Package 'baymedr'

October 21, 2019

Title Computation of Bayes Factors for Common Biomedical Designs

Version 0.1.0

Date 2019-10-17

Description BAYesian inference for MEDical designs in R. Convenience functions for the computation of Bayes factors for common biomedical research designs. Implemented are functions to test the equivalence (equiv_bf), non-inferiority (infer_bf), and superiority (super_bf) of an experimental group compared to a control group. Bayes factors for these three tests can be computed based on raw data (x, y) or summary statistics (n_x, n_y, mean_x, mean_y, sd_x, sd_y [or ci_margin and ci_level]), making it possible to reanalyse findings (e.g., from publications) without the need to obtain the raw data.

Depends R (>= 3.2.0)

Imports methods, rlang, stats, stringr

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

RoxygenNote 6.1.1

License GPL-3

Encoding UTF-8

LazyData true

URL https://github.com/maxlinde/baymedr

BugReports https://github.com/maxlinde/baymedr/issues

NeedsCompilation no

Author Maximilian Linde [aut, cre] (<https://orcid.org/0000-0001-8421-090X>), Don van Ravenzwaaij [aut] (<https://orcid.org/0000-0002-5030-4091>), Quentin F. Gronau [ctb] (<https://orcid.org/0000-0001-5510-6943>)

Maintainer Maximilian Linde <maximilian.linde.92@gmail.com>

Repository CRAN

Date/Publication 2019-10-21 10:50:02 UTC

R topics documented:

baymedr-package *baymedr: Computation of Bayes Factors for Common Biomedical Designs*

Description

`baymedr` provides functions for the computation of Bayes factors for common biomedical research designs. At this point in time, functions for equivalence, non-inferiority, and superiority designs are implemented.

Details

Package:	baymedr
Type:	Package
Version:	0.1.0
License:	GPL-3
Date:	2019-10-17

At this point in time, `baymedr` entails tests for the following research designs:

- Equivalence: `equiv_bf`
- Non-Inferiority: `infer_bf`
- Superiority: `super_bf`

Author(s)

Maintainer: Maximilian Linde – `<maximilian.linde.92@gmail.com>`

Authors:

- Maximilian Linde (aut, cre)
- Don van Ravenzwaaij (aut)
- Quentin F. Gronau (ctb)

See Also

Useful links:

- https://github.com/maxlinde/baymedr
- Report bugs at https://github.com/maxlinde/baymedr/issues

equiv_bf *Bayes factor for equivalence designs*

Description

equiv_bf computes a Bayes factor for equivalence designs.

Usage

```
equiv_bf(x = NULL, y = NULL, n_x = NULL, n_y = NULL,
  mean_x = NULL, mean_y = NULL, sd_x = NULL, sd_y = NULL,
  ci_margin = NULL, ci_level = NULL, interval = 0,
  interval_std = TRUE, prior_scale = 1/sqrt(2))
```

Arguments

x A numeric vector of observations for the control group.

y A numeric vector of observations for the experimental group.

n_x A numeric vector of length one, specifying the sample size of the control group.

n_y A numeric vector of length one, specifying the sample size of the experimental group.

mean_x A numeric vector of length one, specifying the mean of the dependent variable in the control group.

mean_y A numeric vector of length one, specifying the mean of the dependent variable in the experimental group.

sd_x A numeric vector of length one, specifying the standard deviation of the dependent variable in the control group. Only sd_x and sd_y **OR** ci_margin and ci_level should be defined (see Details).

sd_y A numeric vector of length one, specifying the standard deviation of the dependent variable in the experimental group. Only sd_x and sd_y **OR** ci_margin and ci_level should be defined (see Details).

ci_margin A numeric vector of length one, specifying the margin of the confidence interval (i.e., the width of the confidence interval divided by 2) of the mean difference on the dependent variable between the experimental and control groups. Only sd_x and sd_y **OR** ci_margin and ci_level should be defined (see Details).

ci_level A numeric vector of length one, specifying the confidence level of ci_margin. The value must be between 0 and 1 (e.g., 0.95 for a 95% confidence interval). Only sd_x and sd_y **OR** ci_margin and ci_level should be defined (see Details).

`interval` A numeric vector of length one or two, specifying the boundaries of the equivalence interval. If a numeric vector of length one is specified, a symmetric equivalence interval will be used (e.g., a 0.1 is equivalent to c(-0.1, 0.1)). A numeric vector of length two provides the possibility to specify an asymmetric equivalence interval (e.g., c(-0.1, 0.2)). The default is 0, indicating a point null hypothesis rather than an interval (see Details).

`interval_std` A logical vector of length one, specifying whether the equivalence interval (i.e., `interval`) is given in standardised (TRUE; the default) or unstandardised (FALSE) units.

`prior_scale` A numeric vector of length one, specifying the scale of the Cauchy prior distribution for the effect size under the alternative hypothesis (see Details). The default value is r = 1 / sqrt(2).

Details

The Bayes factor resulting from `equiv_bf` tests the null hypothesis that the experimental group (e.g., a new medication) and the control group (e.g., a placebo or an already existing medication) are equivalent. The alternative hypothesis is that the two groups are not equivalent.

Since the main goal of `equiv_bf` is to establish equivalence, the resulting Bayes factor quantifies evidence in favour of the null hypothesis (i.e., BF01). However, evidence for the alternative hypothesis can easily be calculated by taking the reciprocal of the original Bayes factor (i.e., BF10 = 1 / BF01). Quantification of evidence in favour of the null hypothesis is logically sound and legitimate within the Bayesian framework but not in the traditional frequentist framework (see e.g., van Ravenzwaaij et al., 2019).

Importantly, `equiv_bf` can be utilized to calculate a Bayes factor based on raw data (i.e., if arguments `x` and `y` are defined) or summary statistics (i.e., if arguments `n_x`, `n_y`, `mean_x`, and `mean_y` are defined). In the latter case, the user has the freedom to supply values either for the arguments `sd_x` and `sd_y` **OR** `ci_margin` and `ci_level`. The choice should depend on the information that is available to the user. Arguments with 'x' as a name or suffix correspond to the control group, whereas arguments with 'y' as a name or suffix correspond to the experimental group.

Using the argument `interval`, the user can specify the equivalence interval. In contrast to null hypothesis significance testing (NHST), `equiv_bf` has the advantage that it is not compulsory to specify an equivalence interval (see van Ravenzwaaij et al., 2019). Therefore, the default value of the argument `interval` is 0, indicating a point null hypothesis. However, if the user prefers to have an equivalence interval, the argument `interval` can be set in two ways: If a *symmetric* interval is desired, the user can either specify a numeric vector of length one (e.g., 0.1, which is converted to c(-0.1, 0.1)) or a numeric vector of length two (e.g., c(-0.1, 0.1)); if an *asymmetric* interval is desired, the user can specify a numeric vector of length two (e.g., c(-0.1, 0.2)). It can be specified whether the equivalence interval (i.e., `interval`) is given in standardised or unstandardised units with the `interval_std` argument, where TRUE, corresponding to standardised units, is the default.

For the calculation of the Bayes factor, we chose a Cauchy prior density for the effect size under the alternative hypothesis. The shape of the Cauchy distribution can be manipulated with its location and scale parameters. The standard Cauchy distribution, with a location parameter of 0 and a scale parameter of 1, resembles a standard Normal distribution, except that the Cauchy distribution has less mass at the centre but heavier tails (see, e.g., Rouder et al., 2009, for a visualisation). Mathematically, the standard Cauchy distribution is equivalent to a Normal distribution with a mean of 0 and a variance that follows and inverse chi-square distribution with one degree of freedom, for which the variance is integrated out (Liang et al., 2008). The argument `prior_scale` specifies the width of the Cauchy prior, which corresponds to half of the interquartile range. Thus, by adjusting the Cauchy prior scale with `prior_scale`, we can emphasise different ranges of effect sizes that might be expected. The default prior scale is set to r = 1 / sqrt(2).

`equiv_bf` creates an S4 object of class baymedrEquivalence, which has multiple slots/entries (e.g., type of data, prior scale, Bayes factor, etc.; see Value). If it is desired to store or extract solely the Bayes factor, the user can do this with `get_bf`, by setting the S4 object as an argument (see Examples).

Value

An S4 object of class baymedrEquivalence is returned. Contained are a description of the model and the resulting Bayes factor:

- test: The type of analysis
- hypotheses: A statement of the hypotheses
  - h0: The null hypothesis
  - h1: The alternative hypothesis
- interval: Specification of the equivalence interval in standardised and unstandardised units
  - lower_std: The standardised lower boundary of the equivalence interval
  - upper_std: The standardised upper boundary of the equivalence interval
  - lower_unstd: The unstandardised lower boundary of the equivalence interval
  - upper_unstd: The unstandardised upper boundary of the equivalence interval
- data: A description of the data
  - type: The type of data ('raw' when arguments `x` and `y` are used or 'summary' when arguments `n_x`, `n_y`, `mean_x`, `mean_y`, `sd_x`, and `sd_y` (or `ci_margin` and `ci_level` instead of `sd_x` and `sd_y`) are used)
  - ...: values for the arguments used, depending on 'raw' or summary'
- prior_scale: The width of the Cauchy prior distribution
- bf: The resulting Bayes factor

A summary of the model is shown by printing the object.

References

Gronau, Q. F., Ly, A., & Wagenmakers, E.-J. (2019). Informed Bayesian t-tests. *The American Statistician*.

Liang, F., Paulo, R., Molina, G., Clyde, M. A., & Berger, J. O. (2008). Mixtures of g priors for Bayesian variable selection. *Journal of the American Statistical Association*, *103*(481), 410-423.

Rouder, J. N., Speckman, P. L., Sun, D., Morey, R. D., & Iverson, G. (2009). Bayesian t tests for accepting and rejecting the null hypothesis. *Psychonomic Bulletin & Review*, *16*(2), 225-237.

van Ravenzwaaij, D., Monden, R., Tendeiro, J. N., & Ioannidis, J. P. A. (2019). Bayes factors for superiority, non-inferiority, and equivalence designs. *BMC Medical Research Methodology*, *19*(1), 71.

Examples

```
## equiv_bf using raw data:

# Assign model to variable.
equiv_raw <- equiv_bf(x = rnorm(100, 10, 15),
                      y = rnorm(130, 13, 10))

# Extract Bayes factor from variable.
get_bf(equiv_raw)

# ----------
# ----------

## equiv_bf using summary statistics with data from Steiner et al. (2015).
## With a point null hypothesis:

# Assign model to variable.
equiv_sum_point <- equiv_bf(n_x = 560,
                            n_y = 538,
                            mean_x = 8.683,
                            mean_y = 8.516,
                            sd_x = 3.6,
                            sd_y = 3.6)

# Extract Bayes factor from model.
get_bf(equiv_sum_point)

# ----------
# ----------

## equiv_bf using summary statistics with data from Steiner et al. (2015).
## With an interval null hypothesis:

# Assign model to variable.
equiv_sum_interval <- equiv_bf(n_x = 560,
                               n_y = 538,
                               mean_x = 8.683,
                               mean_y = 8.516,
                               sd_x = 3.6,
                               sd_y = 3.6,
                               interval = 0.05)

# Extract Bayes factor from model.
get_bf(equiv_sum_interval)
```

get_bf *Extract Bayes factor from S4 object*

Description

get_bf extracts the Bayes factor from an S4 object (i.e., baymedrSuperiority, baymedrEquivalence, baymedrNonInferiority), created from the functions super_bf, equiv_bf, or infer_bf.

Usage

```
get_bf(object)
```

Arguments

object	An S4 object of class baymedrSuperiority, baymedrEquivalence, or baymedrNonInferiority.

Value

A numeric scalar, providing the Bayes factor from an S4 object.

Examples

```
# Extract Bayes factor from a baymedrSuperiority object using raw data:
mod_super <- super_bf(x = rnorm(100, 10, 15),
                      y = rnorm(130, 13, 10))

get_bf(object = mod_super)

# Extract Bayes factor from a baymedrEquivalence object using raw data:
mod_equiv <- equiv_bf(x = rnorm(100, 10, 15),
                      y = rnorm(130, 13, 10))

get_bf(object = mod_equiv)

# Extract Bayes factor from a baymedrNonInferiority object using raw data:
mod_infer <- infer_bf(x = rnorm(100, 10, 15),
                      y = rnorm(130, 13, 10),
                      ni_margin = 1)

get_bf(object = mod_infer)
```

infer_bf *Bayes factor for non-inferiority designs*

Description

infer_bf computes a Bayes factor for non-inferiority designs.

Usage

```
infer_bf(x = NULL, y = NULL, n_x = NULL, n_y = NULL,
  mean_x = NULL, mean_y = NULL, sd_x = NULL, sd_y = NULL,
  ci_margin = NULL, ci_level = NULL, ni_margin = NULL,
  ni_margin_std = TRUE, prior_scale = 1/sqrt(2), direction = "high")
```

Arguments

`x` A numeric vector of observations for the control group.

`y` A numeric vector of observations for the experimental group.

`n_x` A numeric vector of length one, specifying the sample size of the control group.

`n_y` A numeric vector of length one, specifying the sample size of the experimental group.

`mean_x` A numeric vector of length one, specifying the mean of the dependent variable in the control group.

`mean_y` A numeric vector of length one, specifying the mean of the dependent variable in the experimental group.

`sd_x` A numeric vector of length one, specifying the standard deviation of the dependent variable in the control group. Only `sd_x` and `sd_y` **OR** `ci_margin` and `ci_level` should be defined (see Details).

`sd_y` A numeric vector of length one, specifying the standard deviation of the dependent variable in the experimental group. Only `sd_x` and `sd_y` **OR** `ci_margin` and `ci_level` should be defined (see Details).

`ci_margin` A numeric vector of length one, specifying the margin of the confidence interval (i.e., the width of the confidence interval divided by 2) of the mean difference on the dependent variable between the experimental and control groups. Only `sd_x` and `sd_y` **OR** `ci_margin` and `ci_level` should be defined (see Details).

`ci_level` A numeric vector of length one, specifying the confidence level of `ci_margin`. The value must be between 0 and 1 (e.g., 0.95 for a 95% confidence interval). Only `sd_x` and `sd_y` **OR** `ci_margin` and `ci_level` should be defined (see Details).

`ni_margin` A numeric vector of length one, specifying the non-inferiority margin. The value should be a positive number.

`ni_margin_std` A logical vector of length one, specifying whether the non-inferiority margin (i.e., `ni_margin`) is given in standardised (TRUE; the default) or unstandardised (FALSE) units.

`prior_scale` A numeric vector of length one, specifying the scale of the Cauchy prior distribution for the effect size under the alternative hypothesis (see Details). The default value is r = 1 / sqrt(2).

`direction` A character vector of length one, specifying the direction of non-inferior scores. 'low' indicates that low scores on the measure of interest correspond to a non-inferior outcome and 'high' (the default) indicates that high scores on the measure of interest correspond to a non-inferior outcome (see Details).

Details

The formulation of the null and alternative hypotheses differ depending on whether high or low scores on the measure of interest represent non-inferiority. In the case where high scores correspond to non-inferiority (e.g., amount of social interaction), the Bayes factor resulting from `infer_bf` tests the null hypothesis that the experimental group (e.g., a new medication) is lower than the control group (e.g., a placebo or an already existing medication) minus a constant value, which is given by the non-inferiority margin. The alternative hypothesis is that the experimental group is higher than the control group minus the non-inferiority margin. In turn, when low values on the measure of interest correspond to non-inferiority (e.g., severity of symptoms), the Bayes factor resulting from `infer_bf` tests the null hypothesis that the experimental group is higher than the control group plus the non-inferiority margin. The alternative hypothesis states that the experimental group is lower than the control group plus the non-inferiority margin.

Since the main goal of `infer_bf` is to establish non-inferiority, the resulting Bayes factor quantifies evidence in favour of the alternative hypothesis (i.e., BF10). However, evidence for the null hypothesis can easily be calculated by taking the reciprocal of the original Bayes factor (i.e., BF01 = 1 / BF10). Quantification of evidence in favour of the null hypothesis is logically sound and legitimate within the Bayesian framework but not in the traditional frequentist framework (see e.g., van Ravenzwaaij et al., 2019).

Importantly, `infer_bf` can be utilized to calculate a Bayes factor based on raw data (i.e., if arguments `x` and `y` are defined) or summary statistics (i.e., if arguments `n_x`, `n_y`, `mean_x`, and `mean_y` (or `ci_margin` and `ci_level` instead of `sd_x` and `sd_y`) are defined). Arguments with 'x' as a name or suffix correspond to the control group, whereas arguments with 'y' as a name or suffix correspond to the experimental group (i.e., the group for which we seek to establish non-inferiority).

Since sometimes high scores on the dependent variable are considered non-inferior (e.g., amount of social interactions) and sometimes rather the low scores (e.g., severity of symptoms), the user can specify the direction of non-inferiority with the argument `direction`. For the case where higher values on the dependent variable indicate non-inferiority, the user should specify 'high' (the default) for the argument `direction`; if lower values on the dependent variable indicate non-inferiority, 'low' should be specified for the argument `direction`.

With the argument `ni_margin`, the user can determine the non-inferiority margin. `ni_margin` should be a positive number. Following ethical scientific rigour, `ni_margin` should be defined prior to data collection and data analysis. The user can declare whether the non-inferiority margin was specified in standardised or unstandardised units with the `ni_margin_std` argument, where TRUE, corresponding to standardised units, is the default.

For the calculation of the Bayes factor, we chose a Cauchy prior density for the effect size under the alternative hypothesis. The shape of the Cauchy distribution can be manipulated with its location and scale parameters. The standard Cauchy distribution, with a location parameter of 0 and a scale parameter of 1, resembles a standard Normal distribution, except that the Cauchy distribution has less mass at the centre but heavier tails (see, e.g., Rouder et al., 2009, for a visualisation). Mathematically, the standard Cauchy distribution is equivalent to a Normal distribution with a mean of 0 and a variance that follows and inverse chi-square distribution with one degree of freedom, for which the variance is integrated out (Liang et al., 2008). The argument `prior_scale` specifies the width of the Cauchy prior, which corresponds to half of the interquartile range. Thus, by adjusting the Cauchy prior scale with `prior_scale`, we can emphasise different ranges of effect sizes that might be expected. The default prior scale is set to 1 / sqrt(2).

`infer_bf` creates an S4 object of class baymedrNonInferiority, which has multiple slots/entries (e.g., type of data, prior scale, Bayes factor, etc.; see Value). If it is desired to store or extract solely the Bayes factor, the user can do this with `get_bf`, by setting the S4 object as an argument (see Examples).

Value

An S4 object of class baymedrNonInferiority is returned. Contained are a description of the model and the resulting Bayes factor:

- test: The type of analysis
- hypotheses: A statement of the hypotheses
  - h0: The null hypothesis
  - h1: The alternative hypothesis
- ni_margin: The value for ni_margin in standardised and unstandardised units
  - ni_mar_std: The standardised non-inferiority margin
  - ni_mar_unstd: The unstandardised non-inferiority margin
- data: A description of the data
  - type: The type of data ('raw' when arguments `x` and `y` are used or 'summary' when arguments `n_x`, `n_y`, `mean_x`, `mean_y`, `sd_x`, and `sd_y` (or `ci_margin` and `ci_level` instead of `sd_x` and `sd_y`) are used)
  - ...: values for the arguments used, depending on 'raw' or summary'
- prior_scale: The width of the Cauchy prior distribution
- bf: The resulting Bayes factor

A summary of the model is shown by printing the object.

References

Gronau, Q. F., Ly, A., & Wagenmakers, E.-J. (2019). Informed Bayesian t-tests. *The American Statistician*.

Liang, F., Paulo, R., Molina, G., Clyde, M. A., & Berger, J. O. (2008). Mixtures of g priors for Bayesian variable selection. *Journal of the American Statistical Association*, *103*(481), 410-423.

Rouder, J. N., Speckman, P. L., Sun, D., Morey, R. D., & Iverson, G. (2009). Bayesian t tests for accepting and rejecting the null hypothesis. *Psychonomic Bulletin & Review*, *16*(2), 225-237.

van Ravenzwaaij, D., Monden, R., Tendeiro, J. N., & Ioannidis, J. P. A. (2019). Bayes factors for superiority, non-inferiority, and equivalence designs. *BMC Medical Research Methodology*, *19*(1), 71.

Examples

```
## infer_bf using raw data:

# Assign model to variable.
infer_raw <- infer_bf(x = rnorm(100, 10, 15),
                      y = rnorm(130, 13, 10),
                      ni_margin = 1.5,
                      ni_margin_std = FALSE)

# Extract Bayes factor from model.
get_bf(infer_raw)

# ----------
# ----------

## infer_bf using summary statistics with data from Andersson et al. (2013).
## Test at timepoint 1:

# Assign model to variable.
infer_sum_t1 <- infer_bf(n_x = 33,
                         n_y = 32,
                         mean_x = 17.1,
                         mean_y = 13.6,
                         sd_x = 8,
                         sd_y = 9.8,
                         ni_margin = 2,
                         ni_margin_std = FALSE,
                         direction = ”low”)

# Extract Bayes factor from model
get_bf(infer_sum_t1)

# ----------
# ----------

## infer_bf using summary statistics with data from Andersson et al. (2013).
## Test at timepoint 2:

# Assign model to variable.
infer_sum_t2 <- infer_bf(n_x = 30,
                         n_y = 32,
                         mean_x = 13.5,
                         mean_y = 9.2,
                         sd_x = 8.7,
                         sd_y = 7.6,
                         ni_margin = 2,
                         ni_margin_std = FALSE,
                         direction = ”low”)

# Extract Bayes factor from model
get_bf(infer_sum_t2)
```

model-classes *S4 classes to represent different models*

Description

The S4 classes baymedrSuperiority, baymedrEquivalence, and baymedrNonInferiority represent models for the superiority (`super_bf`), equivalence (`equiv_bf`), and non-inferiority (`infer_bf`) tests, respectively.

Slots

`test` Type of test that was conducted.

`hypotheses` The hypotheses that are tested.

`data` The type of data that was used.

`prior_scale` The Cauchy prior scale that was used.

`bf` The resulting Bayes factor.

`interval` The equivalence interval in case of `equiv_bf`.

`ni_margin` The non-inferiority margin in case of `infer_bf`.

`super_bf` *Bayes factor for superiority designs*

Description

`super_bf` computes a Bayes factor for superiority designs.

Usage

```
super_bf(x = NULL, y = NULL, n_x = NULL, n_y = NULL,
  mean_x = NULL, mean_y = NULL, sd_x = NULL, sd_y = NULL,
  ci_margin = NULL, ci_level = NULL, prior_scale = 1/sqrt(2),
  direction = "high", alternative = "one.sided")
```

Arguments

`x` A numeric vector of observations for the control group.

`y` A numeric vector of observations for the experimental group.

`n_x` A numeric vector of length one, specifying the sample size of the control group.

`n_y` A numeric vector of length one, specifying the sample size of the experimental group.

`mean_x` A numeric vector of length one, specifying the mean of the dependent variable in the control group.

`mean_y` A numeric vector of length one, specifying the mean of the dependent variable in the experimental group.

`sd_x` A numeric vector of length one, specifying the standard deviation of the dependent variable in the control group. Only `sd_x` and `sd_y` **OR** `ci_margin` and `ci_level` should be defined (see Details).

`sd_y` A numeric vector of length one, specifying the standard deviation of the dependent variable in the experimental group. Only `sd_x` and `sd_y` **OR** `ci_margin` and `ci_level` should be defined (see Details).

`ci_margin` A numeric vector of length one, specifying the margin of the confidence interval (i.e., the width of the confidence interval divided by 2) of the mean difference on the dependent variable between the experimental and control groups. Only `sd_x` and `sd_y` **OR** `ci_margin` and `ci_level` should be defined (see Details).

`ci_level` A numeric vector of length one, specifying the confidence level of `ci_margin`. The value must be between 0 and 1 (e.g., 0.95 for a 95% confidence interval). Only `sd_x` and `sd_y` **OR** `ci_margin` and `ci_level` should be defined (see Details).

`prior_scale` A numeric vector of length one, specifying the scale of the Cauchy prior distribution for the effect size under the alternative hypothesis (see Details). The default value is r = 1 / sqrt(2).

`direction` A character vector of length one, specifying the direction of superior scores. 'low' indicates that low scores on the measure of interest correspond to a superior outcome and 'high' (the default) indicates that high scores on the measure of interest correspond to a superior outcome (see Details).

`alternative` A character vector of length one, specifying whether a one-sided or two-sided test shall be conducted. 'one.sided' (the default) corresponds to a one-sided test and 'two.sided' corresponds to a two-sided test (see Details).

Details

The Bayes factor resulting from `super_bf` tests the null hypothesis that the experimental group (e.g., a new medication) is equal to the control group (e.g., a placebo or an already existing medication). The formulation of the alternative hypothesis depends on whether a one- or two-sided test shall be conducted and on whether the low or high values correspond to superiority. As the name 'superiority' implies, the test is typically understood as a one-tailed test. In practice, however, a two-tailed test is often employed, followed by subsequent confirmation that the results follow the expected direction. To cover both research practices, the user has the possibility to specify which of these two alternatives should be employed through the argument `alternative`. Furthermore, since sometimes high scores on the dependent variable are considered superior (e.g., amount of social interactions) and sometimes rather the low scores (e.g., severity of symptoms), the user can specify the direction of superiority with the argument `direction`. For the case where higher values on the dependent variable indicate superiority, the user should specify 'high' (the default) for the argument `direction`; if lower values on the dependent variable indicate superiority, 'low' should be specified for the argument `direction`. Therefore, in case a two-sided test is employed, the alternative hypothesis states that the experimental group is not equal to the control group. In the case of a one-sided test and low scores corresponding to superiority, the alternative hypothesis is that the experimental group is lower than the control group. Lastly, in the case where a one-sided test is employed and high scores indicate superiority, the alternative hypothesis states that the experimental group is higher than the control group.

Since the main goal of `super_bf` is to establish superiority, the resulting Bayes factor quantifies evidence in favour of the alternative hypothesis (i.e., BF10). However, evidence for the null hypothesis can easily be calculated by taking the reciprocal of the original Bayes factor (i.e., BF01 = 1 / BF10). Quantification of evidence in favour of the null hypothesis is logically sound and legitimate within the Bayesian framework but not in the traditional frequentist framework (see e.g., van Ravenzwaaij et al., 2019).

Importantly, `super_bf` can be utilized to calculate a Bayes factor based on raw data (i.e., if arguments `x` and `y` are defined) or summary statistics (i.e., if arguments `n_x`, `n_y`, `mean_x`, and `mean_y` are defined). In the latter case, the user has the freedom to supply values either for the arguments `sd_x` and `sd_y` **OR** `ci_margin` and `ci_level`. The choice should depend on the information that is available to the user. Arguments with 'x' as a name or suffix correspond to the control group, whereas arguments with 'y' as a name or suffix correspond to the experimental group (i.e., the group for which we seek to establish superiority).

For the calculation of the Bayes factor, we chose a Cauchy prior density for the effect size under the alternative hypothesis. The shape of the Cauchy distribution can be manipulated with its location and scale parameters. The standard Cauchy distribution, with a location parameter of 0 and a scale parameter of 1, resembles a standard Normal distribution, except that the Cauchy distribution has less mass at the centre but heavier tails (see, e.g., Rouder et al., 2009, for a visualisation). Mathematically, the standard Cauchy distribution is equivalent to a Normal distribution with a mean of 0 and a variance that follows and inverse chi-square distribution with one degree of freedom, for which the variance is integrated out (Liang et al., 2008). The argument `prior_scale` specifies the width of the Cauchy prior, which corresponds to half of the interquartile range. Thus, by adjusting the Cauchy prior scale with `prior_scale`, we can emphasise different ranges of effect sizes that might be expected. The default prior scale is set to 1 / sqrt(2).

`super_bf` creates an S4 object of class baymedrSuperiority, which has multiple slots/entries (e.g., type of data, prior scale, Bayes factor, etc.; see Value). If it is desired to store or extract solely the Bayes factor, the user can do this with `get_bf`, by setting the S4 object as an argument (see Examples).

Value

An S4 object of class baymedrSuperiority is returned. Contained are a description of the model and the resulting Bayes factor:

- test: The type of analysis
- hypotheses: A statement of the hypotheses
  - h0: The null hypothesis
  - h1: The alternative hypothesis
- data: A description of the data
  - type: The type of data ('raw' when arguments `x` and `y` are used or 'summary' when arguments `n_x`, `n_y`, `mean_x`, `mean_y`, `sd_x`, and `sd_y` (or `ci_margin` and `ci_level` instead of `sd_x` and `sd_y`) are used)
  - ...: values for the arguments used, depending on 'raw' or 'summary'
- prior_scale: The scale of the Cauchy prior distribution
- bf: The resulting Bayes factor

A summary of the model is shown by printing the object.

References

Gronau, Q. F., Ly, A., & Wagenmakers, E.-J. (2019). Informed Bayesian t-tests. *The American Statistician*.

Liang, F., Paulo, R., Molina, G., Clyde, M. A., & Berger, J. O. (2008). Mixtures of g priors for Bayesian variable selection. *Journal of the American Statistical Association*, *103*(481), 410-423.

Rouder, J. N., Speckman, P. L., Sun, D., Morey, R. D., & Iverson, G. (2009). Bayesian t tests for accepting and rejecting the null hypothesis. *Psychonomic Bulletin & Review*, *16*(2), 225-237.

van Ravenzwaaij, D., Monden, R., Tendeiro, J. N., & Ioannidis, J. P. A. (2019). Bayes factors for superiority, non-inferiority, and equivalence designs. *BMC Medical Research Methodology*, *19*(1), 71.

Examples

```
## super_bf using raw data:

# Assign model to variable.
super_raw <- super_bf(x = rnorm(100, 10, 15),
                      y = rnorm(130, 13, 10))

# Extract Bayes factor from model.
get_bf(super_raw)

# ----------
# ----------

## super_bf using summary statistics with data from Skjerven et al. (2013).
## EXAMPLE 1
## The one-sided alternative:

# Assign model to variable.
super_sum_one_ex1 <- super_bf(n_x = 201,
                              n_y = 203,
                              mean_x = 68.1,
                              mean_y = 63.6,
                              ci_margin = (15.5 - (-6.5)) / 2,
                              ci_level = 0.95,
                              direction = "low",
                              alternative = "one.sided")

# Extract Bayes factor from model.
get_bf(super_sum_one_ex1)

# ----------

## super_bf using summary statistics with data from Skjerven et al. (2013).
## EXAMPLE 1
## The two-sided alternative:

# Assign model to variable.
super_sum_two_ex1 <- super_bf(n_x = 201,
                              n_y = 203,
                              mean_x = 68.1,
                              mean_y = 63.6,
                              ci_margin = (15.5 - (-6.5)) / 2,
                              ci_level = 0.95,
                              alternative = "two.sided")

# Extract Bayes factor from model.
get_bf(super_sum_two_ex1)

# ----------
# ----------

## super_bf using summary statistics with data from Skjerven et al. (2013).
## EXAMPLE 2
## The one-sided alternative:

# Assign model to variable.
super_sum_one_ex2 <- super_bf(n_x = 200,
                              n_y = 204,
                              mean_x = 47.6,
                              mean_y = 61.3,
                              ci_margin = (24.4 - 2.9) / 2,
                              ci_level = 0.95,
                              direction = "low",
                              alternative = "one.sided")

# Extract Bayes factor from model.
get_bf(super_sum_one_ex2)

# ----------

## super_bf using summary statistics with data from Skjerven et al. (2013).
## EXAMPLE 2
## The two-sided alternative:

# Assign model to variable.
super_sum_two_ex2 <- super_bf(n_x = 200,
                              n_y = 204,
                              mean_x = 47.6,
                              mean_y = 61.3,
                              ci_margin = (24.4 - 2.9) / 2,
                              ci_level = 0.95,
                              alternative = "two.sided")

# Extract Bayes factor from model.
get_bf(super_sum_two_ex2)
```

Index