic_fit"` object.
`type`	`"mean"` (default) returns the posterior of the mean of `y`; `"response"` returns posterior predictive replicates of `y`.
`conditional`	Logical. Leave `FALSE` (the default) for anything compared against observed data – posterior predictive checks, calibration – and set `TRUE` only to inspect the latent intensity process. With `FALSE`, means and replicates come from the full zero-inflated model (the gate is included, so structural zeros are reproduced); with `TRUE`, they are conditional on the gate being open, i.e. drawn straight from the Poisson/binomial observation model, and show systematically too few zeros under zero inflation. Without zero inflation the two versions are identical. See DynCount-package for the gate notation.
`probs`	Quantile probabilities for the summary. Default `c(0.025, 0.5, 0.975)`.
`...`	Unused.

Value

A list with `summary` (a data frame, one row per observation) and `draws` (the underlying draws matrix, draws x time).

Examples

```
sim <- simulate_dynamic_poisson(40, 0.2, 2, seed = 1)
fit <- fit_dynamic_model(sim$y, nsave = 200, nburn = 100)
head(predict(fit)$summary)
```

`simulate_dynamic_binomial` *Simulate a binomial dynamic series*

Description

Generates a latent logit process (random walk or AR(1)) and binomial counts, optionally with structural (zero-inflation) zeros.

Usage

```
simulate_dynamic_binomial(
  n,
  sigma,
  trials,
  logit0 = 0,
  zero_inflation = 0,
  rho = 1,
  mu = 0,
  offset = 0,
  seed = NULL
)
```

Arguments

`n`	Number of observations.
`sigma`	Standard deviation of the latent increments (Gaussian).
`trials`	Number of trials: a single number (recycled) or a length-`n` vector.
`logit0`	Initial logit z_1. Default `0`.
`zero_inflation`	Structural-zero probability. With probability `zero_inflation` an observation is forced to a structural zero (gate closed) regardless of the binomial draw. Default `0` (no inflation).
`rho`	AR(1) coefficient of the latent process $z_t = \mu + \rho z_{t-1} + \varepsilon_t$. Default 1 (a random walk).
`mu`	Drift (random walk) / intercept (AR(1)) of the latent process. Default `0`.
`offset`	Known offset on the logit scale (length 1 or `n`); the success probability is $\mathrm{logit}^{-1}(\mathrm{offset}_t + z_t)$. Default `0`.
`seed`	Optional random seed.

Value

A list with components `y` (successes), `trials`, `logit` (latent path z_t), `prob` (`plogis(offset + logit)`), `offset`, and `structural` (logical, `TRUE` for structural zeros).

Examples

```
sim <- simulate_dynamic_binomial(n = 50, sigma = 0.15, trials = 40, seed = 1)
head(sim$y)
# with structural zeros:
zi <- simulate_dynamic_binomial(50, 0.15, trials = 40, zero_inflation = 0.2, seed = 1)
mean(zi$structural)
```

`simulate_dynamic_poisson` *Simulate a Poisson dynamic series*

Description

Generates a latent log-rate process (random walk or AR(1)) and Poisson counts, optionally with zero inflation.

Usage

```
simulate_dynamic_poisson(
  n,
  sigma,
  log_rate0 = 1,
  zero_inflation = 0,
  rho = 1,
  mu = 0,
  offset = 0,
  seed = NULL
)
```

Arguments

`n`	Number of observations.
`sigma`	Standard deviation of the latent increments (Gaussian).
`log_rate0`	Initial log-rate z_1. Default `1`.
`zero_inflation`	Probability that the gate is **closed** (i.e. the probability of a structural zero) at each time point. `0` (default) gives an ordinary Poisson series.
`rho`	AR(1) coefficient of the latent process $z_t = \mu + \rho z_{t-1} + \varepsilon_t$. Default `1` (a random walk).
`mu`	Drift (random walk) / intercept (AR(1)) of the latent process. Default `0`.
`offset`	Known log-exposure offset (length 1 or `n`); the Poisson mean is $\exp(\text{offset}_t + z_t)$. Default `0`.
`seed`	Optional random seed.

Value

A list with components `y` (observed counts), `log_rate` (the latent log-rate path z_t), `rate` (the mean `exp(offset + log_rate)`), `offset`, and `structural` (logical, `TRUE` where a structural zero was forced).

Examples

```
sim <- simulate_dynamic_poisson(n = 50, sigma = 0.2, log_rate0 = 2,
                                zero_inflation = 0.2, seed = 1)
table(sim$y == 0, sim$structural)
```

---

structural_zero_prob *Posterior probability that each observed zero is structural*

Description

For a zero-inflated fit (Poisson or binomial), returns for every observed zero the posterior probability that it is a *structural* zero (gate closed) rather than an ordinary *sampling* zero generated by the observation process. The zero inflation is time-constant: a single gate-open probability governs all observations (it does not vary over time or with covariates).

Usage

```
structural_zero_prob(object, zeros_only = TRUE)
```

Arguments

`object`	A `"dynamic_fit"` object fitted with `zeros = "inflated"`.
`zeros_only`	If `TRUE` (default), return only the rows where the observation is zero.

Details

The model introduces a latent gate indicator $v_t \in \{0, 1\}$: when the gate is open ($v_t = 1$) the count comes from the observation model (Poisson or binomial); when closed ($v_t = 0$) the count is a structural zero. The sampler stores v_t for each draw, so the structural-zero probability is

$$\Pr(\text{structural} \mid y_t = 0) = 1 - \overline{v_t},$$

the posterior mean of the gate being closed. Non-zero observations are always sampling observations and have structural probability `0`.

Value

A data frame with columns `time`, `observed`, `p_structural` (posterior probability the zero is structural) and `p_sampling` (`1 - p_structural`).

Examples

```
sim <- simulate_dynamic_poisson(60, 0.2, 2, zero_inflation = 0.25, seed = 1)
fit <- fit_dynamic_model(sim$y, zero_inflation = TRUE, nsave = 300, nburn = 200)
structural_zero_prob(fit)
```

---

summary.dynamic_fit *Summarise a fitted dynamic model*

Description

Summarise a fitted dynamic model

Usage

```
## S3 method for class 'dynamic_fit'
summary(object, probs = c(0.025, 0.5, 0.975), ...)
```

Arguments

object	A "dynamic_fit" object.
probs	Quantile probabilities. Default c(0.025, 0.5, 0.975).
...	Unused.

Value

An object of class "summary.dynamic_fit" containing posterior summaries of the global parameters (the innovation standard deviation innov_sd – the estimated SD for "gaussian" innovations, the marginal SD for "t" and "mixture", and the series-average SD for "sv" – and where relevant the t degrees of freedom, the AR(1) coefficient, the drift/intercept, and the zero-inflation gate-open probability) and the fitted-value summary.

Examples

```
sim <- simulate_dynamic_poisson(50, 0.2, 2, seed = 1)
fit <- fit_dynamic_model(sim$y, nsave = 200, nburn = 100)
summary(fit)
```

uk_weekly *Weekly English Channel crossings (UK example)*

Description

A weekly count series of irregular small-boat crossings of the English Channel towards the United Kingdom, beginning in early 2018. The series has frequent zeros in its early weeks, which makes it a useful example for zero inflation with the Poisson model. The series is used in the irregular-migration application of Zens and Bijak (2026).

Usage

```
uk_weekly
```

Format

A data frame with 376 rows and 2 variables:

week ISO week label, e.g. `"2018-W01"`.

count Non-negative integer count of weekly crossings.

Source

Weekly aggregates of detected irregular English Channel crossings, compiled from operational/agency records as described in Zens and Bijak (2026), doi:10.1214/26AOAS2171.

References

Zens, G. and Bijak, J. (2026). Dynamic Count Models with Flexible Innovation Processes for Irregular Maritime Migration. *The Annals of Applied Statistics*. doi:10.1214/26AOAS2171.

Examples

```
data(uk_weekly)
plot(uk_weekly$count, type = "h")
mean(uk_weekly$count == 0)
```

Index