

Package ‘gp3bayes’

August 9, 2026

Title Contract-First Bayesian Workflows for Hierarchical Behavioural Data

Version 0.1.1

Description Provides transparent, contract-first infrastructure for Bayesian analysis of repeated-measures and hierarchical behavioural data. It supports approved Bernoulli-logit and positive lognormal duration workflows with readiness audits, deterministic simulation, explicit preparation, inspectable priors, prior and posterior predictive checks, restricted optional fitting through 'brms' and 'rstan', sampling diagnostics, prior sensitivity, parameter recovery, and conservative reporting. Core contracts and prior checks remain backend-independent.

License MIT + file LICENSE

URL <https://stefanosbalaskas.github.io/gp3bayes/>,
<https://github.com/stefanosbalaskas/gp3bayes>

BugReports <https://github.com/stefanosbalaskas/gp3bayes/issues>

Encoding UTF-8

RoxygenNote 8.0.0

Imports withr

Suggests bayesplot, brms, knitr, posterior, rmarkdown, rstan, testthat
(>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Stefanos Balaskas [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0003-2444-9796>>)

Maintainer Stefanos Balaskas <s.balaskas@ac.upatras.gr>

Repository CRAN

Date/Publication 2026-08-09 07:10:02 UTC

Contents

assess_binary_prior_sensitivity	2
assess_duration_prior_sensitivity	4
audit_model_readiness	5
build_model_formula	6
check_binary_posterior_predictive	7
check_binary_prior_predictive	8
check_duration_posterior_predictive	9
check_duration_prior_predictive	10
create_binary_model_report	11
create_duration_model_report	12
create_model_contract	13
create_model_specification	15
create_prior_specification	16
diagnose_binary_fit	17
diagnose_duration_fit	18
fit_binary_model	19
fit_duration_model	21
plot_sampling_diagnostics	22
prepare_hierarchical_binary_data	23
prepare_hierarchical_duration_data	25
run_binary_recovery	26
run_duration_recovery	27
simulate_hierarchical_binary_data	29
simulate_hierarchical_duration_data	30
specify_binary_model	32
specify_duration_model	34
summarise_binary_posterior	35
summarise_duration_posterior	35
translate_binary_model_to_brms	36
translate_duration_model_to_brms	37
validate_prior_specification	38
Index	39

assess_binary_prior_sensitivity
Assess Binary Prior Sensitivity

Description

Refits the approved binary model under prespecified tighter and wider prior scales and compares population-level posterior medians.

Usage

```
assess_binary_prior_sensitivity(
  fit,
  scale_multipliers = c(tighter = 0.5, wider = 2),
  chains = fit$sampling$chains,
  iter = fit$sampling$iter,
  warmup = fit$sampling$warmup,
  cores = fit$sampling$cores,
  seed = fit$sampling$seed + 1000L,
  adapt_delta = fit$sampling$adapt_delta,
  max_treedepth = fit$sampling$max_treedepth,
  refresh = 0L,
  maximum_standardized_shift = 0.25,
  review_standardized_shift = 0.5,
  retain_fits = FALSE
)
```

Arguments

fit A gp3bayes_binary_fit.

scale_multipliers Named positive numeric multipliers applied to the intercept, coefficient, and group-scale priors.

chains, iter, warmup, cores, seed, adapt_delta, max_treedepth, refresh Restricted sampling controls passed to [fit_binary_model\(\)](#).

maximum_standardized_shift Maximum absolute posterior-median shift, divided by the reference posterior standard deviation, for a pass.

review_standardized_shift Maximum standardized shift for review.

retain_fits Whether alternative fitted objects should be retained.

Details

This function is computationally expensive. Sensitivity status describes stability under the declared scale changes only; it does not prove prior robustness under all defensible priors.

Value

A gp3bayes_binary_prior_sensitivity object.

assess_duration_prior_sensitivity
Assess Duration Prior Sensitivity

Description

Refits the approved duration model under prespecified tighter and wider prior scales and compares population-level and residual posterior medians.

Usage

```
assess_duration_prior_sensitivity(
  fit,
  scale_multipliers = c(tighter = 0.5, wider = 2),
  chains = fit$sampling$chains,
  iter = fit$sampling$iter,
  warmup = fit$sampling$warmup,
  cores = fit$sampling$cores,
  seed = fit$sampling$seed + 2000L,
  adapt_delta = fit$sampling$adapt_delta,
  max_treedepth = fit$sampling$max_treedepth,
  refresh = 0L,
  maximum_standardized_shift = 0.25,
  review_standardized_shift = 0.5,
  retain_fits = FALSE
)
```

Arguments

fit	A gp3bayes_duration_fit.
scale_multipliers	Named positive prior-scale multipliers.
chains, iter, warmup, cores, seed, adapt_delta, max_treedepth, refresh	Restricted sampling controls passed to fit_duration_model() .
maximum_standardized_shift	Maximum standardized median shift for pass.
review_standardized_shift	Maximum standardized median shift for review.
retain_fits	Whether alternative fits are retained.

Value

A gp3bayes_duration_prior_sensitivity.

audit_model_readiness *Audit Data Readiness for an Approved Model Contract*

Description

Audits whether a data frame satisfies the observable data requirements of an existing model contract created by `create_model_contract()`. The audit is backend-independent and does not construct a formula, define executable priors, fit a model, or establish model adequacy.

Usage

```
audit_model_readiness(data, contract)
```

Arguments

data	A data frame containing the declared outcome, grouping identifiers, predictors, and optional design columns.
contract	A gp3bayes_model_contract created by <code>create_model_contract()</code> .

Details

A readiness audit evaluates observable data properties only. Passing the audit does not establish convergence, model adequacy, predictive validity, causal identification, or substantive validity.

Failures block progression to a later model-building gate. Warnings identify weak or unusual structures that require review but do not automatically block progression.

Binary outcomes must be logical or numeric values encoded exclusively as zero and one, with both classes observed. Duration outcomes must be numeric, finite, strictly positive, uncensored, and variable.

Value

An object of class `gp3bayes_readiness_audit`. The object records:

- whether the data are ready to proceed to a later model-building gate;
- pass, warning, and failure counts;
- one structured row per readiness check;
- declared and observed column summaries; and
- the audited model contract.

The input data are not retained in the returned object.

Interpretation boundaries

Readiness checks cannot determine whether a model is scientifically justified. Behavioural measurements must not be interpreted as direct measures of latent psychological or protected attributes.

Examples

```
binary_data <- data.frame(
  participant_id = rep(c("p1", "p2"), each = 4),
  trial_id = rep(1:4, times = 2),
  condition = rep(c("control", "treatment"), times = 4),
  selected = c(0, 1, 0, 1, 1, 0, 1, 0)
)

binary_contract <- create_model_contract(
  family = "binary",
  outcome_col = "selected",
  participant_col = "participant_id",
  trial_col = "trial_id",
  condition_col = "condition"
)

audit_model_readiness(binary_data, binary_contract)
```

build_model_formula	<i>Build an Approved Model Formula</i>
---------------------	--

Description

Constructs a backend-independent R formula from a `create_model_contract()` result. The formula records the approved fixed effects, one optional interaction, the participant grouping structure, an optional participant-level random slope, and an optional crossed item intercept.

Usage

```
build_model_formula(contract)
```

Arguments

`contract` A `gp3bayes_model_contract` created by `create_model_contract()`.

Details

The participant random intercept is always included. When `contract$random_slope` is `TRUE`, it is replaced by a correlated participant intercept-and-condition-slope term. A declared item identifier adds a crossed item random intercept.

The trial identifier is treated as a row key and is not added as a predictor or grouping factor. A declared time column is included as a linear population-level term only.

Value

An R formula. The formula is a specification only and has not been translated to, validated by, or fitted with a Bayesian backend.

Examples

```
contract <- create_model_contract(
  family = "binary",
  outcome_col = "selected",
  participant_col = "participant_id",
  item_col = "stimulus_id",
  condition_col = "condition",
  predictors = "age_z"
)

build_model_formula(contract)
```

```
check_binary_posterior_predictive
```

Check Binary Posterior Predictive Behaviour

Description

Compares observed binary summaries with replicated outcomes from the fitted posterior predictive distribution.

Usage

```
check_binary_posterior_predictive(
  fit,
  draws = 500,
  seed = 1,
  pass_probability = 0.8,
  review_probability = 0.95
)
```

Arguments

<code>fit</code>	A <code>gp3bayes_binary_fit</code> .
<code>draws</code>	Number of posterior predictive replications.
<code>seed</code>	Non-negative integer seed used to select predictive draws.
<code>pass_probability</code>	Central predictive interval used for a pass.
<code>review_probability</code>	Wider central predictive interval used for review.

Details

The check evaluates prespecified descriptive summaries. It does not prove that the likelihood, link, random-effects structure, or substantive model is adequate.

Value

A gp3bayes_binary_posterior_predictive_check.

check_binary_prior_predictive

Check Binary Prior Predictive Behaviour

Description

Simulates replicated binary outcomes from the declared prior specification and prepared design without calling a Bayesian fitting backend.

Usage

```
check_binary_prior_predictive(
  specification,
  draws = 500,
  seed = 1,
  plausible_rate = c(0.01, 0.99),
  boundary_probability = c(0.01, 0.99),
  extreme_contrast = 0.8,
  maximum_degenerate_participant_fraction = 0.5,
  maximum_boundary_mass = 0.5,
  maximum_extreme_probability = 0.25
)
```

Arguments

specification	A gp3bayes_binary_model_specification.
draws	Number of prior predictive data sets.
seed	Non-negative integer random-number seed.
plausible_rate	Increasing lower and upper limits for plausible overall and condition-specific event rates.
boundary_probability	Probability thresholds used to identify prior mass close to zero and one.
extreme_contrast	Absolute probability-scale condition contrast considered extreme.
maximum_degenerate_participant_fraction	Maximum participant fraction allowed to have all-zero or all-one replicated outcomes in a draw.
maximum_boundary_mass	Maximum fraction of row probabilities allowed beyond the declared boundary thresholds in a draw.
maximum_extreme_probability	Maximum acceptable fraction of prior predictive draws violating each criterion.

Details

Failure does not select or alter priors automatically. It indicates that the declared priors and design generate outcomes that require substantive review. This check assesses prior implications, not posterior adequacy or model fit.

Value

A gp3bayes_binary_prior_predictive_check containing replicated summaries, structured checks, thresholds, and the seed.

Examples

```
simulation <- simulate_hierarchical_binary_data(
  n_participants = 12,
  trials_per_participant = 8,
  seed = 2026
)

contract <- create_model_contract(
  family = "binary",
  outcome_col = "selected",
  participant_col = "participant_id",
  item_col = "item_id",
  trial_col = "trial_id",
  condition_col = "condition"
)

prepared <- prepare_hierarchical_binary_data(
  simulation$data,
  contract,
  condition_levels = c("control", "treatment")
)

specification <- specify_binary_model(
  prepared,
  baseline = 0.35
)

check_binary_prior_predictive(
  specification,
  draws = 100,
  seed = 2027
)
```

Description

Compares observed positive-duration summaries with replicated outcomes from the fitted posterior predictive distribution.

Usage

```
check_duration_posterior_predictive(
  fit,
  draws = 500L,
  seed = 1L,
  pass_probability = 0.8,
  review_probability = 0.95
)
```

Arguments

fit	A gp3bayes_duration_fit.
draws	Number of posterior predictive data sets.
seed	Non-negative integer seed.
pass_probability	Central predictive interval used for pass.
review_probability	Wider predictive interval used for review.

Details

The check covers median, mean, upper-tail, dispersion, condition-ratio, and grouping summaries. It does not prove global model adequacy.

Value

A gp3bayes_duration_posterior_predictive_check.

check_duration_prior_predictive

Check Duration Prior Predictive Behaviour

Description

Simulates positive-duration data from the declared prior specification and prepared design without fitting a model.

Usage

```
check_duration_prior_predictive(
  specification,
  draws = 500L,
  seed = 1L,
  plausible_median = NULL,
  maximum_q99 = NULL,
  maximum_cv = 5,
  maximum_condition_ratio = 10,
  maximum_extreme_probability = 0.25
)
```

Arguments

specification	A gp3bayes_duration_model_specification.
draws	Number of prior predictive data sets.
seed	Non-negative integer seed.
plausible_median	Optional increasing pair for plausible overall medians in the prepared outcome unit.
maximum_q99	Maximum plausible 99th percentile.
maximum_cv	Maximum plausible coefficient of variation.
maximum_condition_ratio	Maximum plausible ratio between condition medians in either direction.
maximum_extreme_probability	Maximum fraction of prior predictive draws allowed to violate each criterion.

Details

Failure requests substantive prior review; it does not select or alter priors automatically.

Value

A gp3bayes_duration_prior_predictive_check.

create_binary_model_report

Create a Structured Binary Model Report

Description

Writes a conservative Markdown report for an approved fitted binary model.

Usage

```
create_binary_model_report(
  fit,
  diagnostics = NULL,
  posterior_summary = NULL,
  posterior_predictive = NULL,
  prior_sensitivity = NULL,
  recovery = NULL,
  file,
  overwrite = FALSE
)
```

Arguments

fit	A gp3bayes_binary_fit.
diagnostics	Optional result from diagnose_binary_fit() .
posterior_summary	Optional result from summarise_binary_posterior() .
posterior_predictive	Optional result from check_binary_posterior_predictive() .
prior_sensitivity	Optional result from assess_binary_prior_sensitivity() .
recovery	Optional result from run_binary_recovery() .
file	Explicit output Markdown path. The caller must supply the destination; the function has no default output path.
overwrite	Whether an existing file may be replaced.

Details

The report never converts diagnostic or predictive statuses into an automatic statement that the model converged or is substantively valid.

Value

A gp3bayes_binary_model_report containing the normalized path and section-status registry.

```
create_duration_model_report
```

Create a Structured Duration Model Report

Description

Writes a conservative Markdown report for an approved fitted lognormal duration model.

Usage

```
create_duration_model_report(
  fit,
  diagnostics = NULL,
  posterior_summary = NULL,
  posterior_predictive = NULL,
  prior_sensitivity = NULL,
  recovery = NULL,
  file,
  overwrite = FALSE
)
```

Arguments

fit	A gp3bayes_duration_fit.
diagnostics	Optional result from diagnose_duration_fit() .
posterior_summary	Optional result from summarise_duration_posterior() .
posterior_predictive	Optional result from check_duration_posterior_predictive() .
prior_sensitivity	Optional result from assess_duration_prior_sensitivity() .
recovery	Optional result from run_duration_recovery() .
file	Explicit output Markdown path. The caller must supply the destination; the function has no default output path.
overwrite	Whether an existing file may be replaced.

Value

A gp3bayes_duration_model_report.

create_model_contract *Create an Approved Bayesian Model Contract*

Description

Creates an inspectable model-contract object for one of the two model families approved for the initial gp3bayes development scope. The function records neutral data-column mappings while preserving the approved likelihood, link, estimands, assumptions, diagnostics, sensitivity requirements, and interpretation boundaries.

Usage

```
create_model_contract(
  family,
  outcome_col,
  participant_col,
  item_col = NULL,
  trial_col = NULL,
  condition_col = NULL,
  time_col = NULL,
  predictors = character(),
  interaction = NULL,
  random_slope = FALSE,
  outcome_unit = NULL,
  notes = character()
)
```

Arguments

family	Character scalar identifying the approved model family. Supported values are "binary" and "duration".
outcome_col	Character scalar naming the outcome column.
participant_col	Character scalar naming the participant identifier column.
item_col	Optional character scalar naming an item or stimulus identifier column.
trial_col	Optional character scalar naming a trial identifier column.
condition_col	Optional character scalar naming the focal condition column.
time_col	Optional character scalar naming a linear time or trial order column. This does not define a time-course or autocorrelation model.
predictors	Character vector naming additional predictors.
interaction	Optional character vector of length two naming one prespecified two-way interaction. Higher-order or multiple interactions are not supported by the initial contract.
random_slope	Logical scalar indicating whether one participant-level random slope for the focal condition is requested. Readiness must be assessed separately before fitting.
outcome_unit	Optional character scalar recording the outcome unit. It is required for the duration family and must be NULL for the binary family.
notes	Optional character vector containing user-supplied design or analysis notes. Notes do not override the approved model contract.

Details

The returned object is a specification and audit record. It does not validate a data frame, construct a backend formula, fit a model, or imply that the proposed analysis is appropriate. Those gates are handled by separate workflows.

The binary contract uses a Bernoulli likelihood with a logit link. The duration contract uses a lognormal likelihood for strictly positive, finite, uncensored durations.

Value

An object of class `gp3bayes_model_contract`. It is a named list containing the approved methodological specification, neutral column mappings, requested model structure, assumptions, diagnostics, sensitivity requirements, limitations, and unsupported uses.

Interpretation boundaries

Contract creation does not establish causal identification, model adequacy, convergence, predictive validity, or substantive validity. Behavioural measurements must not be interpreted as direct measures of latent psychological or protected attributes.

Examples

```
binary_contract <- create_model_contract(
  family = "binary",
  outcome_col = "selected",
  participant_col = "participant_id",
  item_col = "stimulus_id",
  trial_col = "trial_id",
  condition_col = "condition"
)

binary_contract

duration_contract <- create_model_contract(
  family = "duration",
  outcome_col = "response_time",
  participant_col = "participant_id",
  trial_col = "trial_id",
  condition_col = "condition",
  outcome_unit = "milliseconds"
)

duration_contract
```

```
create_model_specification
```

Create a Complete Model Specification

Description

Combines a model contract, a successful readiness audit, an approved formula, and a validated prior specification into one backend-independent model specification.

Usage

```
create_model_specification(contract, audit, priors)
```

Arguments

contract	A gp3bayes_model_contract.
audit	A gp3bayes_readiness_audit.
priors	A gp3bayes_prior_specification.

Value

An object of class gp3bayes_model_specification.

create_prior_specification

Create a Backend-Independent Prior Specification

Description

Creates an inspectable prior table for one approved model family. The returned object contains no backend-specific prior objects and performs no sampling.

Usage

```
create_prior_specification(
  contract,
  baseline = NULL,
  intercept_scale = NULL,
  coefficient_scale = NULL,
  group_sd_scale = 1,
  residual_scale = NULL,
  correlation_eta = 2,
  student_df = 3
)
```

Arguments

contract	A gp3bayes_model_contract created by create_model_contract() .
baseline	Numeric scalar describing the expected baseline outcome. For binary models this is a probability strictly between zero and one. The default is 0.5. For duration models this is a strictly positive baseline median in the recorded outcome unit and must be supplied.
intercept_scale	Optional positive numeric scalar for the normal intercept prior. Defaults to 1.5 for binary models and 1 for duration models.
coefficient_scale	Optional positive numeric scalar for normal population-level coefficient priors. Defaults to 1 for binary models and 0.5 for duration models.
group_sd_scale	Positive numeric scalar for half-Student-t group-level standard-deviation priors.

`residual_scale` Optional positive numeric scalar for the half-Student-t residual standard-deviation prior. It applies only to the duration family and defaults to 1.

`correlation_eta` Numeric scalar greater than or equal to one for the LKJ prior used when a participant-level random slope is requested.

`student_df` Positive numeric scalar giving the degrees of freedom for half-Student-t scale priors.

Details

Binary baseline probabilities are transformed with the logit function. Duration baseline medians are transformed with the natural logarithm.

Value

An object of class `gp3bayes_prior_specification`.

Examples

```
binary_contract <- create_model_contract(
  family = "binary",
  outcome_col = "selected",
  participant_col = "participant_id"
)

create_prior_specification(
  binary_contract,
  baseline = 0.35
)
```

<code>diagnose_binary_fit</code>	<i>Diagnose a Fitted Binary Model</i>
----------------------------------	---------------------------------------

Description

Computes rank-normalized R-hat, bulk and tail effective sample sizes, divergent-transition counts, maximum-treedepth saturation, and chain-level energy diagnostics for an approved binary fit.

Usage

```
diagnose_binary_fit(
  fit,
  rhat_pass = 1.01,
  rhat_fail = 1.05,
  ess_per_chain_pass = 100,
  ess_per_chain_fail = 50,
  maximum_treedepth_fraction = 0.01,
```

```

    ebfmi_pass = 0.3,
    ebfmi_fail = 0.2
  )

```

Arguments

<code>fit</code>	A fitted <code>gp3bayes_fit</code> .
<code>rhat_pass</code>	R-hat value at or below which the component passes.
<code>rhat_fail</code>	R-hat value above which the component fails.
<code>ess_per_chain_pass</code>	Bulk or tail ESS per chain at or above which the component passes.
<code>ess_per_chain_fail</code>	Bulk or tail ESS per chain below which the component fails.
<code>maximum_treedepth_fraction</code>	Maximum fraction of post-warmup draws that may reach the configured maximum treedepth before the component fails.
<code>ebfmi_pass</code>	E-BFMI value at or above which the energy component passes.
<code>ebfmi_fail</code>	E-BFMI value below which the energy component fails.

Details

The overall status is "fail" when any component fails, "review" when any component requires review or cannot be assessed, and "pass" only when every component passes. A pass does not automatically establish convergence or posterior adequacy.

Value

A `gp3bayes_binary_diagnostics` object containing parameter, component, and chain-level diagnostic tables.

`diagnose_duration_fit` *Diagnose a Fitted Duration Model*

Description

Applies the package sampling-diagnostic contract to an approved hierarchical lognormal duration fit.

Usage

```

diagnose_duration_fit(
  fit,
  rhat_pass = 1.01,
  rhat_fail = 1.05,
  ess_per_chain_pass = 100,
  ess_per_chain_fail = 50,

```

```

    maximum_treedepth_fraction = 0.01,
    ebfmi_pass = 0.3,
    ebfmi_fail = 0.2
  )

```

Arguments

fit	A gp3bayes_duration_fit.
rhat_pass	R-hat value at or below which the component passes.
rhat_fail	R-hat value above which the component fails.
ess_per_chain_pass	Bulk or tail ESS per chain at or above which the component passes.
ess_per_chain_fail	Bulk or tail ESS per chain below which the component fails.
maximum_treedepth_fraction	Maximum fraction of post-warmup draws that may reach the configured maximum treedepth before the component fails.
ebfmi_pass	E-BFMI value at or above which the energy component passes.
ebfmi_fail	E-BFMI value below which the energy component fails.

Details

The returned status reports prespecified numerical sampling thresholds. It does not automatically establish convergence or posterior adequacy.

Value

A gp3bayes_duration_diagnostics object.

fit_binary_model	<i>Fit an Approved Hierarchical Binary Model</i>
------------------	--

Description

Fits an approved binary model specification using full MCMC sampling through the fixed brms and rstan route.

Usage

```

fit_binary_model(
  specification,
  chains = 4L,
  iter = 2000L,
  warmup = 1000L,
  cores = .gp3b_default_cores(chains),
  seed = 1L,

```

```

    adapt_delta = 0.95,
    max_treedepth = 12L,
    refresh = 0L
  )

```

Arguments

specification	A gp3bayes_binary_model_specification.
chains	Number of MCMC chains.
iter	Total iterations per chain, including warmup.
warmup	Warmup iterations per chain.
cores	Number of processor cores. It cannot exceed chains.
seed	Non-negative integer random-number seed.
adapt_delta	Target acceptance probability for the No-U-Turn sampler.
max_treedepth	Maximum tree depth for the No-U-Turn sampler.
refresh	Console progress refresh interval. Use zero to suppress iteration progress output.

Details

The function fixes the likelihood to Bernoulli, the link to logit, the interface to brms, the sampling backend to rstan, and the algorithm to full MCMC sampling. It does not expose arbitrary backend arguments.

A returned fit is not evidence of convergence, posterior adequacy, causal identification, or substantive validity. Those assessments require separate diagnostic and reporting gates.

Value

A gp3bayes_binary_fit containing the fitted backend object, original specification, restricted translation, and recorded sampling settings.

Examples

```

if (
  requireNamespace("brms", quietly = TRUE) &&
  requireNamespace("rstan", quietly = TRUE)
) {
  simulation <- simulate_hierarchical_binary_data(
    n_participants = 8,
    trials_per_participant = 6,
    n_items = 4,
    random_slope_sd = 0,
    seed = 2026
  )

  contract <- create_model_contract(
    family = "binary",
    outcome_col = "selected",
    participant_col = "participant_id",

```

```
    item_col = "item_id",
    trial_col = "trial_id",
    condition_col = "condition"
  )

  prepared <- prepare_hierarchical_binary_data(
    simulation$data,
    contract,
    condition_levels = c(
      "control",
      "treatment"
    )
  )

  specification <- specify_binary_model(
    prepared,
    baseline = 0.35
  )

  fit <- fit_binary_model(
    specification,
    chains = 2,
    iter = 200,
    warmup = 100,
    cores = 2,
    seed = 2026,
    refresh = 0
  )
}
```

fit_duration_model	<i>Fit an Approved Hierarchical Lognormal Duration Model</i>
--------------------	--

Description

Fits an approved strictly positive uncensored duration model using full MCMC through the fixed brms and rstan route.

Usage

```
fit_duration_model(
  specification,
  chains = 4L,
  iter = 2000L,
  warmup = 1000L,
  cores = .gp3b_default_cores(chains),
  seed = 1L,
```

```

    adapt_delta = 0.95,
    max_treedepth = 12L,
    refresh = 0L
  )

```

Arguments

specification	A gp3bayes_duration_model_specification.
chains	Number of MCMC chains.
iter	Total iterations per chain.
warmup	Warmup iterations per chain.
cores	Processor cores, not exceeding chains.
seed	Non-negative integer seed.
adapt_delta	Target NUTS acceptance probability.
max_treedepth	Maximum NUTS tree depth.
refresh	Console progress refresh interval.

Details

The likelihood is fixed to lognormal, the link to identity on the mean-log parameter, the interface to brms, the backend to rstan, and the algorithm to full MCMC sampling. A returned fit does not establish convergence or posterior adequacy.

Value

A gp3bayes_duration_fit.

plot_sampling_diagnostics
Plot Sampling Diagnostics

Description

Produces trace, energy, treedepth, or divergence plots for an approved fitted gp3bayes model.

Usage

```

plot_sampling_diagnostics(
  fit,
  type = c("trace", "energy", "treedepth", "divergence"),
  variables = NULL
)

```

Arguments

fit	A fitted gp3bayes_fit.
type	One of "trace", "energy", "treedepth", or "divergence".
variables	Optional posterior parameter names used for trace plots.

Details

Diagnostic plots support interpretation of sampling behaviour. They do not establish convergence or substantive model adequacy by themselves.

Value

A plot object created by bayesplot.

```
prepare_hierarchical_binary_data
```

Prepare Hierarchical Binary Data

Description

Applies explicit binary outcome mapping, explicit two-level condition coding, optional recorded numeric scaling, and a model-readiness gate. No variable is silently scaled or recoded.

Usage

```
prepare_hierarchical_binary_data(
  data,
  contract,
  outcome_mapping = NULL,
  condition_levels = NULL,
  condition_coding = c(-0.5, 0.5),
  scale_predictors = character(),
  scale_time = FALSE,
  missing = c("error", "drop")
)
```

Arguments

data	A data frame containing the columns declared in contract.
contract	A binary gp3bayes_model_contract.
outcome_mapping	Optional named vector mapping two labelled outcome values to 0 and 1. It is required for non-logical, non-0/1 outcomes.
condition_levels	Optional two-value vector listing the condition levels in reference-to-focal order.

<code>condition_coding</code>	Two distinct finite numeric values used to encode the declared condition. The default is <code>c(-0.5, 0.5)</code> .
<code>scale_predictors</code>	Character vector naming declared numeric predictors to centre and divide by their sample standard deviation.
<code>scale_time</code>	Whether to centre and scale the declared linear time variable.
<code>missing</code>	Either "error" or "drop". Dropping is performed only after this explicit argument is selected, and removed row positions are recorded.

Details

This function performs deterministic preparation only. It does not fit a model, create posterior draws, or establish causal or substantive validity.

Value

A `gp3bayes_binary_prepared` object containing the analysis data, contract, readiness audit, transformation registry, fixed-effects formula, design-matrix columns, and row accounting.

Examples

```
simulation <- simulate_hierarchical_binary_data(
  n_participants = 12,
  trials_per_participant = 8,
  seed = 2026
)

contract <- create_model_contract(
  family = "binary",
  outcome_col = "selected",
  participant_col = "participant_id",
  item_col = "item_id",
  trial_col = "trial_id",
  condition_col = "condition",
  predictors = c(
    "participant_covariate",
    "trial_covariate"
  ),
  interaction = c(
    "condition",
    "participant_covariate"
  ),
  random_slope = TRUE
)

prepared <- prepare_hierarchical_binary_data(
  simulation$data,
  contract,
  condition_levels = c("control", "treatment")
)
```

prepared

```
prepare_hierarchical_duration_data
```

Prepare Hierarchical Duration Data

Description

Validates strictly positive finite uncensored durations, applies explicit unit conversion and recorded scaling, and runs the duration readiness gate.

Usage

```
prepare_hierarchical_duration_data(
  data,
  contract,
  condition_levels = NULL,
  condition_coding = c(-0.5, 0.5),
  scale_predictors = character(),
  scale_time = FALSE,
  outcome_multiplier = 1,
  converted_unit = NULL,
  missing = c("error", "drop")
)
```

Arguments

<code>data</code>	A data frame containing columns declared in <code>contract</code> .
<code>contract</code>	A duration <code>gp3bayes_model_contract</code> .
<code>condition_levels</code>	Optional two-value condition order.
<code>condition_coding</code>	Two distinct finite numeric condition codes.
<code>scale_predictors</code>	Declared numeric predictors to centre and scale.
<code>scale_time</code>	Whether to centre and scale the declared linear time variable.
<code>outcome_multiplier</code>	Positive unit-conversion multiplier.
<code>converted_unit</code>	Required destination unit when <code>outcome_multiplier != 1</code> .
<code>missing</code>	Either "error" or "drop".

Details

Zero, negative, non-finite, censored, truncated, or shifted durations are not supported. Unit conversion is never inferred.

Value

A gp3bayes_duration_prepared object.

run_binary_recovery	<i>Run Binary Parameter Recovery</i>
---------------------	--------------------------------------

Description

Repeatedly simulates from the approved hierarchical Bernoulli-logit generator, fits the restricted model, and compares posterior intervals with known generating values.

Usage

```
run_binary_recovery(
  repetitions = 20L,
  n_participants = 30L,
  trials_per_participant = 16L,
  n_items = 12L,
  include_items = TRUE,
  random_slope = TRUE,
  seed = 1001L,
  chains = 4L,
  iter = 1500L,
  warmup = 750L,
  cores = min(chains, .gp3b_default_cores(chains)),
  adapt_delta = 0.95,
  max_treedepth = 12L,
  refresh = 0L,
  interval_probability = 0.95,
  minimum_repetitions = 20L,
  maximum_standardized_bias = 0.25,
  minimum_coverage = 0.8,
  minimum_diagnostic_pass_fraction = 0.8,
  continue_on_error = TRUE
)
```

Arguments

repetitions	Number of simulation-fit repetitions.
n_participants, trials_per_participant, n_items	Synthetic design sizes.
include_items	Whether crossed item effects are included.
random_slope	Whether a participant condition slope is generated and fitted.
seed	First simulation seed.
chains, iter, warmup, cores, adapt_delta, max_treedepth, refresh	Restricted sampling controls.

interval_probability	Central posterior interval probability.
minimum_repetitions	Repetitions required before an overall pass is possible.
maximum_standardized_bias	Maximum absolute bias divided by the empirical standard deviation of estimates for a pass.
minimum_coverage	Minimum empirical interval coverage for a pass.
minimum_diagnostic_pass_fraction	Minimum fraction of fits with a diagnostic pass.
continue_on_error	Whether failed repetitions are recorded instead of stopping.

Details

A small recovery run is a smoke test, not validation. Even when all declared thresholds pass, the object records no automatic validation claim.

Value

A gp3bayes_binary_recovery object.

run_duration_recovery *Run Duration Parameter Recovery*

Description

Repeatedly simulates and fits the approved hierarchical lognormal duration model and compares posterior intervals with known generating values.

Usage

```
run_duration_recovery(
  repetitions = 20L,
  n_participants = 30L,
  trials_per_participant = 16L,
  n_items = 12L,
  include_items = TRUE,
  random_slope = TRUE,
  baseline_median = 500,
  outcome_unit = "milliseconds",
  seed = 2001L,
  chains = 4L,
  iter = 1500L,
  warmup = 750L,
  cores = min(chains, .gp3b_default_cores(chains)),
```

```

    adapt_delta = 0.95,
    max_treedepth = 12L,
    refresh = 0L,
    interval_probability = 0.95,
    minimum_repetitions = 20L,
    maximum_standardized_bias = 0.25,
    minimum_coverage = 0.8,
    minimum_diagnostic_pass_fraction = 0.8,
    continue_on_error = TRUE
  )

```

Arguments

<code>repetitions</code>	Number of simulation-fit repetitions.
<code>n_participants</code> , <code>trials_per_participant</code> , <code>n_items</code>	Synthetic design sizes.
<code>include_items</code>	Whether crossed item effects are included.
<code>random_slope</code>	Whether a participant condition slope is generated and fitted.
<code>baseline_median</code>	Baseline synthetic duration median.
<code>outcome_unit</code>	Synthetic duration unit.
<code>seed</code>	First simulation seed.
<code>chains</code> , <code>iter</code> , <code>warmup</code> , <code>cores</code> , <code>adapt_delta</code> , <code>max_treedepth</code> , <code>refresh</code>	Restricted sampling controls.
<code>interval_probability</code>	Central posterior interval probability.
<code>minimum_repetitions</code>	Repetitions required before an overall pass is possible.
<code>maximum_standardized_bias</code>	Maximum absolute bias divided by the empirical standard deviation of estimates for a pass.
<code>minimum_coverage</code>	Minimum empirical interval coverage for a pass.
<code>minimum_diagnostic_pass_fraction</code>	Minimum fraction of fits with a diagnostic pass.
<code>continue_on_error</code>	Whether failed repetitions are recorded instead of stopping.

Details

A small run is a smoke test. No result creates an automatic validation claim.

Value

A `gp3bayes_duration_recovery`.

simulate_hierarchical_binary_data

Simulate Hierarchical Binary Data

Description

Generates deterministic synthetic repeated-measures data from the approved Bernoulli-logit contract. The simulator records all generating parameters, participant effects, optional crossed item effects, and the random-number seed. It performs no model fitting.

Usage

```
simulate_hierarchical_binary_data(
  n_participants = 40,
  trials_per_participant = 20,
  n_items = 20,
  intercept = stats::qlogis(0.35),
  condition_effect = 0.8,
  participant_covariate_effect = 0.3,
  trial_covariate_effect = 0.15,
  interaction_effect = 0.25,
  participant_sd = 0.7,
  item_sd = 0.35,
  random_slope_sd = 0.3,
  random_slope_cor = 0,
  condition_probability = 0.5,
  balanced_condition = TRUE,
  include_items = TRUE,
  seed = 1
)
```

Arguments

n_participants	Number of participants.
trials_per_participant	Number of observations per participant.
n_items	Number of crossed items when include_items = TRUE.
intercept	Population intercept on the log-odds scale.
condition_effect	Population condition contrast on the log-odds scale.
participant_covariate_effect	Participant-covariate coefficient.
trial_covariate_effect	Trial-covariate coefficient.
interaction_effect	Condition-by-participant-covariate coefficient.

participant_sd Participant random-intercept standard deviation.
 item_sd Crossed item random-intercept standard deviation.
 random_slope_sd Participant condition-slope standard deviation.
 random_slope_cor Correlation between participant intercepts and condition slopes. It must lie strictly between -1 and 1.
 condition_probability Treatment probability when balanced_condition = FALSE.
 balanced_condition Whether each participant receives an approximately balanced condition sequence.
 include_items Whether to generate a crossed item identifier.
 seed Non-negative integer random-number seed.

Details

The condition is generated using -0.5 and 0.5 internally and returned as a factor with levels control and treatment. The data-generating model includes one participant random intercept, one optional correlated participant condition slope, and one optional crossed item intercept.

Value

A gp3bayes_binary_simulation containing synthetic data, stored truth, generated random effects, and design metadata.

Examples

```

simulation <- simulate_hierarchical_binary_data(
  n_participants = 12,
  trials_per_participant = 8,
  n_items = 6,
  seed = 2026
)

simulation
head(simulation$data)

```

simulate_hierarchical_duration_data

Simulate Hierarchical Lognormal Duration Data

Description

Generates deterministic strictly positive uncensored durations from the approved hierarchical log-normal contract.

Usage

```
simulate_hierarchical_duration_data(
  n_participants = 40L,
  trials_per_participant = 20L,
  n_items = 20L,
  baseline_median = 500,
  condition_effect = log(1.15),
  participant_covariate_effect = log(1.08),
  trial_covariate_effect = log(1.04),
  interaction_effect = log(1.05),
  participant_sd = 0.35,
  item_sd = 0.2,
  random_slope_sd = 0.15,
  random_slope_cor = 0,
  residual_sd = 0.4,
  condition_probability = 0.5,
  balanced_condition = TRUE,
  include_items = TRUE,
  outcome_unit = "milliseconds",
  seed = 1L
)
```

Arguments

n_participants Number of participants.

trials_per_participant Observations per participant.

n_items Number of crossed items when `include_items = TRUE`.

baseline_median Baseline duration median in `outcome_unit`.

condition_effect Population condition contrast on the log-duration scale.

participant_covariate_effect Participant-covariate coefficient on the log-duration scale.

trial_covariate_effect Trial-covariate coefficient on the log-duration scale.

interaction_effect Condition-by-participant-covariate coefficient on the log-duration scale.

participant_sd Participant random-intercept standard deviation on the log scale.

item_sd Crossed item random-intercept standard deviation.

random_slope_sd Participant condition-slope standard deviation.

random_slope_cor Correlation between participant intercepts and condition slopes.

residual_sd Lognormal residual standard deviation.

condition_probability	Focal-condition probability for an unbalanced design.
balanced_condition	Whether each participant receives an approximately balanced condition sequence.
include_items	Whether crossed items are generated.
outcome_unit	Recorded duration unit.
seed	Non-negative integer random-number seed.

Details

The generated outcome is strictly positive, finite, and uncensored. The function does not generate zero, censored, truncated, shifted, or survival outcomes.

Value

A gp3bayes_duration_simulation containing synthetic data, stored truth, random effects, and design metadata.

specify_binary_model *Specify a Backend-Independent Binary Model*

Description

Combines prepared binary data, a successful readiness audit, the restricted hierarchical formula, and validated family-specific priors. The returned object is not executable and performs no model fitting.

Usage

```
specify_binary_model(
  prepared,
  baseline = 0.5,
  intercept_scale = 1.5,
  coefficient_scale = 0.75,
  group_sd_scale = 1,
  correlation_eta = 2,
  student_df = 3
)
```

Arguments

prepared	A gp3bayes_binary_prepared object.
baseline	Plausible baseline event probability.
intercept_scale	Optional scale for the normal intercept prior.

coefficient_scale	Optional common scale for normal population-level coefficient priors, including the approved interaction.
group_sd_scale	Scale for half-Student-t group standard deviations.
correlation_eta	LKJ shape used when a random slope is requested.
student_df	Degrees of freedom for half-Student-t scale priors.

Details

The specification retains the prepared data because backend-independent prior predictive simulation must reproduce the declared design. It contains no backend object, posterior draws, or fitted model.

Value

A `gp3bayes_binary_model_specification` that also inherits from `gp3bayes_model_specification`.

Examples

```
simulation <- simulate_hierarchical_binary_data(
  n_participants = 12,
  trials_per_participant = 8,
  seed = 2026
)

contract <- create_model_contract(
  family = "binary",
  outcome_col = "selected",
  participant_col = "participant_id",
  item_col = "item_id",
  trial_col = "trial_id",
  condition_col = "condition"
)

prepared <- prepare_hierarchical_binary_data(
  simulation$data,
  contract,
  condition_levels = c("control", "treatment")
)

specification <- specify_binary_model(
  prepared,
  baseline = 0.35
)

specification
```

```
specify_duration_model
```

Specify a Backend-Independent Duration Model

Description

Combines prepared positive-duration data with the approved hierarchical lognormal formula and explicit prior specification.

Usage

```
specify_duration_model(
  prepared,
  baseline,
  intercept_scale = 1,
  coefficient_scale = 0.5,
  group_sd_scale = 1,
  residual_scale = 1,
  correlation_eta = 2,
  student_df = 3
)
```

Arguments

<code>prepared</code>	A <code>gp3bayes_duration_prepared</code> .
<code>baseline</code>	Plausible baseline median in the prepared outcome unit.
<code>intercept_scale</code>	Positive normal-intercept prior scale.
<code>coefficient_scale</code>	Positive population-coefficient prior scale.
<code>group_sd_scale</code>	Positive group standard-deviation prior scale.
<code>residual_scale</code>	Positive lognormal residual-scale prior scale.
<code>correlation_eta</code>	LKJ shape for an approved random slope.
<code>student_df</code>	Degrees of freedom for half-Student-t scale priors.

Value

A `gp3bayes_duration_model_specification`.

`summarise_binary_posterior`*Summarise a Binary Posterior*

Description

Reports posterior location, uncertainty intervals, R-hat, effective sample sizes, probability of a positive coefficient, and odds-ratio transforms for population-level coefficients.

Usage

```
summarise_binary_posterior(fit, probability = 0.95, variables = NULL)
```

Arguments

<code>fit</code>	A <code>gp3bayes_binary_fit</code> .
<code>probability</code>	Central posterior interval probability.
<code>variables</code>	Optional supported posterior variable names.

Details

Probability-positive values and intervals are descriptive posterior summaries. They are not frequentist significance tests and do not establish causal or substantive validity.

Value

A `gp3bayes_binary_posterior_summary`.

`summarise_duration_posterior`*Summarise a Duration Posterior*

Description

Reports posterior location, uncertainty, diagnostics, and multiplicative duration-ratio transforms for population-level coefficients.

Usage

```
summarise_duration_posterior(fit, probability = 0.95, variables = NULL)
```

Arguments

<code>fit</code>	A <code>gp3bayes_duration_fit</code> .
<code>probability</code>	Central posterior interval probability.
<code>variables</code>	Optional supported posterior variable names.

Details

Exponentiating a population-level coefficient gives its conditional multiplicative effect on the median duration under the approved lognormal model. This is not automatically a causal effect.

Value

A gp3bayes_duration_posterior_summary.

translate_binary_model_to_brms

Translate a Binary Model Specification to brms

Description

Converts an approved backend-independent binary specification into a restricted brms representation. The formula, Bernoulli-logit family, and priors are derived entirely from the existing gp3bayes specification.

Usage

```
translate_binary_model_to_brms(specification)
```

Arguments

specification A gp3bayes_binary_model_specification.

Details

This function performs translation and prior validation only. It does not compile Stan code, run MCMC, create posterior draws, or assess convergence. Users cannot supply an alternative formula, family, backend, algorithm, or arbitrary backend arguments.

Value

A gp3bayes_binary_backend_specification containing the restricted formula, family, translated priors, validated prior table, and backend metadata.

Examples

```
if (requireNamespace("brms", quietly = TRUE)) {
  simulation <- simulate_hierarchical_binary_data(
    n_participants = 12,
    trials_per_participant = 8,
    n_items = 6,
    random_slope_sd = 0,
    seed = 2026
  )
}
```

```

contract <- create_model_contract(
  family = "binary",
  outcome_col = "selected",
  participant_col = "participant_id",
  item_col = "item_id",
  trial_col = "trial_id",
  condition_col = "condition"
)

prepared <- prepare_hierarchical_binary_data(
  simulation$data,
  contract,
  condition_levels = c("control", "treatment")
)

specification <- specify_binary_model(
  prepared,
  baseline = 0.35
)

translate_binary_model_to_brms(specification)
}

```

translate_duration_model_to_brms

Translate a Duration Model Specification to brms

Description

Converts an approved backend-independent duration specification into a fixed hierarchical lognormal brms representation.

Usage

```
translate_duration_model_to_brms(specification)
```

Arguments

`specification` A `gp3bayes_duration_model_specification`.

Details

Translation validates the formula and priors but does not compile Stan code or fit a model. Users cannot supply an alternative family, formula, backend, algorithm, Stan extension, or arbitrary backend arguments.

Value

A `gp3bayes_duration_backend_specification`.

validate_prior_specification
Validate a Prior Specification

Description

Validates the completeness and internal consistency of a gp3bayes_prior_specification.

Usage

```
validate_prior_specification(priors, contract = NULL)
```

Arguments

priors	A gp3bayes_prior_specification created by create_prior_specification() .
contract	Optional gp3bayes_model_contract.

Value

priors, invisibly.

Index

assess_binary_prior_sensitivity, [2](#)
assess_binary_prior_sensitivity(), [12](#)
assess_duration_prior_sensitivity, [4](#)
assess_duration_prior_sensitivity(),
[13](#)
audit_model_readiness, [5](#)

build_model_formula, [6](#)

check_binary_posterior_predictive, [7](#)
check_binary_posterior_predictive(),
[12](#)
check_binary_prior_predictive, [8](#)
check_duration_posterior_predictive, [9](#)
check_duration_posterior_predictive(),
[13](#)
check_duration_prior_predictive, [10](#)
create_binary_model_report, [11](#)
create_duration_model_report, [12](#)
create_model_contract, [13](#)
create_model_contract(), [5](#), [6](#), [16](#)
create_model_specification, [15](#)
create_prior_specification, [16](#)
create_prior_specification(), [38](#)

diagnose_binary_fit, [17](#)
diagnose_binary_fit(), [12](#)
diagnose_duration_fit, [18](#)
diagnose_duration_fit(), [13](#)

fit_binary_model, [19](#)
fit_binary_model(), [3](#)
fit_duration_model, [21](#)
fit_duration_model(), [4](#)

plot_sampling_diagnostics, [22](#)
prepare_hierarchical_binary_data, [23](#)
prepare_hierarchical_duration_data, [25](#)

run_binary_recovery, [26](#)
run_binary_recovery(), [12](#)

run_duration_recovery, [27](#)
run_duration_recovery(), [13](#)

simulate_hierarchical_binary_data, [29](#)
simulate_hierarchical_duration_data,
[30](#)
specify_binary_model, [32](#)
specify_duration_model, [34](#)
summarise_binary_posterior, [35](#)
summarise_binary_posterior(), [12](#)
summarise_duration_posterior, [35](#)
summarise_duration_posterior(), [13](#)

translate_binary_model_to_brms, [36](#)
translate_duration_model_to_brms, [37](#)

validate_prior_specification, [38](#)