

# Package ‘bain’

October 12, 2022

**Type** Package

**Date** 2021-12-06

**Title** Bayes Factors for Informative Hypotheses

**Version** 0.2.8

**Description** Computes approximated adjusted fractional Bayes factors for equality, inequality, and about equality constrained hypotheses. For a tutorial on this method, see Hoijtink, Mulder, van Lissa, & Gu, (2019) <[doi:10.31234/osf.io/v3shc](https://doi.org/10.31234/osf.io/v3shc)>. For applications in structural equation modeling, see: Van Lissa, Gu, Mulder, Rosseel, Van Zundert, & Hoijtink, (2021) <[doi:10.1080/10705511.2020.1745644](https://doi.org/10.1080/10705511.2020.1745644)>. For the statistical underpinnings, see Gu, Mulder, and Hoijtink (2018) <[doi:10.1111/bmsp.12110](https://doi.org/10.1111/bmsp.12110)>; Hoijtink, Gu, & Mulder, J. (2019) <[doi:10.1111/bmsp.12145](https://doi.org/10.1111/bmsp.12145)>; Hoijtink, Gu, Mulder, & Rosseel, (2019) <[doi:10.31234/osf.io/q6h5w](https://doi.org/10.31234/osf.io/q6h5w)>.

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**URL** <https://informative-hypotheses.sites.uu.nl/software/bain/>

**BugReports** <https://github.com/cjvanlissa/bain/>

**NeedsCompilation** yes

**RoxygenNote** 7.1.2

**Depends** R (>= 3.0.0), stats

**Imports** lavaan

**Suggests** MASS, testthat, knitr, rmarkdown

**VignetteBuilder** knitr

**Author** Xin Gu [aut],  
Herbert Hoijtink [aut],  
Joris Mulder [aut],  
Caspar J van Lissa [aut, cre],  
Van Zundert Camiel [ctb],  
Jeff Jones [ctb],  
Niels Waller [ctb]

**Maintainer** Caspar J van Lissa <c.j.vanlissa@uu.nl>

**Repository** CRAN

**Date/Publication** 2021-12-06 13:20:02 UTC

## R topics documented:

bain . . . . .	2
bain_sensitivity . . . . .	4
seBeta . . . . .	5
sesamesim . . . . .	7
t_test . . . . .	8

<b>Index</b>	<b>10</b>
--------------	-----------

---

bain	<i>Bayes factors for informative hypotheses</i>
------	---

---

## Description

bain is an acronym for "Bayesian informative hypotheses evaluation". It uses the Bayes factor to evaluate hypotheses specified using equality and inequality constraints among (linear combinations of) parameters in a wide range of statistical models. A **tutorial** by Hoijtink, Mulder, van Lissa, and Gu (2018), was published in Psychological Methods. The preprint of that tutorial is available on PsyArxiv (doi: [10.31234/osf.io/v3shc](https://doi.org/10.31234/osf.io/v3shc)) or on the bain website at <https://informative-hypotheses.sites.uu.nl/software/bain/> **Users are advised to read the tutorial AND the vignette that is provided with this package before using bain.**

## Usage

```
bain(x, hypothesis, fraction = 1, ...)
```

## Arguments

x	An R object containing the outcome of a statistical analysis. Currently, the following objects can be processed: <code>lm()</code> , <code>t_test()</code> , lavaan objects created with the <code>sem()</code> , <code>cfa()</code> , and <code>growth()</code> functions, and named vector objects. See the vignette for elaborations.
hypothesis	A character string containing the informative hypotheses to evaluate. See the vignette for elaborations.
fraction	A number representing the fraction of information in the data used to construct the prior distribution. The default value 1 denotes the minimal fraction, 2 denotes twice the minimal fraction, etc. See the vignette for elaborations.
...	Additional arguments. See the vignette for elaborations.

## Value

The main output resulting from analyses with `bain` are Bayes factors and posterior model probabilities associated with the hypotheses that are evaluated. See the **tutorial** and the **vignette** for further elaborations.

## Author(s)

The main authors of the `bain` package are Xin Gu, Caspar van Lissa, Herbert Hoijtink and Joris Mulder with smaller contributions by Marlyne Bosman, Camiel van Zundert, and Fayette Klaassen. Contact information can be found on the `bain` website at <https://informative-hypotheses.sites.uu.nl/software/bain/>

## References

For a tutorial on this method, see:

Hoijtink, H., Mulder, J., van Lissa, C., & Gu, X. (2019). A tutorial on testing hypotheses using the Bayes factor. *Psychological methods*, 24(5), 539. doi: [10.31234/osf.io/v3shc](https://doi.org/10.31234/osf.io/v3shc)

For applications in structural equation modeling, see:

Van Lissa, C. J., Gu, X., Mulder, J., Rosseel, Y., Van Zundert, C., & Hoijtink, H. (2021). Teacher's corner: Evaluating informative hypotheses using the Bayes factor in structural equation models. *Structural Equation Modeling: A Multidisciplinary Journal*, 28(2), 292-301. doi: [10.1080/10705511.2020.1745644](https://doi.org/10.1080/10705511.2020.1745644).

For the statistical underpinnings, see:

Gu, Mulder, and Hoijtink (2018). Approximated adjusted fractional Bayes factors: A general method for testing informative hypotheses. *British Journal of Mathematical and Statistical Psychology*, 71(2), 229-261. doi: [10.1111/bmsp.12110](https://doi.org/10.1111/bmsp.12110).

Hoijtink, H., Gu, X., & Mulder, J. (2019). Bayesian evaluation of informative hypotheses for multiple populations. *British Journal of Mathematical and Statistical Psychology*, 72(2), 219-243. doi: [10.1111/bmsp.12145](https://doi.org/10.1111/bmsp.12145).

Hoijtink, H., Gu, X., Mulder, J., & Rosseel, Y. (2019). Computing Bayes factors from data with missing values. *Psychological Methods*, 24(2), 253. doi: [10.31234/osf.io/q6h5w](https://doi.org/10.31234/osf.io/q6h5w)

## Examples

```
# Evaluation of informative hypotheses for an ANOVA
# make a factor of variable site
sesamesim$site <- as.factor(sesamesim$site)
# execute an analysis of variance using lm() which, due to the -1, returns
# estimates of the means of postnumb per group
anov <- lm(postnumb~site-1,sesamesim)
# take a look at the estimated means and their names
coef(anov)
# set a seed value
set.seed(100)
# use the names to formulate and test hypotheses with bain
results <- bain(anov, "site1=site2=site3=site4=site5; site2>site5>site1>
site3>site4")
```

```
#
# SEE THE TUTORIAL AND VIGNETTE FOR MANY ADDITIONAL EXAMPLES
```

---

bain_sensitivity	<i>Sensitivity analysis for bain</i>
------------------	--------------------------------------

---

### Description

Conducts a sensitivity analysis for [bain](#).

### Usage

```
bain_sensitivity(x, hypothesis, fractions = 1, ...)
```

### Arguments

x	An R object containing the outcome of a statistical analysis. Currently, the following objects can be processed: <code>lm()</code> , <code>t_test()</code> , lavaan objects created with the <code>sem()</code> , <code>cfa()</code> , and <code>growth()</code> functions, and named vector objects. See the vignette for elaborations.
hypothesis	A character string containing the informative hypotheses to evaluate. See the vignette for elaborations.
fractions	A number representing the fraction of information in the data used to construct the prior distribution. The default value 1 denotes the minimal fraction, 2 denotes twice the minimal fraction, etc. See the vignette for elaborations.
...	Additional arguments passed to <a href="#">bain</a> .

### Details

The Bayes factor for equality constraints is sensitive to a scaling factor applied to the prior distribution. The argument `fraction` adjusts this scaling factor. The function `bain_sensitivity` is a wrapper for [bain](#), which accepts a vector for the `fractions` argument, and returns a list of bain results objects. A table with a sensitivity analysis for specific statistics can be obtained using the `summary()` function, which accepts the argument `summary(which_stat = ...)`. The available statistics are elements of the `$fit` table (`Fit_eq`, `Com_eq`, `Fit_in`, `Com_in`, `Fit`, `Com`, `BF`, `PMPa`, and `PMPb`), and elements of the `BFmatrix`, which can be accessed by matrix notation, e.g.: `summary(bain_sens, which_stat = "BFmatrix[1,2]")`.

### Value

A data.frame of class "bain\_sensitivity".

**Examples**

```

sesamesim$site <- as.factor(sesamesim$site)
res <- lm(sesamesim$postnumb~sesamesim$site-1)
set.seed(4583)
bain_sens <- bain_sensitivity(res, "site1=site2;
                                site2>site5",
                                fractions = c(1,2,3))
summary(bain_sens, which_stat = "BF.c")
summary(bain_sens, which_stat = "BFmatrix[1,2]")

```

seBeta

*Standard Errors and CIs for Standardized Regression Coefficients***Description**

Computes Normal Theory and ADF Standard Errors and CIs for Standardized Regression Coefficients

**Usage**

```

seBeta(
  X = NULL,
  y = NULL,
  cov.x = NULL,
  cov.xy = NULL,
  var.y = NULL,
  Nobs = NULL,
  alpha = 0.05,
  estimator = "ADF"
)

```

**Arguments**

X	Matrix of predictor scores.
y	Vector of criterion scores.
cov.x	Covariance or correlation matrix of predictors.
cov.xy	Vector of covariances or correlations between predictors and criterion.
var.y	Criterion variance.
Nobs	Number of observations.
alpha	Desired Type I error rate; default = .05.
estimator	'ADF' or 'Normal' confidence intervals - requires raw X and raw y; default = 'ADF'.

**Value**

cov.Beta	Normal theory or ADF covariance matrix of standardized regression coefficients.
se.Beta	standard errors for standardized regression coefficients.
alpha	desired Type-I error rate.
CI.Beta	Normal theory or ADF (1-alpha) intervals for standardized regression coefficients.
estimator	estimator = "ADF" or "Normal".

**Author(s)**

Jeff Jones and Niels Waller

**References**

Jones, J. A, and Waller, N. G. (2015). The Normal-Theory and Asymptotic Distribution-Free (ADF) covariance matrix of standardized regression coefficients: Theoretical extensions and finite sample behavior. *Psychometrika*, 80, 365-378.

**Examples**

```
set.seed(123)

R <- matrix(.5, 3, 3)
diag(R) <- 1
X <- sesamesim[, c("peabody", "prenumb", "postnumb")]
y <- sesamesim$age
results <- seBeta(X, y, Nobs = nrow(sesamesim), alpha = .05, estimator = 'ADF')
print(results, digits = 3)

library(MASS)

set.seed(123)

R <- matrix(.5, 3, 3)
diag(R) <- 1
X <- mvrnorm(n = 200, mu = rep(0, 3), Sigma = R, empirical = TRUE)
Beta <- c(.2, .3, .4)
y <- X %*% Beta + .64 * scale(rnorm(200))
results <- seBeta(X, y, Nobs = 200, alpha = .05, estimator = 'ADF')
print(results, digits = 3)
```

sesamesim

*Simulated Sesame Street Data***Description**

This is a simulated counterpart of part of the Sesame Street data presented by Stevens (1996, Appendix A) concerning the effect of the first year of the Sesame street series on the knowledge of 240 children in the age range 34 to 69 months. We will use the following variables: sex; site of child's origin; setting in which Sesame Street is watched; age; whether or not the child is encouraged to watch; Peabody mental age score; score on numbers test before, after and in a follow up measurement; and scores on knowledge of body parts, letters, forms, numbers, relations, and classifications, both before and after watching Sesame Street for a year.

**Usage**

```
data(sesamesim)
```

**Format**

A data frame with 240 rows and 21 variables.

**Details**

<b>sex</b>	integer	Sex of the child; 1 = boy, 2 = girl
<b>site</b>	integer	Site of the child's origin; 1 = disadvantaged inner city, 2 = advantaged suburban , 3 = advantaged rural
<b>setting</b>	integer	Setting in which the child watches Sesame Street; 1 = at home, 2 = at school
<b>age</b>	integer	Age of the child in months
<b>viewenc</b>	integer	Whether or not the child is encouraged to watch Sesame Street; 0 = no, 1 = yes
<b>peabody</b>	integer	Peabody mental age score of the child; the higher the score the higher the mental age
<b>prenumb</b>	integer	score on a numbers test before watching Sesame Street for a year
<b>postnumb</b>	integer	score on a numbers test after watching Sesame Street for a year
<b>funumb</b>	integer	follow up numbers test score measured one year after postnumb
<b>Bb</b>	integer	Knowledge of body parts before
<b>Bl</b>	integer	Knowledge of letters before
<b>Bf</b>	integer	Knowledge of forms before
<b>Bn</b>	integer	Knowledge of numbers before
<b>Br</b>	integer	Knowledge of relations before
<b>Bc</b>	integer	Knowledge of classifications before
<b>Ab</b>	integer	Knowledge of body parts after
<b>Al</b>	integer	Knowledge of letters after
<b>Af</b>	integer	Knowledge of forms after
<b>An</b>	integer	Knowledge of numbers after
<b>Ar</b>	integer	Knowledge of relations after
<b>Ac</b>	integer	Knowledge of classifications after

## References

Stevens, J. (1996). Applied Multivariate Statistics for the Social Sciences. Mahwah NJ: Lawrence Erlbaum.

---

t_test	<i>Student's t-test</i>
--------	-------------------------

---

## Description

This function is a wrapper for the function `t.test`, which returns group-specific sample sizes and variances, in addition to the usual output of `t.test`.

## Usage

```
t_test(x, ...)
```

## Arguments

`x` An object for which an S3 method of `t.test` exists (vector or formula).  
`...` arguments passed to `t.test`.

## Details

This wrapper allows users to enjoy the functionality of `bain` with the familiar interface of the stats-function `t.test`.

For more documentation, see `t.test`.

## Value

A list with class "t\_test" containing the following components:

<code>statistic</code>	the value of the t-statistic.
<code>parameter</code>	the degrees of freedom for the t-statistic.
<code>p.value</code>	the p-value for the test.
<code>conf.int</code>	a confidence interval for the mean appropriate to the specified alternative hypothesis.
<code>estimate</code>	the estimated mean or difference in means depending on whether it was a one-sample test or a two-sample test.
<code>null.value</code>	the specified hypothesized value of the mean or mean difference depending on whether it was a one-sample test or a two-sample test.
<code>alternative</code>	a character string describing the alternative hypothesis.
<code>method</code>	a character string indicating what type of t-test was performed.
<code>data.name</code>	a character string giving the name(s) of the data.
<code>v</code>	The variance or group-specific variances.
<code>n</code>	The sample size, or group-specific sample size.



**See Also**[t.test](#)**Examples**

```
tmp <- t_test(extra ~ group, data = sleep)
tmp$n
tmp$v
tmp2 <- t_test(extra ~ group, data = sleep)
tmp2$n
tmp2$v
tmp <- t_test(extra ~ group, data = sleep, paired = TRUE)
tmp$n
tmp$v
tmp2 <- t_test(extra ~ group, data = sleep, paired = TRUE)
tmp2$n
tmp2$v
t_test(sesamesim$postnumb)
tmp <- t_test(sesamesim$prenumb)
tmp$n
tmp$v
tmp2 <- t_test(sesamesim$prenumb)
tmp2$n
tmp2$v
tmp <- t_test(sesamesim$prenumb, sesamesim$postnumb)
tmp$n
tmp$v
tmp2 <- t_test(sesamesim$prenumb, sesamesim$postnumb)
tmp2$n
tmp2$v
tmp <- t_test(sesamesim$prenumb, sesamesim$postnumb, paired = TRUE)
tmp$n
tmp$v
tmp2 <- t_test(sesamesim$prenumb, sesamesim$postnumb, paired = TRUE)
tmp2$n
tmp2$v
```

# Index

- \* **Statistics**

- seBeta, 5

- \* **datasets**

- sesamesim, 7

- \* **htest**

- t\_test, 8

bain, 2, 4

bain\_sensitivity, 4

seBeta, 5

sesamesim, 7

t.test, 8, 9

t\_test, 8