Package 'RoBTT'

November 4, 2026

Title Robust Bayesian T-Test

Version 1.4.0

Maintainer František Bartoš <f.bartos96@gmail.com>

Description An implementation of Bayesian model-averaged t-tests that allows users to draw inferences about the presence versus absence of an effect, variance heterogeneity, and potential outliers. The 'RoBTT' package estimates ensembles of models created by combining competing hypotheses and applies Bayesian model averaging using posterior model probabilities. Users can obtain model-averaged posterior distributions and inclusion Bayes factors, accounting for uncertainty in the data-generating process (Maier et al., 2024, <doi:10.3758/s13423-024-02590-5>). The package also provides a truncated likelihood version of the model-averaged t-test, enabling users to exclude potential outliers without introducing bias (Godmann et al., 2024, <doi:10.31234/osf.io/j9f3s>). Users can specify a wide range of informative priors for all parameters of interest. The package offers convenient functions for summary, visualization, and fit diagnostics.

URL https://fbartos.github.io/RoBTT/

BugReports https://github.com/FBartos/RoBTT/issues

License GPL-3

Encoding UTF-8

LazyData true

SystemRequirements GNU make

RoxygenNote 7.3.2

NeedsCompilation yes

ByteCompile true

LinkingTo StanHeaders (>= 2.26.0), rstan (>= 2.26.0), BH (>= 1.69.0), Rcpp (>= 0.12.15), RcppEigen (>= 0.3.3.4.0), RcppParallel (>= 5.0.1)

Depends R (>= 4.0.0), Rcpp (>= 0.12.19)

Imports rstan(>= 2.26.0), rstantools(>= 1.5.0), RcppParallel (>= 5.0.1), BayesTools (>= 0.4.0), bridgesampling, methods, ggplot2, Rdpack

Suggests parallel, testthat, vdiffr, knitr, rmarkdown, covr

RdMacros Rdpack

VignetteBuilder knitr

Author František Bartoš [aut, cre] (ORCID: <https://orcid.org/0000-0002-0018-5573>), Maximilian Maier [aut] (ORCID: <https://orcid.org/0000-0002-9873-6096>), Henrik R Godmann [aut] (ORCID: <https://orcid.org/0009-0003-6410-4765>)

Repository CRAN

Date/Publication 2026-11-04 20:20:02 UTC

Contents

RoBTT-package . . . 2
check_RoBTT . . . 3
check_setup . . . 4
diagnostics . . . 5
fertilization . . . 8
interpret . . . 8
is.RoBTT . . . 9
plot.RoBTT . . . 9
print.RoBTT . . . 11
print.summary.RoBTT . . . 11
prior . . . 12
prior_none . . . 13
rho2logsdr . . . 14
RoBTT . . . 15
RoBTT_control . . . 18
RoBTT_options . . . 19
summary.RoBTT . . . 19
update.RoBTT . . . 21

Index **24**

`RoBTT-package` *RoBTT: Robust Bayesian t-test*

Description

RoBTT: Bayesian model-averaged t-test extends the Bayesian t-test by incorporating inference about heterogeneity of variances and outliers.

User guide

See Maier et al. (2024) for details regarding the RoBTT methodology.

More details regarding customization of the Bayesian model-averaged t-test are provided in **Introduction to RoBTT** vignette. Please, use the "Issues" section in the GitHub repository to ask any further questions.

Author(s)

František Bartoš <f.bartos96@gmail.com>

References

Maier M, Bartoš F, Quintana DS, van den Bergh D, Marsman M, Ly A, Wagenmakers E (2024). "Model-averaged Bayesian t-tests." *Psychonomic Bulletin & Review*. doi:10.3758/s13423024-025905.

See Also

Useful links:

- https://fbartos.github.io/RoBTT/
- Report bugs at https://github.com/FBartos/RoBTT/issues

---

check_RoBTT *Check fitted 'RoBTT' object for errors and warnings*

Description

Checks fitted 'RoBTT' object for warnings and errors and prints them to the console.

Usage

```
check_RoBTT(fit)
```

Arguments

fit	a fitted 'RoBTT' object.

Value

check_RoBTT returns a vector of error and warning messages.

check_setup *Prints summary of* "RoBTT" *ensemble implied by the specified priors*

Description

check_setup prints summary of "RoBTT" ensemble implied by the specified prior distributions. It is useful for checking the ensemble configuration prior to fitting all of the models.

Usage

```
check_setup(
 prior_delta = prior(distribution = "cauchy", parameters = list(location = 0, scale =
    sqrt(2)/2)),
 prior_rho = prior(distribution = "beta", parameters = list(alpha = 1, beta = 1)),
  prior_nu = prior(distribution = "exp", parameters = list(rate = 1)),
 prior_delta_null = prior(distribution = "spike", parameters = list(location = 0)),
 prior_rho_null = prior(distribution = "spike", parameters = list(location = 0.5)),
  prior_nu_null = prior_none(),
  prior_mu = NULL,
  prior_sigma2 = NULL,
  truncation = NULL,
  models = FALSE,
  silent = FALSE
)
```

Arguments

prior_delta	prior distributions for the effect size delta parameter that will be treated as belonging to the alternative hypothesis. Defaults to prior(distribution = "Cauchy", parameters = list(location = 0, scale = sqrt(2)/2)).
prior_rho	prior distributions for the precision allocation rho parameter that will be treated as belonging to the alternative hypothesis. Defaults to prior(distribution = "beta", parameters = list(alpha = 1, beta = 1)).
prior_nu	prior distribution for the degrees of freedom + 2 nu parameter that will be treated as belonging to the alternative hypothesis. Defaults to prior(distribution = "exp", parameters = list(rate = 1)) if no truncation is specified. If truncation is specified, the default is NULL (i.e., use only normal likelihood).
prior_delta_null	prior distribution for the delta parameter that will be treated as belonging to the null hypothesis. Defaults to point distribution with location at 0 (prior(distribution = "point", parameters = list(location = 0))).
prior_rho_null	prior distribution for the rho parameter that will be treated as belonging to the null hypothesis. Defaults to point distribution with location at 0.5 (prior(distribution = "point", parameters = list(location = 0.5))).
prior_nu_null	prior distribution for the nu parameter that will be treated as belonging to the null hypothesis. Defaults to prior_none ((i.e., normal likelihood)).

`prior_mu` prior distribution for the grand mean parameter. Defaults to `NULL` which sets Jeffreys prior for the grand mean in case of no truncation or an unit Cauchy prior distributions for the grand mean in case of truncation (which greatly improves sampling efficiency).

`prior_sigma2` prior distribution for the grand variance parameter. Defaults to `NULL` which sets Jeffreys prior for the variance in case of no truncation or an exponential prior distribution for the variance in case of truncation (which greatly improves sampling efficiency).

`truncation` an optional list specifying truncation applied to the data. Defaults to `NULL`, i.e., no truncation was applied and the full likelihood is applied. Alternative the truncation can be specified via a named list with:

- `"x"` where `x` is a vector of two values specifying the lower and upper truncation points common across the groups
- `"x1"` **and** `"x2"` where `x1` is a vector of two values specifying the lower and upper truncation points for the first group and `x2` is a vector of two values specifying the lower and upper truncation points for the second group.
- `"sigma"` where `sigma` corresponds to the number of standard deviations from the common mean where the truncation points should be set.
- `"sigma1"` **and** `"sigma2"` where `sigma1` corresponds to the number of standard deviations from the mean of the first group where the truncation points should be set and `sigma2` corresponds to the number of standard deviations from the mean of the second group where the truncation points should be set.

`models` should the models' details be printed.

`silent` do not print the results.

Value

`check_setup` invisibly returns list of summary tables.

See Also

`RoBTT()`

`diagnostics` *Checks a fitted RoBTT object*

Description

`diagnostics` creates visual checks of individual models convergence. Numerical overview of individual models can be obtained by `summary(object, type = "models", diagnostics = TRUE)`, or even more detailed information by `summary(object, type = "individual")`.

Usage

```
diagnostics(
  fit,
  parameter,
  type,
  show_models = NULL,
  lags = 30,
  title = is.null(show_models) | length(show_models) > 1,
  ...
)

diagnostics_autocorrelation(
  fit,
  parameter = NULL,
  show_models = NULL,
  lags = 30,
  title = is.null(show_models) | length(show_models) > 1,
  ...
)

diagnostics_trace(
  fit,
  parameter = NULL,
  show_models = NULL,
  title = is.null(show_models) | length(show_models) > 1,
  ...
)

diagnostics_density(
  fit,
  parameter = NULL,
  show_models = NULL,
  title = is.null(show_models) | length(show_models) > 1,
  ...
)
```

Arguments

`fit`	a fitted RoBTT object
`parameter`	a parameter to be plotted. Either `"delta"`, `"rho"`, `"nu"`, `"mu"`, or `"sigma"`.
`type`	type of MCMC diagnostic to be plotted. Options are `"chains"` for the chains' trace plots, `"autocorrelation"` for autocorrelation of the chains, and `"densities"` for the overlaying densities of the individual chains. Can be abbreviated to first letters.
`show_models`	MCMC diagnostics of which models should be plotted. Defaults to `NULL` which plots MCMC diagnostics for a specified parameter for every model that is part of the ensemble.

lags	number of lags to be shown for type = "autocorrelation". Defaults to 30.
title	whether the model number should be displayed in title. Defaults to TRUE when more than one model is selected.
...	additional arguments to be passed to par if plot_type = "base".

Details

The visualization functions are based on stan_plot function and its color schemes.

Value

diagnostics returns either NULL if plot_type = "base" or an object/list of objects (depending on the number of parameters to be plotted) of class 'ggplot2' if plot_type = "ggplot2".

See Also

RoBTT(), summary.RoBTT()

Examples

```
## Not run:
# using the example data from Darwin
data("fertilization", package = "RoBTT")
fit <- RoBTT(
  x1       = fertilization$Self,
  x2       = fertilization$Crossed,
  prior_delta = prior("cauchy", list(0, 1/sqrt(2))),
  prior_rho   = prior("beta",   list(3, 3)),
  seed        = 1,
  chains      = 1,
  warmup      = 1000,
  iter        = 2000,
  control     = set_control(adapt_delta = 0.95)
)

### ggplot2 version of all of the plots can be obtained by adding 'model_type = "ggplot"
# diagnostics function allows to visualize diagnostics of a fitted RoBTT object, for example,
# the trace plot for the mean parameter in each model model
diagnostics(fit, parameter = "delta", type = "chain")

# in order to show the trace plot only for the 11th model, add show_models parameter
diagnostics(fit, parameter = "delta", type = "chain", show_models = 11)

# furthermore, the autocorrelations
diagnostics(fit, parameter = "delta", type = "autocorrelation")

# and overlying densities for each plot can also be visualize
diagnostics(fit, parameter = "delta", type = "densities")

## End(Not run)
```

fertilization *Height of 15 plant pairs collected by Darwin*

Description

The data set contains heights of self fertilized and cross fertilized plants as coded by Darwin.

Usage

```
fertilization
```

Format

A data.frame with 2 columns and 15 observations.

Value

a data.frame.

interpret *Interprets results of a 'RoBTT' model.*

Description

`interpret` creates a brief textual summary of a fitted 'RoBTT' object.

Usage

```
interpret(object)
```

Arguments

`object`	a fitted 'RoBTT' object

Value

`interpret` returns a character.

is.RoBTT *Reports whether x is a 'RoBTT' object*

Description

Reports whether x is a 'RoBTT' object

Usage

```
is.RoBTT(x)
```

Arguments

x	an object to test

Value

is.RoBTT returns a boolean.

plot.RoBTT *Plots a fitted 'RoBTT' object*

Description

plot.RoBTT allows to visualize different "RoBTT" object parameters in various ways. See type for the different model types.

Usage

```
## S3 method for class 'RoBTT'
plot(
  x,
  parameter = "mu",
  transform_rho = FALSE,
  conditional = FALSE,
  plot_type = "base",
  prior = FALSE,
  dots_prior = NULL,
  ...
)
```

Arguments

x	a fitted 'RoBTT' object
parameter	a parameter to be plotted. Defaults to "delta" (for the effect size). The additional options are "rho" (for the heterogeneity), "nu" (for the degrees of freedom).
transform_rho	whether rho parameter should be translated into log standard deviation ratio
conditional	whether conditional estimates should be plotted. Defaults to FALSE which plots the model-averaged estimates.
plot_type	whether to use a base plot "base" or ggplot2 "ggplot" for plotting. Defaults to "base".
prior	whether prior distribution should be added to figure. Defaults to FALSE.
dots_prior	list of additional graphical arguments to be passed to the plotting function of the prior distribution. Supported arguments are lwd, lty, col, and col.fill, to adjust the line thickness, line type, line color, and fill color of the prior distribution respectively.
...	list of additional graphical arguments to be passed to the plotting function. Supported arguments are lwd, lty, col, col.fill, xlab, ylab, main, xlim, ylim to adjust the line thickness, line type, line color, fill color, x-label, y-label, title, x-axis range, and y-axis range respectively.

Value

plot.RoBTT returns either NULL if plot_type = "base" or an object object of class 'ggplot2' if plot_type = "ggplot2".

See Also

RoBTT()

Examples

```
## Not run:
data("fertilization", package = "RoBTT")
fit <- RoBTT(
  x1       = fertilization$Self,
  x2       = fertilization$Crossed,
  prior_delta = prior("cauchy", list(0, 1/sqrt(2))),
  prior_rho   = prior("beta",   list(3, 3)),
  seed        = 1,
  chains      = 1,
  warmup      = 1000,
  iter        = 2000,
  control     = set_control(adapt_delta = 0.95)
)

# plot the model-averaged effect size estimate
plot(fit, parameter = "delta")
```

```
# plot prior and posterior of the conditional effect size estimate
plot(fit, parameter = "delta", conditional = TRUE, prior = TRUE)

## End(Not run)
```

`print.RoBTT` *Prints a fitted 'RoBTT' object*

Description

Prints a fitted 'RoBTT' object

Usage

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

Arguments

`x`	a fitted 'RoBTT' object.
`...`	additional arguments.

Value

`print.RoBTT` invisibly returns the print statement.

See Also

`RoBTT()`

`print.summary.RoBTT` *Prints summary object for 'RoBTT' method*

Description

Prints summary object for 'RoBTT' method

Usage

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

Arguments

`x`	a summary of a 'RoBTT' object
`...`	additional arguments

Value

`print.summary.RoBTT` invisibly returns the print statement.

See Also

`RoBTT()`

`prior` *Creates a prior distribution*

Description

`prior` creates a prior distribution. The prior can be visualized by the `plot` function.

Usage

```
prior(
  distribution,
  parameters,
  truncation = list(lower = -Inf, upper = Inf),
  prior_weights = 1
)
```

Arguments

`distribution` name of the prior distribution. The possible options are

- `"point"` for a point density characterized by a `location` parameter.
- `"normal"` for a normal distribution characterized by a `mean` and `sd` parameters.
- `"lognormal"` for a lognormal distribution characterized by a `meanlog` and `sdlog` parameters.
- `"cauchy"` for a Cauchy distribution characterized by a `location` and `scale` parameters. Internally converted into a generalized t-distribution with `df = 1`.
- `"t"` for a generalized t-distribution characterized by a `location`, `scale`, and `df` parameters.
- `"gamma"` for a gamma distribution characterized by either `shape` and `rate`, or `shape` and `scale` parameters. The later is internally converted to the `shape` and `rate` parametrization
- `"invgamma"` for an inverse-gamma distribution characterized by a `shape` and `scale` parameters. The JAGS part uses a 1/gamma distribution with a shape and rate parameter.
- `"beta"` for a beta distribution characterized by an `alpha` and `beta` parameters.
- `"exp"` for an exponential distribution characterized by either `rate` or `scale` parameter. The later is internally converted to `rate`.
- `"uniform"` for a uniform distribution defined on a range from `a` to `b`

parameters	list of appropriate parameters for a given distribution.
truncation	list with two elements, lower and upper, that define the lower and upper truncation of the distribution. Defaults to list(lower = -Inf, upper = Inf). The truncation is automatically set to the bounds of the support.
prior_weights	prior odds associated with a given distribution. The value is passed into the model fitting function, which creates models corresponding to all combinations of prior distributions for each of the model parameters and sets the model priors odds to the product of its prior distributions.

Value

prior and prior_none return an object of class 'prior'. A named list containing the distribution name, parameters, and prior weights.

See Also

plot.prior(), Normal, Lognormal, Cauchy, Beta, Exponential, LocationScaleT, InvGamma.

Examples

```
# create a standard normal prior distribution
p1 <- prior(distribution = "normal", parameters = list(mean = 1, sd = 1))

# create a half-normal standard normal prior distribution
p2 <- prior(distribution = "normal", parameters = list(mean = 1, sd = 1),
truncation = list(lower = 0, upper = Inf))

# the prior distribution can be visualized using the plot function
# (see ?plot.prior for all options)
plot(p1)
```

---

prior_none *Creates a prior distribution*

Description

prior creates a prior distribution. The prior can be visualized by the plot function.

Usage

```
prior_none(prior_weights = 1)
```

Arguments

prior_weights	prior odds associated with a given distribution. The value is passed into the model fitting function, which creates models corresponding to all combinations of prior distributions for each of the model parameters and sets the model priors odds to the product of its prior distributions.

Value

`prior` and `prior_none` return an object of class 'prior'. A named list containing the distribution name, parameters, and prior weights.

See Also

`plot.prior()`, Normal, Lognormal, Cauchy, Beta, Exponential, LocationScaleT, InvGamma.

Examples

```
# create a standard normal prior distribution
p1 <- prior(distribution = "normal", parameters = list(mean = 1, sd = 1))

# create a half-normal standard normal prior distribution
p2 <- prior(distribution = "normal", parameters = list(mean = 1, sd = 1),
truncation = list(lower = 0, upper = Inf))

# the prior distribution can be visualized using the plot function
# (see ?plot.prior for all options)
plot(p1)
```

`rho2logsdr` *rho to log standard deviation ratio transformations*

Description

A list containing the transformation function, inverse transformation function, and the jacobian function.

Usage

```
rho2logsdr
```

Format

An object of class `list` of length 3.

Value

a list with the corresponding functions

RoBTT *Estimate a Robust Bayesian T-Test*

Description

RoBTT is used to estimate a robust Bayesian t-test or truncated Bayesian t-test (if `truncation` is used). The input either requires the vector of observations for each group, `x1`, `x2`, or the summary statistics (only if the normal likelihood models are used).

Usage

```
RoBTT(
  x1 = NULL,
  x2 = NULL,
  mean1 = NULL,
  mean2 = NULL,
  sd1 = NULL,
  sd2 = NULL,
  N1 = NULL,
  N2 = NULL,
  truncation = NULL,
 prior_delta = prior(distribution = "cauchy", parameters = list(location = 0, scale =
    sqrt(2)/2)),
 prior_rho = prior(distribution = "beta", parameters = list(alpha = 1, beta = 1)),
 prior_nu = if (is.null(truncation)) prior(distribution = "exp", parameters = list(rate
    = 1)),
 prior_delta_null = prior(distribution = "spike", parameters = list(location = 0)),
 prior_rho_null = prior(distribution = "spike", parameters = list(location = 0.5)),
  prior_nu_null = prior_none(),
  prior_mu = NULL,
  prior_sigma2 = NULL,
  chains = 4,
  iter = 10000,
  warmup = 5000,
  thin = 1,
  parallel = FALSE,
  control = set_control(),
  convergence_checks = set_convergence_checks(),
  save = "all",
  seed = NULL,
  silent = TRUE,
  ...
)
```

Arguments

x1	vector of observations of the first group

x2 vector of observations of the second group

mean1 mean of the first group

mean2 mean of the first group

sd1 standard deviation of the first group

sd2 standard deviation of the first group

N1 sample size of the first group

N2 sample size of the first group

truncation an optional list specifying truncation applied to the data. Defaults to NULL, i.e., no truncation was applied and the full likelihood is applied. Alternative the truncation can be specified via a named list with:

- "x" where x is a vector of two values specifying the lower and upper truncation points common across the groups
- "x1" **and** "x2" where x1 is a vector of two values specifying the lower and upper truncation points for the first group and x2 is a vector of two values specifying the lower and upper truncation points for the second group.
- "sigma" where sigma corresponds to the number of standard deviations from the common mean where the truncation points should be set.
- "sigma1" **and** "sigma2" where sigma1 corresponds to the number of standard deviations from the mean of the first group where the truncation points should be set and sigma2 corresponds to the number of standard deviations from the mean of the second group where the truncation points should be set.

prior_delta prior distributions for the effect size delta parameter that will be treated as belonging to the alternative hypothesis. Defaults to prior(distribution = "Cauchy", parameters = list(location = 0, scale = sqrt(2)/2)).

prior_rho prior distributions for the precision allocation rho parameter that will be treated as belonging to the alternative hypothesis. Defaults to prior(distribution = "beta", parameters = list(alpha = 1, beta = 1)).

prior_nu prior distribution for the degrees of freedom + 2 nu parameter that will be treated as belonging to the alternative hypothesis. Defaults to prior(distribution = "exp", parameters = list(rate = 1)) if no truncation is specified. If truncation is specified, the default is NULL (i.e., use only normal likelihood).

prior_delta_null prior distribution for the delta parameter that will be treated as belonging to the null hypothesis. Defaults to point distribution with location at 0 (prior(distribution = "point", parameters = list(location = 0))).

prior_rho_null prior distribution for the rho parameter that will be treated as belonging to the null hypothesis. Defaults to point distribution with location at 0.5 (prior(distribution = "point", parameters = list(location = 0.5))).

prior_nu_null prior distribution for the nu parameter that will be treated as belonging to the null hypothesis. Defaults to prior_none ((i.e., normal likelihood)).

prior_mu prior distribution for the grand mean parameter. Defaults to NULL which sets Jeffreys prior for the grand mean in case of no truncation or an unit Cauchy prior distributions for the grand mean in case of truncation (which greatly improves sampling efficiency).

prior_sigma2	prior distribution for the grand variance parameter. Defaults to NULL which sets Jeffreys prior for the variance in case of no truncation or an exponential prior distribution for the variance in case of truncation (which greatly improves sampling efficiency).
chains	a number of chains of the MCMC algorithm.
iter	a number of sampling iterations of the MCMC algorithm. Defaults to 10000, with a minimum of 4000.
warmup	a number of warmup iterations of the MCMC algorithm. Defaults to 5000.
thin	a thinning of the chains of the MCMC algorithm. Defaults to 1.
parallel	whether the individual models should be fitted in parallel. Defaults to FALSE. The implementation is not completely stable and might cause a connection error.
control	allows to pass control settings with the set_control() function. See ?set_control for options and default settings.
convergence_checks	automatic convergence checks to assess the fitted models, passed with set_convergence_checks() function. See ?set_convergence_checks for options and default settings.
save	whether all models posterior distributions should be kept after obtaining a model-averaged result. Defaults to "all" which does not remove anything. Set to "min" to significantly reduce the size of final object, however, some model diagnostics and further manipulation with the object will not be possible.
seed	a seed to be set before model fitting, marginal likelihood computation, and posterior mixing for reproducibility of results. Defaults to NULL - no seed is set.
silent	whether all print messages regarding the fitting process should be suppressed. Defaults to TRUE. Note that parallel = TRUE also suppresses all messages.
...	additional arguments.

Details

See Maier et al. (2024) for more details regarding the robust Bayesian t-test methodology and the corresponding vignette (vignette("Introduction_to_RoBTT", package = "RoBTT")).

See Godmann et al. (2024) for more details regarding the truncated Bayesian t-test methodology and the corresponding vignette (vignette("Truncated_t_test", package = "RoBTT")).

Generic summary.RoBTT(), print.RoBTT(), and plot.RoBTT() functions are provided to facilitate manipulation with the ensemble.

Value

RoBTT returns an object of class "RoBTT".

References

Godmann HR, Bartoš F, Wagenmakers E (2024). "Truncating the Likelihood Allows Outlier Exclusion Without Overestimating the Evidence in the Bayes Factor t-Test." doi:10.31234/osf.io/j9f3s. PsyArxiv Preprint.

Maier M, Bartoš F, Quintana DS, van den Bergh D, Marsman M, Ly A, Wagenmakers E (2024). "Model-averaged Bayesian t-tests." *Psychonomic Bulletin & Review*. doi:10.3758/s13423024-025905.

See Also

summary.RoBTT(), prior()

Examples

```
## Not run:
# using the example data from Darwin
data("fertilization", package = "RoBTT")
fit <- RoBTT(
  x1       = fertilization$Self,
  x2       = fertilization$Crossed,
  prior_delta = prior("cauchy", list(0, 1/sqrt(2))),
  prior_rho   = prior("beta",   list(3, 3)),
  seed        = 1,
  chains      = 1,
  warmup      = 1000,
  iter        = 2000,
  control     = set_control(adapt_delta = 0.95)
)

# summary can provide many details about the model
summary(fit)

## End(Not run)
```

RoBTT_control *Convergence checks of the fitting process*

Description

Set values for the convergence checks of the fitting process.

Usage

```
set_convergence_checks(max_Rhat = 1.05, min_ESS = 500)

set_control(adapt_delta = 0.8, max_treedepth = 15, bridge_max_iter = 1000)
```

Arguments

max_Rhat	maximum value of the R-hat diagnostic. Defaults to 1.05.
min_ESS	minimum estimated sample size. Defaults to 500.
adapt_delta	tuning parameter of HMC. Defaults to 0.80.

`max_treedepth` tuning parameter of HMC. Defaults to 15.

`bridge_max_iter` maximum number of iterations for the bridge_sampler function. Defaults to `10000`

Value

`set_control` returns a list of control settings and `set_convergence_checks` returns a list of convergence checks settings.

RoBTT_options *Options for the 'RoBTT' package*

Description

A placeholder object and functions for the 'RoBTT' package. (adapted from the runjags R package).

Usage

```
RoBTT.options(...)

RoBTT.get_option(name)
```

Arguments

`...`	named option(s) to change - for a list of available options, see details below.
`name`	the name of the option to get the current value of - for a list of available options, see details below.

Value

The current value of all available 'RoBTT' options (after applying any changes specified) is returned invisibly as a named list.

---

summary.RoBTT *Summarize fitted 'RoBTT' object*

Description

`summary.RoBTT` creates summary tables for a RoBTT object.

Usage

```
## S3 method for class 'RoBTT'
summary(
  object,
  type = "ensemble",
  conditional = FALSE,
  group_estimates = FALSE,
  probs = c(0.025, 0.975),
  logBF = FALSE,
  BF01 = FALSE,
  short_name = FALSE,
  remove_spike_0 = FALSE,
  ...
)
```

Arguments

object	a fitted 'RoBTT' object
type	whether to show the overall 'RoBTT' results ("ensemble"), an overview of the individual models ("models"), an overview of the individual models MCMC diagnostics ("diagnostics"), or a detailed summary of the individual models ("individual"). Can be abbreviated to first letters.
conditional	show the conditional estimates (assuming that the alternative is true). Defaults to FALSE. Only available for type == "conditional".
group_estimates	show the model-averaged mean and standard deviation estimates for each group.
probs	quantiles of the posterior samples to be displayed. Defaults to c(.025, .975)
logBF	show log of Bayes factors. Defaults to FALSE.
BF01	show Bayes factors in support of the null hypotheses. Defaults to FALSE.
short_name	whether priors names should be shortened to the first (couple) of letters. Defaults to FALSE.
remove_spike_0	whether spike prior distributions with location at zero should be omitted from the summary. Defaults to FALSE.
...	additional arguments

Value

summary.RoBTT returns a list of tables of class 'BayesTools_table'.

See Also

RoBTT()

Examples

```
## Not run:
# using the example data from Darwin
data("fertilization", package = "RoBTT")
fit <- RoBTT(
  x1       = fertilization$Self,
  x2       = fertilization$Crossed,
  prior_delta = prior("cauchy", list(0, 1/sqrt(2))),
  prior_rho   = prior("beta",   list(3, 3)),
  seed        = 1,
  chains      = 1,
  warmup      = 1000,
  iter        = 2000,
  control     = set_control(adapt_delta = 0.95)
)

# summary can provide many details about the model
summary(fit)

# estimates from the conditional models can be obtained with
summary(fit, conditional = TRUE)

# overview of the models and their prior and posterior probability, marginal likelihood,
# and inclusion Bayes factor can be obtained with
summary(fit, type = "models")

# diagnostics overview, containing the maximum R-hat, minimum ESS, maximum MCMC error, and
# maximum MCMC error / sd across parameters for each individual model can be obtained with
summary(fit, type = "diagnostics")

# summary of individual models and their parameters can be further obtained by
summary(fit, type = "individual")

## End(Not run)
```

`update.RoBTT` *Updates a fitted RoBTT object*

Description

`update.RoBTT` can be used to

1. change the prior odds of fitted models by specifying a vector `prior_weights` of the same length as the fitted models,
2. refitting models that failed to converge with updated settings of control parameters,
3. or changing the convergence criteria and recalculating the ensemble results by specifying new `control` argument and setting `refit_failed == FALSE`.

Usage

```
## S3 method for class 'RoBTT'
update(
  object,
  refit_failed = TRUE,
  prior_weights = NULL,
  chains = NULL,
  iter = NULL,
  warmup = NULL,
  thin = NULL,
  parallel = NULL,
  control = NULL,
  convergence_checks = NULL,
  save = "all",
  seed = NULL,
  silent = TRUE,
  ...
)
```

Arguments

object	a fitted RoBTT object
refit_failed	whether failed models should be refitted. Relevant only prior_weights are not supplied. Defaults to TRUE.
prior_weights	either a single value specifying prior model weight of a newly specified model using priors argument, or a vector of the same length as already fitted models to update their prior weights.
chains	a number of chains of the MCMC algorithm.
iter	a number of sampling iterations of the MCMC algorithm. Defaults to 10000, with a minimum of 4000.
warmup	a number of warmup iterations of the MCMC algorithm. Defaults to 5000.
thin	a thinning of the chains of the MCMC algorithm. Defaults to 1.
parallel	whether the individual models should be fitted in parallel. Defaults to FALSE. The implementation is not completely stable and might cause a connection error.
control	allows to pass control settings with the set_control() function. See ?set_control for options and default settings.
convergence_checks	automatic convergence checks to assess the fitted models, passed with set_convergence_checks() function. See ?set_convergence_checks for options and default settings.
save	whether all models posterior distributions should be kept after obtaining a model-averaged result. Defaults to "all" which does not remove anything. Set to "min" to significantly reduce the size of final object, however, some model diagnostics and further manipulation with the object will not be possible.
seed	a seed to be set before model fitting, marginal likelihood computation, and posterior mixing for reproducibility of results. Defaults to NULL - no seed is set.

silent	whether all print messages regarding the fitting process should be suppressed. Defaults to TRUE. Note that parallel = TRUE also suppresses all messages.
...	additional arguments.

Details

See RoBTT() for more details.

Value

RoBTT returns an object of class 'RoBTT'.

See Also

RoBTT(), summary.RoBTT(), prior(), check_setup()

Index

∗ **datasets**
fertilization, 8
rho2logsdr, 14
∗ **package**
RoBTT-package, 2

Beta, *13, 14*
bridge_sampler, *19*

Cauchy, *13, 14*
check_RoBTT, 3
check_setup, 4
check_setup(), *23*
class, *17*

diagnostics, 5
diagnostics_autocorrelation *(*diagnostics*)*, 5
diagnostics_density *(*diagnostics*)*, 5
diagnostics_trace *(*diagnostics*)*, 5

Exponential, *13, 14*

fertilization, 8

interpret, 8
InvGamma, *13, 14*
is.RoBTT, 9

LocationScaleT, *13, 14*
Lognormal, *13, 14*

Normal, *13, 14*

par, *7*
plot.prior(), *13, 14*
plot.RoBTT, 9
plot.RoBTT(), *17*
print.RoBTT, 11
print.RoBTT(), *17*
print.summary.RoBTT, 11

prior, 12
prior(), *18*, *23*
prior_none, 13

rho2logsdr, 14
RoBTT, 15
RoBTT(), *5*, *7*, *10–12*, *20*, *23*
RoBTT-package, 2
RoBTT.get_option *(*RoBTT_options*)*, 19
RoBTT.options *(*RoBTT_options*)*, 19
RoBTT.package *(*RoBTT-package*)*, 2
RoBTT_control, 18
RoBTT_options, 19
RoBTT_package *(*RoBTT-package*)*, 2

set_control *(*RoBTT_control*)*, 18
set_control(), *17*, *22*
set_control, *(*RoBTT_control*)*, 18
set_convergence_checks *(*RoBTT_control*)*, 18
set_convergence_checks(), *17*, *22*
stan_plot, *7*
summary.RoBTT, 19
summary.RoBTT(), *7*, *17*, *18*, *23*

update.RoBTT, 21