Package 'RoBTT'

March 13, 2023

Title Robust Bayesian T-Test

Version 1.0.1

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

Description An implementation of Bayesian model-averaged t-test that allows
users to draw inference about the presence vs absence of the effect,
heterogeneity of variances, and outliers. The 'RoBTT' packages estimates model
ensembles of models created as a combination of the competing hypotheses and uses
Bayesian model-averaging to combine the models using posterior model probabilities.
Users can obtain the model-averaged posterior distributions and inclusion Bayes
factors which account for the uncertainty in the data generating process
(Maier et al., 2022, <doi:10.31234/osf.io/d5zwc>).
Users can define a wide range of informative priors for all parameters
of interest. The package provides convenient functions for summary, visualizations,
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 7.2.1

NeedsCompilation yes

ByteCompile true

LinkingTo StanHeaders (>= 2.18.1), rstan (>= 2.21.2), 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(>= 2.21.2), rstantools(>= 1.5.0), RcppParallel (>=
5.0.1), BayesTools (>= 0.2.12), bridgesampling, methods, Rdpack

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

RdMacros Rdpack

VignetteBuilder knitr

Author František Bartoš [aut, cre] (<<https://orcid.org/0000-0002-0018-5573>>),
      Maximilian Maier [aut] (<<https://orcid.org/0000-0002-9873-6096>>)

Repository CRAN

Date/Publication 2023-03-13 16:50:02 UTC

R topics documented:

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. (2022) 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 (2022). "Model-averaged Bayesian t-tests." doi:10.31234/osf.io/d5zwc, PsyArxiv Preprint.

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.

---

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",
  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).
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

```
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)),
  likelihood  = "normal",
  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)
```

---

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)
```

RoBTT   *Estimate a Robust Bayesian T-Test*

Description

RoBTT is used to estimate a Robust Bayesian T-Test. The input either requires the vector of obser-
vations for each group, x1, x2, or the summary statistics (in case only the "normal" likelihood is
used).

Usage

```
RoBTT(
  x1 = NULL,
  x2 = NULL,
  mean1 = NULL,
  mean2 = NULL,
  sd1 = NULL,
  sd2 = NULL,
  N1 = NULL,
  N2 = 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 = 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 = NULL,
  likelihood = c("normal", if (!is.null(prior_nu)) "t"),
  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
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))`.
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 NULL (`NULL`, i.e., normal distribution).
likelihood	types of likelihood to be used. Defaults to `c("normal", "t")` for a normal and t likelihoods.
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. (2022) for more details regarding the methodology.

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

Maier M, Bartoš F, Quintana DS, van den Bergh D, Marsman M, Ly A, Wagenmakers E (2022). "Model-averaged Bayesian t-tests." doi:10.31234/osf.io/d5zwc, PsyArxiv Preprint.

See Also

summary.RoBTT(), prior()

Examples

```
# 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)),
  likelihood  = "normal",
  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)
```

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()](RoBTT)

Examples

```
# 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)),
  likelihood = "normal",
  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")
```

Index