Package ‘schuirmann.constant’

October 29, 2020

Version 0.1.0

Title Asymptotic Properties of the TOST and the Schuirmann Constant

Description TOST power and sample size calculations are provided in the balanced, unpaired setting, where the true mean-difference is assumed to follow a predefined a-priori distribution, e.g. the uniform distribution. Every a-priori distribution corresponds asymptotically to a classical point mean-difference, its Schuirmann constant, which does not depend on the true standard deviation. Functions for calculating different asymptotic properties of the TOST, such as the Schuirmann constant of any a-priori distribution, are implemented as well.

License GPL (>= 2)

Imports PowerTOST, Rdpack, stats

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

RdMacros Rdpack

Author Christian Palmes [aut, cre] (<christian.palmes@gmx.de>), Benedikt Funke [aut] (<b.funke@fh-aachen.de>), Tobias Bluhmki [ctb] (<tobias.bluhmki@uni-ulm.de>), Erich Bluhmki [ctb] (<erich.bluhmki@boehringer-ingelheim.com>)

Maintainer Christian Palmes <christian.palmes@gmx.de>

R topics documented:

`apriori.density` *Returns a standard a-priori density function*

Description

Let U denote the uniform distribution on $[0, 1]$, T the triangle distribution with center point 0.5 and full support on $[0, 1]$ and HT the half-triangle distribution with center point 0.5 and support on $[0.25, 0.75]$. This function returns one of these a-priori densities in dependence of its input parameter. These three a-priori distributions are deemed as helpful in experimenting with the Schuirmann constant.

Usage

```
apriori.density(density.id)
```

Arguments

`density.id`	'U', 'T' or 'HT', depending on which a-priori density should be returned

Value

Uniform (U), triangle (T) or half triangle (HT) a-priori density function

Examples

```
rho <- apriori.density('T')
r <- seq(0, 1, length = 100)
plot(r, rho(r), type = 'l')
```

`n.point` *Exact sample size; point mean-difference*

Description

Exact sample size calculation of a balanced, unpaired TOST. The result n are the samples in one group. The true mean-difference is assumed to be a point, i.e. fixed value.

Usage

```
n.point(alpha, theta1, theta2, theta0, sigma, pwr)
```

Arguments

`alpha`	type I error
`theta1`	lower limit of equivalence interval
`theta2`	upper limit of equivalence interval
`theta0`	mean-difference
`sigma`	standard deviation of the measurements
`pwr`	Exact power of the TOST

Value

sample size in each group

Examples

```
pwr <- power.point(alpha = 0.05, theta1 = -1, theta2 = 2, theta0 = 1, sigma = 1, n = 15)
n.point(alpha = 0.05, theta1 = -1, theta2 = 2, theta0 = 1, sigma = 1, pwr = pwr)
```

`n.weighted` *Exact sample size; weighted mean-difference*

Description

Exact sample size calculation of a balanced, unpaired TOST, where the true mean-difference follows a prescribed a-priori distribution. The lowest sample size n that exceeds the target power pwd is calculated using a standard binary search algorithm. The result n is the sample size in one group.

Usage

```
n.weighted(alpha, theta1, theta2, density, sigma, pwr)
```

Arguments

`alpha`	type I error
`theta1`	lower limit of equivalence interval
`theta2`	upper limit of equivalence interval
`density`	a-priori density for the mean-difference
`sigma`	standard deviation of the measurements
`pwr`	Exact power of the TOST

Value

sample size in each group

Examples

```
rho <- apriori.density('U')
pwr <- power.weighted(alpha = 0.05, theta1 = -2, theta2 = 2, density = rho, sigma = 1, n = 22)
n.weighted(alpha = 0.05, theta1 = -2, theta2 = 2, density = rho, sigma = 1, pwr = pwr)
```

nabla.point *nabla; point mean-difference*

Description

The equality

$$power_{asymp}(\nabla, r, \alpha) = pwr$$

is solved for ∇ in dependence of α, pwr and r.

Usage

```
nabla.point(alpha, pwr, r)
```

Arguments

`alpha`	type I error
`pwr`	asymptotic power of the TOST
`r`	relative mean-difference

Value

∇ for a fixed point mean-difference

Examples

```
nabla <- nabla.point(alpha = 0.05, pwr = 0.8, r = 0.55)
power.point.asymp(alpha = 0.05, r = 0.55, nabla = nabla)
```

nabla.to.standard *TOST sensitivity to standard parameterization*

Description

The TOST sensitivity parameterization is transformed to the standard parameterization.

Usage

```
nabla.to.standard(r, nabla, theta1, theta2, n)
```

Arguments

`r`	relative mean-difference
`nabla`	TOST sensitivity index
`theta1`	lower limit of equivalence interval
`theta2`	upper limit of equivalence interval
`n`	sample size in each group

Value

A list with the transformed parameters in standard parameterization, i.e. $(\theta_1, \theta_2, \theta_0, \sigma, n)$

Examples

```
l <- nabla.to.standard(r = 0.75, nabla = 10, theta1 = 0, theta2 = 4, n = 20)
standard.to.nabla(l$theta1,l$theta2,l$theta0, l$sigma, l$n)
```

---

nabla.weighted *nabla; weighted mean-difference*

Description

The equality

$$power_{asymp,density}(\nabla, \alpha) = pwr$$

is solved for ∇ in dependence of α, pwr and $density$.

Usage

```
nabla.weighted(alpha, pwr, density)
```

Arguments

alpha	type I error
pwr	asymptotic weighted power of the TOST
density	a-priori density for the mean-difference

Value

∇ for a weighted mean-difference

Examples

```
rho <- apriori.density('U')
nabla <- nabla.weighted(alpha = 0.05, pwr = 0.8, density = rho)
power.weighted.asymp(alpha = 0.05, density = rho, nabla = nabla)
```

power.point *Exact power; point mean-difference; standard parameterization*

Description

Calculates the exact power of an unpaired, balanced TOST with n samples in each group. The true mean-difference is assumed to be a point, i.e. fixed value.

Usage

```
power.point(alpha, theta1, theta2, theta0, sigma, n)
```

Arguments

alpha	type I error
theta1	lower limit of equivalence interval
theta2	upper limit of equivalence interval
theta0	mean-difference
sigma	standard deviation of the measurements
n	sample size in each group

Value

Exact power of the TOST for a fixed point mean-difference

Examples

```
pwr <- power.point(alpha = 0.05, theta1 = -1, theta2 = 2, theta0 = 1, sigma = 1, n = 15)
n.point(alpha = 0.05, theta1 = -1, theta2 = 2, theta0 = 1, sigma = 1, pwr = pwr)
```

power.point.asymp *Asymptotical TOST power; point mean-difference*

Description

Calculates the asymptotical power of an unpaired, balanced TOST for a fixed point mean-difference.

Usage

```
power.point.asymp(alpha, r, nabla)
```

Arguments

alpha	type I error
r	relative mean-difference
nabla	TOST sensitivity index

Value

Asymptotic power of the TOST for a fixed point mean-difference

Examples

```
power.point.nabla(alpha = 0.05, r = 2/3, nabla = 10, n = 10)
power.point.nabla(alpha = 0.05, r = 2/3, nabla = 10, n = 50)
power.point.asymp(alpha = 0.05, r = 2/3, nabla = 10)
```

---

power.point.nabla *Exact power; point mean-difference; sensitivity parameterization*

Description

Calculates the exact power of an unpaired, balanced TOST with n samples in each group in the TOST sensitivity parameterization. The true mean-difference is assumed to be a point, i.e. fixed value.

Usage

```
power.point.nabla(alpha, r, nabla, n)
```

Arguments

alpha	type I error
r	relative mean-difference
nabla	TOST sensitivity index
n	sample size in each group

Value

Exact power of the TOST for a fixed point mean-difference

Examples

```
power.point(alpha = 0.05, theta1 = -1, theta2 = 2, theta0 = 1, sigma = 1, n = 15)
l <- standard.to.nabla(theta1 = -1, theta2 = 2, theta0 = 1, sigma = 1, n = 15)
power.point.nabla(alpha = 0.05, r = l$r, nabla = l$nabla, n = 15)
```

power.weighted *Exact weighted power; standard parameterization*

Description

Calculates the exact power of an unpaired, balanced TOST with n samples in each group, where the true mean-difference is modeled with an a-priori density rather than a fixed mean-difference.

Usage

```
power.weighted(alpha, theta1, theta2, density, sigma, n)
```

Arguments

`alpha`	type I error
`theta1`	lower limit of equivalence interval
`theta2`	upper limit of equivalence interval
`density`	a-priori density for the mean-difference
`sigma`	standard deviation of the measurements
`n`	sample size in each group

Value

Exact power of the TOST for a weighted true mean-difference

Examples

```
rho <- apriori.density('U')
pwr <- power.weighted(alpha = 0.05, theta1 = -2, theta2 = 2, density = rho, sigma = 1, n = 22)
n.weighted(alpha = 0.05, theta1 = -2, theta2 = 2, density = rho, sigma = 1, pwr = pwr)
```

power.weighted.asymp *Asymptotical weighted power*

Description

Calculates the asymptotical power of an unpaired, balanced TOST for a weighted a-priori mean-difference.

Usage

```
power.weighted.asymp(alpha, density, nabla)
```

Arguments

`alpha`	type I error
`density`	a-priori density for the mean-difference
`nabla`	TOST sensitivity index

Value

Asymptotic power of the TOST for a weighted mean-difference

Examples

```
rho <- apriori.density('HT')
power.weighted.nabla(alpha = 0.05, density = rho, nabla = 7, n = 10)
power.weighted.nabla(alpha = 0.05, density = rho, nabla = 7, n = 50)
power.weighted.asymp(alpha = 0.05, density = rho, nabla = 7)
```

---

power.weighted.nabla *Exact weighted power; sensitivity parameterization*

Description

Calculates the exact power of an unpaired, balanced TOST with n samples in each group, where the true mean-difference is modeled with an a-priori density rather than a fixed mean-difference. The calculation is performed in the TOST sensitivity parameterization.

Usage

```
power.weighted.nabla(alpha, density, nabla, n)
```

Arguments

alpha	type I error
density	a-priori density for the mean-difference
nabla	TOST sensitivity index
n	sample size in each group

Value

Exact power of the TOST for a weighted mean-difference

Examples

```
rho <- apriori.density('U')
power.weighted.nabla(alpha = 0.05, density = rho, nabla = 8, n = 22)
## Since we are performing a weighted power calculation, the r respectively
## theta0 value is not needed. Thus, the following calculation does not depend
## on r.
l <- nabla.to.standard(r = 0.99, nabla = 8, theta1 = -1, theta2 = 4, n = 22)
power.weighted(alpha = 0.05, theta1 = -1, theta2 = 4, density = rho, sigma = l$sigma, n = 22)
```

`schuirmann.constant` *Calculation of the schuirmann-constant*

Description

Calculation of the schuirmann-constant, cf. (Palmes et al. 2020/2021)

Usage

```
schuirmann.constant(alpha, pwr, density)
```

Arguments

`alpha`	type I error
`pwr`	asymptotic weighted power of the TOST
`density`	a-priori density for the mean-difference

Value

The Schuirmann constant

References

Palmes C, Bluhmki T, Funke B, Bluhmki E (2020/2021). "Asymptotic Properties of the Two One-Sided t-Tests - New Insights and the Schuirmann-Constant." *International Journal of Biostatistics*. to appear.

Examples

```
rho <- apriori.density('U')
theta1 <- 0
theta2 <- 3
s <- schuirmann.constant(alpha = 0.05, pwr = 0.8, density = rho)
theta0 <- (1-s)*theta1 + s*theta2
n <- n.point(alpha = 0.05, theta1 = theta1, theta2 = theta2, theta0 = theta0, sigma = 1, pwr = 0.8)
## Note that the weighted power is calculated with a sample size that was found for a
## point-difference. Due to the newly introduced duality concept it is -nevertheless-
## very close to the target power 0.8.
power.weighted(alpha = 0.05, theta1 = theta1, theta2 = theta2, density = rho, sigma = 1, n = n)
```

`schuirmann.constant.uniform` *Calculation of the uniform schuirmann-constant*

Description

Calculation of the uniform schuirmann-constant, cf. (Palmes et al. 2020/2021)

Usage

```
schuirmann.constant.uniform(alpha, pwr, theta1 = 0, theta2 = 1)
```

Arguments

`alpha`	type I error
`pwr`	asymptotic weighted power of the TOST
`theta1`	lower limit of equivalence interval
`theta2`	upper limit of equivalence interval

Details

The uniform density is assumed as a-priori distribution. This density is unique with the property that each possible mean-difference is equally weighted. It represents the lack of any information about the true mean-difference. Due to its importance, this special setting is included as a separate function for convenience. Technical, this function is merely a wrapper that calls the schuirmann.constant function with the uniform distribution as a-priori density.

Value

The Schuirmann constant of the uniform a-priori distribtion is returned. If $[\theta_1, \theta_2]$ differs from $[0, 1]$, then the appropriately scaled true-mean difference

$$\theta_0 = (S - 1) \cdot \theta_1 + S \cdot \theta_2$$

is returned.

References

Palmes C, Bluhmki T, Funke B, Bluhmki E (2020/2021). "Asymptotic Properties of the Two One-Sided t-Tests - New Insights and the Schuirmann-Constant." *International Journal of Biostatistics*. to appear.

Examples

```
rho <- apriori.density('U')
theta1 <- 0
theta2 <- 3
s <- schuirmann.constant(alpha = 0.05, pwr = 0.8, density = rho)
(1-s)*theta1 + s*theta2
schuirmann.constant.uniform(alpha = 0.05, pwr = 0.8, theta1 = theta1, theta2 = theta2)
```

`standard.to.nabla` *standard to TOST sensitivity parameterization*

Description

The standard parameterization is transformed to the TOST sensitivity parameterization.

Usage

```
standard.to.nabla(theta1, theta2, theta0, sigma, n)
```

Arguments

theta1	lower limit of equivalence interval
theta2	upper limit of equivalence interval
theta0	mean-difference
sigma	standard deviation of the measurements
n	sample size in each group

Value

A list with the transformed parameters in TOST sensitivity parameterization, i.e. $(r, \nabla, \theta_1, \theta_2, n)$

Examples

```
l <- nabla.to.standard(r = 0.75, nabla = 10, theta1 = 0, theta2 = 4, n = 20)
standard.to.nabla(l$theta1,l$theta2,l$theta0, l$sigma, l$n)
```

Index