

Package ‘bayestestR’

April 8, 2019

Type Package

Title Understand and Describe Bayesian Models and Posterior Distributions

Version 0.1.0

Maintainer Dominique Makowski <dom.makowski@gmail.com>

URL <https://github.com/easystats/bayestestR>

BugReports <https://github.com/easystats/bayestestR/issues>

Description Provides utilities to describe posterior distributions and Bayesian models. It includes point-estimates such as Maximum A Posteriori (MAP), measures of dispersion (Highest Density Interval - HDI; Kruschke, 2014 <doi:10.1016/B978-0-12-405888-0.09999-2>) and indices used for null-hypothesis testing (such as ROPE percentage and pd).

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 3.0), stats

Imports insight

Suggests brms, broom, covr, dplyr, tidyr, ggplot2, ggridges, knitr, rmarkdown, rstanarm, stringr, testthat

RoxygenNote 6.1.1

Language en-GB

VignetteBuilder knitr

NeedsCompilation no

Author Dominique Makowski [aut, cre] (<<https://orcid.org/0000-0001-5375-9967>>), Daniel Lüdtke [aut] (<<https://orcid.org/0000-0002-8895-3206>>)

Repository CRAN

Date/Publication 2019-04-08 09:42:41 UTC

R topics documented:

as.data.frame.density	2
ci	2
density_at	4
equivalence_test	5
hdi	7
map_estimate	9
p_direction	11
p_map	12
p_rope	14
rnorm_perfect	15
rope	16
rope_range	18
Index	20

as.data.frame.density *Coerce to a Data Frame*

Description

Functions to check if an object is a data frame, or coerce it if possible.

Usage

```
## S3 method for class 'density'
as.data.frame(x, ...)
```

Arguments

x Any R object.
 ... Additional arguments to be passed to or from methods.

ci *Confidence/Credible Interval*

Description

Compute Confidence/Credible Intervals (CI) for Bayesian and frequentist models using quantiles.

Usage

```

ci(x, ...)

## S3 method for class 'numeric'
ci(x, ci = 0.9, verbose = TRUE, ...)

## S3 method for class 'stanreg'
ci(x, ci = 0.9, effects = c("fixed", "random",
  "all"), parameters = NULL, verbose = TRUE, ...)

## S3 method for class 'brmsfit'
ci(x, ci = 0.9, effects = c("fixed", "random",
  "all"), component = c("conditional", "zi", "zero_inflated", "all"),
  parameters = NULL, verbose = TRUE, ...)

```

Arguments

x	A stanreg or brmsfit model , or a vector representing a posterior distribution.
...	Currently not used.
ci	Value or vector of probability of the interval (between 0 and 1) to be estimated. Named Credible Interval (CI) for consistency.
verbose	Toggle off warnings.
effects	Should results for fixed effects, random effects or both be returned? Only applies to mixed models. May be abbreviated.
parameters	Regular expression pattern that describes the parameters that should be returned. Meta-parameters (like lp__ or prior_) are filtered by default, so only parameters that typically appear in the summary() are returned. Use parameters to select specific parameters for the output.
component	Should results for all parameters, parameters for the conditional model or the zero-inflated part of the model be returned? May be abbreviated. Only applies to brms -models.

Details

Documentation is accessible for:

- [Bayesian models](#)

Value

A data frame with following columns:

- Parameter The model parameter(s), if x is a model-object. If x is a vector, this column is missing.
- CI The probability of the credible interval.
- CI_low , CI_high The lower and upper credible interval limits for the parameters.

Examples

```
library(bayestestR)

ci(rnorm(1000))
## Not run:
library(rstanarm)
model <- rstanarm::stan_glm(mpg ~ wt + cyl, data = mtcars)
ci(model)
ci(model, ci = c(.80, .90, .95))

library(brms)
model <- brms::brm(mpg ~ wt + cyl, data = mtcars)
ci(model)
ci(model, ci = c(.80, .90, .95))

## End(Not run)
```

density_at

Probability of a Given Point

Description

Compute the density of a given point of a distribution.

Usage

```
density_at(posterior, x, precision = 2^10)
```

Arguments

posterior	Vector representing a posterior distribution.
x	The value of which to get the approximate probability.
precision	Number of points for density estimation. See the n parameter in density .

Examples

```
library(bayestestR)
posterior <- rnorm_perfect(n = 10)
density_at(posterior, 0)
density_at(posterior, c(0, 1))
```

equivalence_test	<i>Test for Practical Equivalence</i>
------------------	---------------------------------------

Description

Perform a **Test for Practical Equivalence** based on the "*HDI+ROPE decision rule*" (Kruschke, 2018) to check whether parameter values should be accepted or rejected against an explicitly formulated "null hypothesis" (*i.e.*, a [ROPE](#)).

Usage

```
equivalence_test(x, ...)

## S3 method for class 'numeric'
equivalence_test(x, range = "default", ci = 0.95,
  verbose = TRUE, ...)

## S3 method for class 'stanreg'
equivalence_test(x, range = "default", ci = 0.95,
  parameters = NULL, verbose = TRUE, ...)

## S3 method for class 'brmsfit'
equivalence_test(x, range = "default", ci = 0.95,
  parameters = NULL, verbose = TRUE, ...)
```

Arguments

x	Vector representing a posterior distribution. Can also be a stanreg or brmsfit model.
...	Currently not used.
range	ROPE's lower and higher bounds. Should be a vector of length two (e.g., <code>c(-0.1, 0.1)</code>) or "default". If "default", the range is set to <code>c(-0.1, 0.1)</code> if input is a vector, and <code>x +/- 0.1*SD(response)</code> if a Bayesian model is provided.
ci	The Credible Interval (CI) probability, corresponding to the proportion of HDI, to use.
verbose	Toggle off warnings.
parameters	Regular expression pattern that describes the parameters that should be returned. Meta-parameters (like <code>lp__</code> or <code>prior_</code>) are filtered by default, so only parameters that typically appear in the <code>summary()</code> are returned. Use <code>parameters</code> to select specific parameters for the output.

Details

Using the [ROPE](#) and the [HDI](#), Kruschke (2018) suggest using the percentage of the 95% (or 90%, considered more stable) [HDI](#) that falls within the [ROPE](#) as a decision rule. If the [HDI](#) is completely outside the [ROPE](#), the "null hypothesis" for this parameter is "rejected". If the [ROPE](#) completely covers the [HDI](#), i.e. all most credible values of a parameter are inside the region of practical equivalence, the null hypothesis is accepted. Else, it's undecided whether to accept or reject the null hypothesis. If the full [ROPE](#) is used (i.e., 100% of the [HDI](#)), then the null hypothesis is rejected or accepted if the percentage of the posterior within the [ROPE](#) is smaller than to 2.5% or greater than 97.5%. Desirable results are low proportions inside the [ROPE](#) (the closer to zero the better) and the null hypothesis should be rejected.

Some attention is required for finding suitable values for the [ROPE](#) limits (argument range). See 'Details' in [rope_range](#) for further information.

Value

A data frame with following columns:

- Parameter The model parameter(s), if x is a model-object. If x is a vector, this column is missing.
- CI The probability of the [HDI](#).
- ROPE_low, ROPE_high The limits of the [ROPE](#). These values are identical for all parameters.
- ROPE_Percentage The proportion of the [HDI](#) that lies inside the [ROPE](#).
- ROPE_Equivalence The "test result", as character. Either "rejected", "accepted" or "undecided".
- HDI_low , HDI_high The lower and upper [HDI](#) limits for the parameters.

Note

There is a `print()`-method with a `digits`-argument to control the amount of digits in the output, and there is a `plot()`-method to visualize the results from the equivalence-test (for models only).

References

Kruschke, J. K. (2018). Rejecting or accepting parameter values in Bayesian estimation. *Advances in Methods and Practices in Psychological Science*, 1(2), 270-280. doi: [10.1177/2515245918771304](https://doi.org/10.1177/2515245918771304).

Examples

```
library(bayestestR)

equivalence_test(x = rnorm(1000, 0, 0.01), range = c(-0.1, 0.1))
equivalence_test(x = rnorm(1000, 0, 1), range = c(-0.1, 0.1))
equivalence_test(x = rnorm(1000, 1, 0.01), range = c(-0.1, 0.1))
equivalence_test(x = rnorm(1000, 1, 1), ci = c(.50, .99))

# print more digits
test <- equivalence_test(x = rnorm(1000, 1, 1), ci = c(.50, .99))
```

```

print(test, digits = 4)

## Not run:
library(rstanarm)
model <- rstanarm::stan_glm(mpg ~ wt + cyl, data = mtcars)
equivalence_test(model)
equivalence_test(model, ci = c(.50, 1))

# plot result
test <- equivalence_test(model)
plot(test)

library(brms)
model <- brms::brm(mpg ~ wt + cyl, data = mtcars)
equivalence_test(model)
equivalence_test(model, ci = c(.50, .99))

## End(Not run)

```

hdi

Highest Density Interval (HDI)

Description

Compute the **Highest Density Interval (HDI)** of a posterior distribution, i.e., all points within the interval have a higher probability density than points outside the interval. The HDI can be used in the context of Bayesian posterior characterisation as **Credible Interval (CI)**.

Usage

```

hdi(x, ...)

## S3 method for class 'numeric'
hdi(x, ci = 0.9, verbose = TRUE, ...)

## S3 method for class 'stanreg'
hdi(x, ci = 0.9, effects = c("fixed", "random",
  "all"), parameters = NULL, verbose = TRUE, ...)

## S3 method for class 'brmsfit'
hdi(x, ci = 0.9, effects = c("fixed", "random",
  "all"), component = c("conditional", "zi", "zero_inflated", "all"),
  parameters = NULL, verbose = TRUE, ...)

```

Arguments

x Vector representing a posterior distribution. Can also be a stanreg or brmsfit model.

...	Currently not used.
ci	Value or vector of probability of the interval (between 0 and 1) to be estimated. Named Credible Interval (CI) for consistency.
verbose	Toggle off warnings.
effects	Should results for fixed effects, random effects or both be returned? Only applies to mixed models. May be abbreviated.
parameters	Regular expression pattern that describes the parameters that should be returned. Meta-parameters (like <code>lp_</code> or <code>prior_</code>) are filtered by default, so only parameters that typically appear in the <code>summary()</code> are returned. Use <code>parameters</code> to select specific parameters for the output.
component	Should results for all parameters, parameters for the conditional model or the zero-inflated part of the model be returned? May be abbreviated. Only applies to brms -models.

Details

Unlike equal-tailed intervals (see `ci`) that typically exclude 2.5% from each tail of the distribution, the HDI is *not* equal-tailed and therefore always includes the mode(s) of posterior distributions.

By default, `hdi()` returns the 90% intervals (`ci = 0.9`), deemed to be more stable than, for instance, 95% intervals (*Kruschke, 2015*). An effective sample size of at least 10.000 is recommended if 95% intervals should be computed (*Kruschke, 2015, p. 183ff*).

Value

A data frame with following columns:

- Parameter The model parameter(s), if `x` is a model-object. If `x` is a vector, this column is missing.
- CI The probability of the HDI.
- CI_low , CI_high The lower and upper HDI limits for the parameters.

Author(s)

Credits go to [ggdistribute](#) and [HDInterval](#).

References

Kruschke, J. (2015). Doing Bayesian data analysis: A tutorial with R, JAGS, and Stan. Academic Press.

Examples

```
library(bayestestR)

posterior <- rnorm(1000)
hdi(posterior, ci = .90)
hdi(posterior, ci = c(.80, .90, .95))
```



```
## Not run:
library(rstanarm)
model <- rstanarm::stan_glm(mpg ~ wt + cyl, data = mtcars)
hdi(model)
hdi(model, ci = c(.80, .90, .95))

library(brms)
model <- brms::brm(mpg ~ wt + cyl, data = mtcars)
hdi(model)
hdi(model, ci = c(.80, .90, .95))

## End(Not run)
```

map_estimate

Maximum A Posteriori (MAP) Estimate

Description

Find the **Highest Maximum A Posteriori (MAP)** estimate of a posterior, *i.e.*, the most probable value. It corresponds to the "peak" (or the *mode*) of the posterior distribution. This function returns a dataframe containing the MAP value. If the density is set to TRUE, it will include a second column containing the *probability* (*i.e.*, the value of the estimated density function) associated with the MAP (the value of the y axis of the density curve at the MAP).

Usage

```
map_estimate(x, ...)

## S3 method for class 'numeric'
map_estimate(x, precision = 2^10, density = FALSE,
  ...)

## S3 method for class 'stanreg'
map_estimate(x, precision = 2^10,
  effects = c("fixed", "random", "all"), parameters = NULL,
  density = FALSE, ...)

## S3 method for class 'brmsfit'
map_estimate(x, precision = 2^10,
  effects = c("fixed", "random", "all"), component = c("conditional",
  "zi", "zero_inflated", "all"), parameters = NULL, density = FALSE,
  ...)
```

Arguments

x Vector representing a posterior distribution. Can also be a stanreg or brmsfit model.

...	Currently not used.
precision	Number of points for density estimation. See the n parameter in density .
density	Turning this parameter
effects	Should results for fixed effects, random effects or both be returned? Only applies to mixed models. May be abbreviated.
parameters	Regular expression pattern that describes the parameters that should be returned. Meta-parameters (like lp__ or prior_) are filtered by default, so only parameters that typically appear in the summary() are returned. Use parameters to select specific parameters for the output.
component	Should results for all parameters, parameters for the conditional model or the zero-inflated part of the model be returned? May be abbreviated. Only applies to brms -models.

Value

A numeric value if posterior is a vector and density = FALSE. If density = TRUE, or if posterior is a model-object, returns a data frame with following columns:

- Parameter The model parameter(s), if x is a model-object. If x is a vector, this column is missing.
- MAP The MAP estimate for the posterior or each model parameter.
- MAP_density

Examples

```
library(bayestestR)

posterior <- rnorm(10000)

map_estimate(posterior)
map_estimate(posterior, density = TRUE)

plot(density(posterior))
abline(v=map_estimate(posterior),
       col="red") # The x coordinate of MAP
abline(h=map_estimate(posterior, density = TRUE)$MAP_density,
       col="blue") # The y coordinate of MAP

## Not run:
library(rstanarm)
model <- rstanarm::stan_glm(mpg ~ wt + cyl, data = mtcars)
map_estimate(model)

library(brms)
model <- brms::brm(mpg ~ wt + cyl, data = mtcars)
map_estimate(model)

## End(Not run)
```

p_direction *Probability of Direction (pd)*

Description

Compute the **Probability of Direction** (*pd*, also known as the Maximum Probability of Effect - *MPE*). It varies between 50% and 100% and can be interpreted as the probability (expressed in percentage) that a parameter (described by its posterior distribution) is strictly positive or negative (whichever is the most probable). It is mathematically defined as the proportion of the posterior distribution that is of the median's sign. Although differently expressed, this index is fairly similar (*i.e.*, is strongly correlated) to the frequentist *p-value*.

Usage

```
p_direction(posterior, ...)

## S3 method for class 'numeric'
p_direction(posterior, ...)

## S3 method for class 'stanreg'
p_direction(posterior, effects = c("fixed", "random",
  "all"), parameters = NULL, ...)

## S3 method for class 'brmsfit'
p_direction(posterior, effects = c("fixed", "random",
  "all"), component = c("conditional", "zi", "zero_inflated", "all"),
  parameters = NULL, ...)

pd(posterior, ...)
```

Arguments

posterior	Vector representing a posterior distribution. Can also be a stanreg or brmsfit model.
...	Currently not used.
effects	Should results for fixed effects, random effects or both be returned? Only applies to mixed models. May be abbreviated.
parameters	Regular expression pattern that describes the parameters that should be returned. Meta-parameters (like lp__ or prior_) are filtered by default, so only parameters that typically appear in the summary() are returned. Use parameters to select specific parameters for the output.
component	Should results for all parameters, parameters for the conditional model or the zero-inflated part of the model be returned? May be abbreviated. Only applies to brms -models.

Details

Relationship with the p-value: In most cases, it seems that the *pd* corresponds to the frequentist one-sided *p*-value through the formula $p_{\text{twosided}} = 2 * (1 - \frac{pd}{100})$ and to the two-sided *p*-value (*the most commonly reported one*) through the formula $p_{\text{onesided}} = 2 * (1 - \frac{pd}{100})$. Thus, a *pd* of 95%, 97.5%, 99.5% and 99.95% corresponds approximately to a *p*-value of respectively .1, .05, .01 and .001. See the *reporting guidelines*.

Examples

```
library(bayestestR)

# Simulate a posterior distribution of mean 1 and SD 1
posterior <- rnorm(1000, mean = 1, sd = 1)
p_direction(posterior)
## Not run:
library(rstanarm)
model <- rstanarm::stan_glm(mpg ~ wt + cyl, data = mtcars)
p_direction(model)

library(brms)
model <- brms::brm(mpg ~ wt + cyl, data = mtcars)
p_direction(model)

## End(Not run)
```

p_map

Bayesian p-value based on the density at the Maximum A Priori (MAP)

Description

Compute a Bayesian equivalent of the *p*-value, related to the odds that a parameter (described by its posterior distribution) has against the null hypothesis (*H*₀) using Mills' (2014, 2017) *Objective Bayesian Hypothesis Testing* framework. It is mathematically based on the density at the Maximum A Priori (MAP) and corresponds to the density value at 0 divided by the density of the MAP estimate.

Usage

```
p_map(posterior, ...)

## S3 method for class 'numeric'
p_map(posterior, precision = 2^10, ...)

## S3 method for class 'stanreg'
p_map(posterior, precision = 2^10,
      effects = c("fixed", "random", "all"), parameters = NULL, ...)
```

```
## S3 method for class 'brmsfit'
p_map(posterior, precision = 2^10,
      effects = c("fixed", "random", "all"), component = c("conditional",
      "zi", "zero_inflated", "all"), parameters = NULL, ...)
```

Arguments

posterior	Vector representing a posterior distribution. Can also be a stanreg or brmsfit model.
...	Currently not used.
precision	Number of points for density estimation. See the n-parameter in density .
effects	Should results for fixed effects, random effects or both be returned? Only applies to mixed models. May be abbreviated.
parameters	Regular expression pattern that describes the parameters that should be returned. Meta-parameters (like lp__ or prior_) are filtered by default, so only parameters that typically appear in the summary() are returned. Use parameters to select specific parameters for the output.
component	Should results for all parameters, parameters for the conditional model or the zero-inflated part of the model be returned? May be abbreviated. Only applies to brms -models.

References

[Mill's talk](#)

Examples

```
library(bayestestR)

p_map(posterior = rnorm(1000, 0, 1))
p_map(posterior = rnorm(1000, 10, 1))
## Not run:
library(rstanarm)
model <- rstanarm::stan_glm(mpg ~ wt + cyl, data = mtcars)
p_map(model)

library(brms)
model <- brms::brm(mpg ~ wt + cyl, data = mtcars)
p_map(model)

## End(Not run)
```

p_rope

*ROPE-based p-value***Description**

Compute the ROPE-based p-value, an exploratory index representing the maximum percentage of HDI that does not contain (positive values) or is entirely contained (negative values) in the negligible values space defined by the ROPE. It differs from the ROPE percentage, *i.e.*, from the proportion of a given CI in the ROPE, as it represents the maximum CI to reach a ROPE proportion of 0% (positive values) or 100% (negative values). A ROPE-based p of 97% means that there is a probability of .97 that a parameter (described by its posterior distribution) is outside the ROPE. On the contrary, a ROPE-based p of -97% means that there is a probability of .97 that the parameter is inside the ROPE.

Usage

```
p_rope(posterior, ...)

## S3 method for class 'numeric'
p_rope(posterior, range = "default", precision = 0.1,
      ...)

## S3 method for class 'stanreg'
p_rope(posterior, range = "default", precision = 0.1,
      effects = c("fixed", "random", "all"), parameters = NULL, ...)

## S3 method for class 'brmsfit'
p_rope(posterior, range = "default", precision = 0.1,
      effects = c("fixed", "random", "all"), component = c("conditional",
      "zi", "zero_inflated", "all"), parameters = NULL, ...)
```

Arguments

posterior	Vector representing a posterior distribution. Can also be a ‘stanreg’ or ‘brmsfit’ model.
...	Currently not used.
range	ROPE’s lower and higher bounds. Should be a list of two values (e.g., <code>c(-0.1, 0.1)</code>) or “default”. If “default”, the range is set to <code>c(0.1, 0.1)</code> if input is a vector and <code>x +/- 0.1*SD(response)</code> if a Bayesian model is provided.
precision	The precision by which to explore the ROPE space (in percentage). Lower values increase the precision of the returned p value but can be quite computationally costly.
effects	Should results for fixed effects, random effects or both be returned? Only applies to mixed models. May be abbreviated.

parameters	Regular expression pattern that describes the parameters that should be returned. Meta-parameters (like lp__ or prior_) are filtered by default, so only parameters that typically appear in the summary() are returned. Use parameters to select specific parameters for the output.
component	Should results for all parameters, parameters for the conditional model or the zero-inflated part of the model be returned? May be abbreviated. Only applies to brms -models.

Examples

```
library(bayestestR)

p_rope(posterior = rnorm(1000, mean = 1, sd = 1), range = c(-0.1, 0.1))
## Not run:
library(rstanarm)
model <- rstanarm::stan_glm(mpg ~ wt + cyl, data = mtcars)
p_rope(model)

library(brms)
model <- brms::brm(mpg ~ wt + cyl, data = mtcars)
p_rope(model)

## End(Not run)
```

rnorm_perfect *The (Perfect) Normal Distribution*

Description

Generate a sample of size n with a near-perfect normal distribution.

Usage

```
rnorm_perfect(n, mean = 0, sd = 1)
```

Arguments

n	Number of observations. If $\text{length}(n) > 1$, the length is taken to be the number required.
mean	Vector of means.
sd	Vector of standard deviations.

Examples

```
library(bayestestR)
x <- rnorm_perfect(n = 10)
plot(density(x))
```

rope

*Region of Practical Equivalence (ROPE)***Description**

Compute the proportion (in percentage) of the HDI (default to the 90% HDI) of a posterior distribution that lies within a region of practical equivalence.

Usage

```
rope(x, ...)

## S3 method for class 'numeric'
rope(x, range = "default", ci = 0.9,
      verbose = TRUE, ...)

## S3 method for class 'stanreg'
rope(x, range = "default", ci = 0.9,
      effects = c("fixed", "random", "all"), parameters = NULL,
      verbose = TRUE, ...)

## S3 method for class 'brmsfit'
rope(x, range = "default", ci = 0.9,
      effects = c("fixed", "random", "all"), component = c("conditional",
        "zi", "zero_inflated", "all"), parameters = NULL, verbose = TRUE,
      ...)
```

Arguments

x	Vector representing a posterior distribution. Can also be a stanreg or brmsfit model.
...	Currently not used.
range	ROPE's lower and higher bounds. Should be a vector of length two (e.g., <code>c(-0.1, 0.1)</code>) or "default". If "default", the range is set to <code>c(-0.1, 0.1)</code> if input is a vector, and <code>x +/- 0.1*SD(response)</code> if a Bayesian model is provided.
ci	The Credible Interval (CI) probability, corresponding to the proportion of HDI, to use.
verbose	Toggle off warnings.
effects	Should results for fixed effects, random effects or both be returned? Only applies to mixed models. May be abbreviated.
parameters	Regular expression pattern that describes the parameters that should be returned. Meta-parameters (like <code>lp__</code> or <code>prior_</code>) are filtered by default, so only parameters that typically appear in the <code>summary()</code> are returned. Use <code>parameters</code> to select specific parameters for the output.

component Should results for all parameters, parameters for the conditional model or the zero-inflated part of the model be returned? May be abbreviated. Only applies to **brms**-models.

Details

Statistically, the probability of a posterior distribution of being different from 0 does not make much sense (the probability of it being different from a single point being infinite). Therefore, the idea underlining ROPE is to let the user define an area around the null value enclosing values that are *equivalent to the null* value for practical purposes (Kruschke 2010, 2011, 2014).

Kruschke (2018) suggests that such null value could be set, by default, to the -0.1 to 0.1 range of a standardized parameter (negligible effect size according to Cohen, 1988). This could be generalized: For instance, for linear models, the ROPE could be set as $\theta \pm .1 * sd(y)$. This ROPE range can be automatically computed for models using the `rope_range` function.

Kruschke (2010, 2011, 2014) suggests using the proportion of the 95% (or 90%, considered more stable) HDI that falls within the ROPE as an index for "null-hypothesis" testing (as understood under the Bayesian framework, see `equivalence_test`).

References

- Cohen, J. (1988). Statistical power analysis for the behavioural sciences.
- Kruschke, J. K. (2010). What to believe: Bayesian methods for data analysis. Trends in cognitive sciences, 14(7), 293-300. doi: [10.1016/j.tics.2010.05.001](https://doi.org/10.1016/j.tics.2010.05.001).
- Kruschke, J. K. (2011). Bayesian assessment of null values via parameter estimation and model comparison. Perspectives on Psychological Science, 6(3), 299-312. doi: [10.1177/1745691611406925](https://doi.org/10.1177/1745691611406925).
- Kruschke, J. K. (2014). Doing Bayesian data analysis: A tutorial with R, JAGS, and Stan. Academic Press. doi: [10.1177/2515245918771304](https://doi.org/10.1177/2515245918771304).
- Kruschke, J. K. (2018). Rejecting or accepting parameter values in Bayesian estimation. Advances in Methods and Practices in Psychological Science, 1(2), 270-280. doi: [10.1177/2515245918771304](https://doi.org/10.1177/2515245918771304).

Examples

```
library(bayestestR)

rope(x = rnorm(1000, 0, 0.01), range = c(-0.1, 0.1))
rope(x = rnorm(1000, 0, 1), range = c(-0.1, 0.1))
rope(x = rnorm(1000, 1, 0.01), range = c(-0.1, 0.1))
rope(x = rnorm(1000, 1, 1), ci = c(.90, .95))
## Not run:
library(rstanarm)
model <- rstanarm::stan_glm(mpg ~ wt + cyl, data = mtcars)
rope(model)
rope(model, ci = c(.90, .95))

library(brms)
```

```

model <- brms::brm(mpg ~ wt + cyl, data = mtcars)
rope(model)
rope(model, ci = c(.90, .95))

## End(Not run)

```

rope_range

Find Default Equivalence (ROPE) Region Bounds

Description

This function attempts at automatically finding suitable "default" values for the Region Of Practical Equivalence (ROPE). Kruschke (2018) suggests that such null value could be set, by default, to a range from -0.1 to 0.1 of a standardized parameter (negligible effect size according to Cohen, 1988), which can be generalised for linear models to $[-0.1 * SD_y, 0.1 * SD_y]$.

For logistic models, the parameters expressed in log odds ratio can be converted to standardized difference through the formula $\sqrt{3}/\pi$, resulting in a range of -0.055 to 0.055 .

For other models with binary outcome, it is strongly recommended to manually specify the rope argument. Currently, the same default is applied that for logistic models.

For all other models, -0.1 , 0.1 is used to determine the ROPE limits.

Usage

```
rope_range(x)
```

Arguments

x Vector representing a posterior distribution. Can also be a stanreg or brmsfit model.

References

Kruschke, J. K. (2018). Rejecting or accepting parameter values in Bayesian estimation. *Advances in Methods and Practices in Psychological Science*, 1(2), 270-280. doi: [10.1177/2515245918771304](https://doi.org/10.1177/2515245918771304).

Examples

```

## Not run:
library(rstanarm)
model <- rstanarm::stan_glm(vs ~ mpg, data = mtcars, family = "binomial")
rope_range(model)

library(brms)
model <- brms::brm(mpg ~ wt + cyl, data = mtcars)
rope_range(model)

```

rope_range

19

End(Not run)

Index

`as.data.frame.density`, 2

`ci`, 2, 8

`density`, 4, 10, 13

`density_at`, 4

`equivalence_test`, 5, 17

HDI, 6, 14, 17

`hdi`, 7

`map_estimate`, 9

`p_direction`, 11

`p_map`, 12

`p_rope`, 14

`pd(p_direction)`, 11

`rnorm_perfect`, 15

ROPE, 5, 6, 14

`rope`, 16

`rope_range`, 6, 17, 18