Package 'ssev'

May 9, 2026

Title Sample Size Computation for Fixed N with Optimal Reward

Version 0.1.0

Description Computes the optimal sample size for various 2-group designs (e.g., when comparing the means of two groups assuming equal variances, unequal variances, or comparing proportions) when the aim is to maximize the rewards over the full decision procedure of a) running a trial (with the computed sample size), and b) subsequently administering the winning treatment to the remaining N-n units in the population. Sample sizes and expected rewards for standard t- and z- tests are also provided.

Depends R (>= 3.4)

License GPL-3

Encoding UTF-8

LazyData true

Imports pwr, MESS, stats

RoxygenNote 6.1.1

NeedsCompilation no

Author Maurits Kaptein [aut, cre]

Maintainer Maurits Kaptein <maurits@mauritskaptein.com>

Repository CRAN

Date/Publication 2019-01-25 17:30:03 UTC

Contents

compute_sample_size *Compute sample size*

Description

Function to compute the optimal sample size for a comparison of two means (with equal or unequal variances) or proportions. Function returns the standard sample size for an RCT with the specified power, as well as the optimal sample size for a population of size N.

Usage

```
compute_sample_size(means = NULL, sds = NULL, proportions = NULL,
  N = Inf, power = 0.8, sig.level = 0.05, ties = 0.5,
  .verbose = FALSE, ...)
```

Arguments

means	A vector of length 2 containing the (assumed) means of the two groups
sds	A vector containing the (assumed) standard deviations of the two groups. When only one element is supplied equal variances are assumed.
proportions	A vector of length 2 containing the (assumed) proportions of the two groups
N	Estimated population size
power	Desired power for the classical RCT
sig.level	Significance level of the test used (alpha)
ties	Probability of choosing the first group in case of a tie (i.e., H0 is not rejected)
.verbose	Whether or not verbose output should be provided, default FALSE
...	further arguments passed to or from other methods.

Value

An object of type ssev

Examples

```
compute_sample_size(means=c(0,1), sds=2, N=100)
compute_sample_size(means=c(0,1), sds=2, N=10000, power=.9)
compute_sample_size(means=c(0,1), sds=c(1,2), N=10000)
compute_sample_size(proportions=c(.5,.7), N=5000)
```

ev_means_equal *Compute expected value as function of n, N*

Description

Comparing means with equal variances

Usage

```
ev_means_equal(n, N, means, sd, sig.level, ties)
```

Arguments

n	Sample size per group
N	Population size (estimate)
means	Vector of estimated means
sd	Standard deviation of the groups (assumed equal)
sig.level	Significance level
ties	Tie-breaking probability

Value

A scalar indicating the expected mean reward per unit in the population

ev_means_unequal *Compute expected value as function of n, N*

Description

Comparing means with unequal variances

Usage

```
ev_means_unequal(n, N, means, sds, sig.level, ties)
```

Arguments

n	Sample size per group
N	Population size (estimate)
means	Vector of estimated means
sds	Vector of standard deviation of the groups
sig.level	Significance level
ties	Tie-breaking probability

Value

A scalar indicating the expected mean reward per unit in the population

---

`ev_proportions` *Compute expected value as function of n, N*

Description

Comparing proportions

Usage

```
ev_proportions(n, N, proportions, sig.level, ties)
```

Arguments

`n`	Sample size per group
`N`	Population size (estimate)
`proportions`	Vector of two proportions
`sig.level`	Significance level
`ties`	Tie-breaking probability

Value

A scalar indicating the expected mean reward per unit in the population

---

`print.ssev` *Pretty printing of ssev object*

Description

Pretty printing of ssev object

Usage

```
## S3 method for class 'ssev'
print(x, digits = getOption("digits"), ...)
```

Arguments

`x`	Object of type ssev for pretty printing
`digits`	Standard number of digits for pretty printing, default is getOption("digits")
`...`	further arguments passed to or from other methods.

Index