Package ‘BayesianPower’

June 9, 2020

Type Package
Title Sample Size and Power for Comparing Inequality Constrained Hypotheses
Version 0.2.0
Author Fayette Klaassen
Maintainer Fayette Klaassen <klaassen.fayette@gmail.com>
Description A collection of methods to determine the required sample size for the evaluation of inequality constrained hypotheses by means of a Bayes factor. Alternatively, for a given sample size, the unconditional error probabilities or the expected conditional error probabilities can be determined. Additional material on the methods in this package is available in Klaassen, F., Hoijtink, H. & Gu, X. (2019) <doi:10.31219/osf.io/d5kf3>.
License LGPL-3
Encoding UTF-8
LazyData true
RoxygenNote 7.1.0
Suggests knitr, rmarkdown, testthat
VignetteBuilder knitr
NeedsCompilation no
Repository CRAN
Date/Publication 2020-06-09 16:00:03 UTC

R topics documented:

bayes_error .............................................................. 2
bayes_power ............................................................. 2
bayes_sampsize .......................................................... 4
calc_bf ................................................................. 5
calc_fc ................................................................. 6
eval_const .............................................................. 6
samp_bf ................................................................. 7
samp_dist ............................................................... 7
bayes_error

Determine the unconditional error probabilities for a set of simulated Bayes factors.

Usage

bayes_error(BFs1, BFs2, bound1 = 1, bound2 = 1/bound1)

Arguments

- **BFs1**: A vector. Simulated BF12 under H1 for a given n
- **BFs2**: A vector. Simulated BF12 under H2 for a given n
- **bound1**: A number. The boundary above which BF12 favors H1
- **bound2**: A number. The boundary below which BF12 favors H2

Value

A named vector. The Type 1, Type 2, Decision error and Area of Indecision probabilities and the median Bayes factors under H1 and H2

bayes_power

Determine the 'power' for a Bayesian hypothesis test

Usage

bayes_power(
  n,
  h1,
  h2,
  m1,
  m2,
  sd1 = 1,
  sd2 = 1,
  ngroup = NULL,
  scale = 1000,
  bound1 = 1,
bayes_power

```
bound2 = 1/bound1,
datasets = 1000,
nsamp = 1000,
seed = NULL
```

**Arguments**

- **n**: A number. The sample size
- **h1**: A constraint matrix defining H1
- **h2**: A constraint matrix defining H2
- **m1**: A vector of expected population means under H1
- **m2**: A vector of expected population means under H2. m2 must be of same length as m1
- **sd1**: A vector of standard deviations under H1. Must be a single number (equal standard deviation under all populations), or a vector of the same length as m1
- **sd2**: A vector of standard deviations under H2. Must be a single number (equal standard deviation under all populations), or a vector of the same length as m2
- **ngroup**: A number or NULL. The number of groups. If NULL, the number of groups is determined from the length of m1
- **scale**: A number specifying the prior scale
- **bound1**: A number. The boundary above which BF12 favors H1
- **bound2**: A number. The boundary below which BF12 favors H2
- **datasets**: A number. The number of datasets to compute the error probabilities
- **nsamp**: A number. The number of prior or posterior samples to determine the fit and complexity
- **seed**: A number. The random seed to be set

**Value**

The Type 1, Type 2, Decision error and Area of Indecision probability and the median BF12s under H1 and H2

**Examples**

```R
# Short example WITH SMALL AMOUNT OF SAMPLES
h1 <- matrix(c(1,-1,0,0,1,-1), nrow= 2, byrow= TRUE)
h2 <- "c"
m1 <- c(.4,.2,0)
m2 <- c(.2,0,.1)
bayes_power(40, h1, h2, m1, m2, datasets = 50, nsamp = 50)
```
bayes_sampsize  

Determine the required sample size for a Bayesian hypothesis test

Description
Determine the required sample size for a Bayesian hypothesis test

Usage
bayes_sampsize(
  h1,  
h2,  
m1,  
m2,  
sd1 = 1,  
sd2 = 1,  
scale = 1000,  
type = 1,  
cutoff,  
bound1 = 1,  
bound2 = 1/bound1,  
datasets = 1000,  
nsamp = 1000,  
minss = 2,  
maxss = 1000,  
seed = 31
)

Arguments

h1  A constraint matrix defining H1.
h2  A constraint matrix defining H2.
m1  A vector of expected population means under H1 (standardized).
m2  A vector of expected populations means under H2 (standardized). m2 must be of same length as m1
sd1  A vector of standard deviations under H1. Must be a single number (equal standard deviation under all populations), or a vector of the same length as m1
sd2  A vector of standard deviations under H2. Must be a single number (equal standard deviation under all populations), or a vector of the same length as m2
scale  A number specifying the prior scale
type  A character. The type of error to be controlled options are: "1", "2", "de", "aoi", "med.1", "med.2"
cutoff  A number. The cutoff criterion for type. If type is "1", "2", "de", "aoi", cutoff must be between 0 and 1 If type is "med.1" or "med.2", cutoff must be larger than 1
**calc_bf**

Compute a Bayes factor

**Description**

Compute a Bayes factor

**Usage**

`calc_bf(data, h1, h2, scale, nsamp = 1000)`

**Arguments**

- `data` A matrix. The dataset for which the BF must be computed
- `h1` A constraint matrix defining H1.
- `h2` A constraint matrix defining H2.
- `scale` A number specifying the prior scale.
- `nsamp` A number. The number of prior or posterior samples to determine the

**Value**

BF12, that is, the evidence for H1 relative to H2
calc_fc  Compute the complexity or fit for two hypotheses.

Description
Compute the complexity or fit for two hypotheses.

Usage
calc_fc(hyp, hyp2, means, sds, nsamp = 1000)

Arguments
- hyp: A constraint matrix defining H1.
- hyp2: A constraint matrix defining H2 OR a character 'u' or 'c' specifying an unconstrained or complement hypothesis
- means: A vector of posterior or prior means
- sds: A vector or posterior or prior standard deviation
- nsamp: A number. The number of prior or posterior samples to determine the fit and complexity

Value
A vector. The proportion of posterior samples in agreement with H1 and with H2

eval_const  Evaluate a constraint matrix for a set of prior/posterior samples

Description
Evaluate a constraint matrix for a set of prior/posterior samples

Usage
eval_const(hyp, samples)

Arguments
- hyp: A constraint matrix defining a hypothesis.
- samples: A matrix. Prior or posterior samples, the number of columns corresponds to the number of groups, the number of rows the number of samples

Value
A number between 0 and 1. The proportion of samples in which the constraints are met.
### samp_bf

**Sample multiple datasets and compute the Bayes factor in each**

**Description**

Sample multiple datasets and compute the Bayes factor in each

**Usage**

```r
samp_bf(datasets, n, ngroup, means, sds, h1, h2, scale, nsamp)
```

**Arguments**

- `datasets`: A number. The number of datasets to simulate for each sample size `n`
- `n`: A number. The group sample size to be used in data simulation
- `ngroup`: A number. The number of groups.
- `means`: A vector of expected population means.
- `sds`: A vector of expected population standard deviations. Note, when standardized, this is a vector of 1s
- `h1`: A constraint matrix defining H1.
- `h2`: A constraint matrix defining H2.
- `scale`: A number specifying the prior scale.
- `nsamp`: A number. The number of samples for the fit and complexity. See `?BayesianPower::calc_fc`

**Value**

A vector of Bayes factors BF12 for each of the simulated datasets

### samp_dist

**Sample from prior or posterior distribution**

**Description**

Sample from prior or posterior distribution

**Usage**

```r
samp_dist(nsamp, means, sds)
```
Arguments
nsamp A number. The number of prior or posterior samples to determine the fit and complexity
means A vector. The prior or posterior means for each group
sds A number or a vector. The standard deviations for each group. If a number is used, the same prior or posterior standard deviation is used for each group.

Value
A matrix of nsamp rows and as many columns as the length of means.
Index

bayes_error, 2
bayes_power, 2
bayes_sampsize, 4

calc_bf, 5
calc_fc, 6

eval_const, 6

samp_bf, 7
samp_dist, 7