Package ‘ssanv’

October 14, 2022

Type Package

Title Sample Size Adjusted for Nonadherence or Variability of Input Parameters

Version 1.1

Date 2015-06-22

Author Michael P. Fay <`mfay@niaid.nih.gov`>

Maintainer Michael P. Fay <`mfay@niaid.nih.gov`>

Depends R (>= 2.1.1), stats

Description
A set of functions to calculate sample size for two-sample difference in means tests. Does adjustments for either nonadherence or variability that comes from using data to estimate parameters.

License GPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2015-06-23 17:26:38

R topics documented:

`ssanv-package` *Sample Size Adjusted for Nonadherence or Variability of Input Parameters*

Description

Calculate sample size for two sample difference in means tests. The function, `ss.nonadh` accounts for known proportions of nonadherence. There are three functions that calculate sample size accounting for the variability that comes from estimating parameters from data.

Details

Package:	ssanv
Type:	Package
Version:	1.1
Date:	2015-06-22
License:	GPL3

This package does adjustments for sample size calculations for two-sample difference in means tests. The function `ss.nonadh` adjusts for nonadherence, where the responses may be continuous, discrete, or binary. There are 3 functions which account for variability of estimating parameters from data: `ss.fromdata.nvar`– accouts for variability of standard deviation estimate when data are normal and the clincially significant difference is known, `ss.fromdata.neff`– accounts for variability of estimated standardized difference in means when both groups come from normal distributions with the same variance, `ss.fromdata.pois`– accounts for variability of means when both means are estimated from data assumed Poisson.

Author(s)

Michael Fay <mfay@niaid.nih.gov>

Maintainer: Michael Fay <mfay@niaid.nih.gov>

References

Fay, M.P., Halloran, M.E., and Follmann, D.A. (2007). ‘Accounting for Variability in Sample Size Estimation with Applications to Nonadherence and Estimation of Variance and Effect Size’ Biometrics 63: 465-474.

Examples

```
### Suppose treatment will not work on about 30 percent of subjects
### then rho0=.3 and clinically significant difference in means is .5
ss.nonadh(delta=.5,rho0=.3)

### Suppose you estimate standard deviation of .32 from a sample of size 12
### and the clinically significant difference is .2
```

```
ss.fromdata.nvar(delta=.2,sdhat=.32,df=11)
### The usual method is to assume that the standard deviation is known
### this result is the same as df=Inf
ss.fromdata.nvar(delta=.2,sdhat=.32,df=Inf)
```

`example.of.Fisher.exact` *Object of class 'power.htest'*

Description

Result from ss.nonadh(mu0=.1,mu1=.4,refinement="Fisher.exact"). It took about 8 seconds to calculate.

Usage

```
data(example.of.Fisher.exact)
```

Details

Compared with StatXact, gives same sample sizes.

`find.calibrated.beta` *Find calibrated beta*

Description

Called by `ss.fromdata.nvar`. Finds calibrated beta (1-power) for use when estimating standard deviation from data in sample size calculations for differences in two normal means.

Usage

```
find.calibrated.beta(beta, df, alpha = 0.05)
```

Arguments

`beta`	1-nominal.power
`df`	degrees of freedom
`alpha`	one-sided significance level of test

Details

See `ss.fromdata.nvar`.

Value

Calibrated beta value = 1 - calibrated power.

Author(s)

Michael P. Fay

References

Fay, M.P., Halloran, M.E., and Follmann, D.A. (2007). ‘Accounting for Variability in Sample Size Estimation with Applications to Nonadherence and Estimation of Variance and Effect Size’ Biometrics 63: 465-474.

See Also

`ss.fromdata.nvar`

Examples

```
## see Table 2 of Fay et al 2006.
1-find.calibrated.beta(.2,12,.025)
```

`ss.fromdata.neff` *Find sample sizes when normal standardized difference is estimated from data*

Description

Calculate sample sizes for two-sample differences in normal means when standardized difference is estimated from existing data

Usage

```
ss.fromdata.neff(thetahat, m0, m1, ss.ratio = 1, thetaB = 0,
    sig.level = 0.05, real.power = 0.8, nominal.power = NULL,
    alternative = c("two.sided", "one.sided"),
    MINN0 = 2, MAXN0 = Inf, subdivisions = 1000)
```

Arguments

`thetahat`	estimated standardized difference in means
`m0`	sample size from control group of existing data
`m1`	sample size from treatment group of existing data
`ss.ratio`	n1/n0, where n0 (n1) is sample size of control (treatment) group for proposed study
`thetaB`	boundary value between null and alternative hypotheses for one-sided tests (see details)
`sig.level`	significance level (Type I error)
`real.power`	minimum power that you want the sample size to achieve, only .8 or .9 allowed

`nominal.power`	see details
`alternative`	One- or two-sided test
`MINN0`	minimum sample size for control group
`MAXN0`	maximum sample size for control group
`subdivisions`	number of subdivisions for numerical integration

Details

Calculates the sample sizes for a study designed to test the difference between the means of two groups, where it is assumed that the responses from each group are distributed normally with the same variance. The standardized difference in means (`thetahat`) is estimated from existing data that is assumed to also follow the same normal distribution. The method is inherently conservative, so that with a nominal power of .76 the real power will be about .80, and a nominal power of .88 the real power will be about .90. Other values of nominal power are allowed, but only real powers of .80 or .90 are allowed. The one-sided tests are designed to test either $H_0 : \theta \leq \theta_B$ vs. $H_1 : \theta > \theta_B$ or to test $H_0 : \theta \geq \theta_B$ vs. $H_1 : \theta < \theta_B$. The choice of hypotheses is determined by the value of `thetahat`; if `thetahat` > `thetaB` then the former hypotheses are tested, otherwise the latter are. See Fay, Halloran and Follmann (2007) for details.

Value

Object of class "power.htest", a list of the arguments (including the computed sample sizes) augmented with 'METHOD' and 'NOTE' elements. The values 'n0' and 'n1' are the samples sizes for the two groups, rounded up to the nearest integer.

Note

The function `ss.fromdata.neff` calls `uniroot.integer`, a function written for this package that finds the nearest integer to the root.

Author(s)

Michael P. Fay

References

Fay, M.P., Halloran, M.E., and Follmann, D.A. (2007). 'Accounting for Variability in Sample Size Estimation with Applications to Nonadherence and Estimation of Variance and Effect Size' Biometrics 63: 465-474.

See Also

`ss.fromdata.nvar`, `ss.fromdata.pois`, `ss.nonadh`, `uniroot.integer`

Examples

```
ss.fromdata.neff(.588,23,25)
```

`ss.fromdata.nvar` *Find sample sizes when normal standard deviation is estimated from data*

Description

Calculate sample sizes for two-sample differences in normal means when the standard deviation (or variance) is estimated from existing data.

Usage

```
ss.fromdata.nvar(delta, sdhat = NULL, vhat = NULL,
    df = Inf, ss.ratio = 1, var.ratio = 1, deltaB = 0,
    sig.level = 0.05, power = 0.8,
    alternative = c("two.sided", "one.sided"))
```

Arguments

`delta`	clinically significant difference in means
`sdhat`	estimate of standard deviation from existing data (must supply either sdhat or vhat)
`vhat`	estimate of variance from existing data (must supply either sdhat or vhat)
`df`	degrees of freedom associated with standard deviation (or variance) estimate
`ss.ratio`	n1/n0, where n0 (n1) is sample size of control (treatment) group for proposed study
`var.ratio`	σ_1^2/σ_0^2, where σ_0 (σ_1) is standard deviation of control (treatment) group for proposed study
`deltaB`	boundary value between null and alternative hypotheses for one-sided tests (see details)
`sig.level`	significance level (Type I error)
`power`	minimum power that you want the sample size to achieve
`alternative`	One- or two-sided test

Details

Calculates the sample sizes for a study designed to test the difference between the means of two groups, where it is assumed that the responses from each group are distributed normally. The standard deviation (sdhat) or variance (vhat) is estimated from existing data that is assumed to also follow a normal distribution with variance the same as the control group of the proposed study. If sdhat (or vhat) is estimated from one group with a sample size of m, then df=m-1. If sdhat (or vhat) is estimated from two groups with sample sizes of m0 and m1, then df=m0+m1-2.

The one-sided tests are designed to test either $H_0 : \delta \leq \delta_B$ vs. $H_1 : \delta > \delta_B$ or to test $H_0 : \delta \geq \delta_B$ vs. $H_1 : \delta < \delta_B$. The choice of hypotheses is determined by the value of `delta`; if `delta` > `deltaB` then the former hypotheses are tested, otherwise the latter are.

See Fay, Halloran and Follmann (2007) for details.

Value

Object of class "power.htest", a list of the arguments (including the computed sample sizes) augmented with 'METHOD' and 'NOTE' elements. The values 'n0' and 'n1' are the samples sizes for the two groups, rounded up to the nearest integer.

Note

The function `ss.fromdata.nvar` calls `find.calibrated.beta`, a function written for this package that finds the calibrated beta value (see Fay, Halloran and Follmann, 2007).

Author(s)

Michael P. Fay

References

Fay, M.P., Halloran, M.E., and Follmann, D.A. (2007). 'Accounting for Variability in Sample Size Estimation with Applications to Nonadherence and Estimation of Variance and Effect Size' Biometrics 63: 465-474.

See Also

`ss.fromdata.neff`, `ss.fromdata.pois`, `ss.nonadh`, `find.calibrated.beta`

Examples

```
ss.fromdata.nvar(.4,sdhat=.682,df=46)
```

`ss.fromdata.pois` *Find sample sizes when 2 Poisson means are estimated from data*

Description

Calculate sample sizes for two-sample differences in Poisson means when means are estimated from existing data

Usage

```
ss.fromdata.pois(xbar0, xbar1, m0, m1, ss.ratio = 1, sig.level = 0.05,
    real.power = 0.8, nominal.power = NULL,
    alternative = c("two.sided", "one.sided"), MINN0 = 1, MAXN0 = 10^5)
```

Arguments

`xbar0`	mean from control group of existing data
`xbar1`	mean from treatment group of existing data
`m0`	sample size of control group of existing data
`m1`	sample size of treatment group of existing data
`ss.ratio`	n1/n0, where n0 (n1) is sample size of control (treatment) group for proposed study
`sig.level`	significance level (Type I error)
`real.power`	minimum power that you want the sample size to achieve, only .8 or .9 allowed
`nominal.power`	see details
`alternative`	One- or two-sided test
`MINN0`	minimum sample size for control group
`MAXN0`	maximum sample size for control group

Details

Calculates the sample sizes for a study designed to test the difference between the means of two groups, where it is assumed that the responses from both groups are distributed Poisson. The means from each group (`xbar0` and `xbar1`) come from existing data that is assumed to also follow the same Poisson distributions. The method is inherently conservative, so that with a nominal power of .77 the real power will be about .80, and a nominal power of .89 the real power will be about .90. Other values of nominal power are allowed, but only real powers of .80 or .90 are allowed. If mu0 and mu1 are the means from the two groups, the one-sided tests are designed to test either $H_0 : \mu_0 \leq \mu_1$ vs. $H_1 : \mu_0 > \mu_1$ or to test $H_0 : \mu_0 \geq \mu_1$ vs. $H_1 : \mu_0 < \mu_1$. We estimate μ_0 and μ_1 with $\hat{\mu}_0 = xbar0 + \frac{1}{2m_0}$ and $\hat{\mu}_1 = xbar1 + \frac{1}{2m_1}$. The choice of hypotheses is determined by the value of $\hat{\mu}_0$ and $\hat{\mu}_1$; if $\hat{\mu}_0 > \hat{\mu}_1$ then the former hypotheses are tested, otherwise the latter are. See Fay, Halloran and Follmann (2007) for details.

Value

Object of class "power.htest", a list of the arguments (including the computed sample sizes) augmented with 'METHOD' and 'NOTE' elements. The values 'n0' and 'n1' are the samples sizes for the two groups, rounded up to the nearest integer.

Note

The function `ss.fromdata.pois` calls `uniroot.integer`, a function written for this package that finds the nearest integer to the root.

Author(s)

Michael P. Fay

References

Fay, M.P., Halloran, M.E., and Follmann, D.A. (2007). ‘Accounting for Variability in Sample Size Estimation with Applications to Nonadherence and Estimation of Variance and Effect Size’ Biometrics 63: 465-474.

See Also

`ss.fromdata.nvar`, `ss.fromdata.neff`, `ss.nonadh`, `uniroot.integer`

Examples

```
ss.fromdata.pois(1.65,.88,23,25)
```

`ss.nonadh` *Find sample sizes accounting for nonadherence*

Description

Calculate sample sizes needed for a two-sample difference in means test accounting for nonadherence (i.e., noncompliance) in a proportion of subjects.

Usage

```
ss.nonadh(mu0 = NULL, mu1 = NULL, delta = NULL, sigma0.sq = 1,
    rho0 = 0, rho1 = 0, ss.ratio = 1, var.ratio = 1, deltaB = 0,
    sig.level = 0.05, power = 0.8, alternative = c("two.sided", "one.sided"),
    refinement = NULL,error.fisher=10^-6)
```

Arguments

`mu0`	mean of control group
`mu1`	mean of treatment group
`delta`	mu0 - mu1
`sigma0.sq`	variance of control group
`rho0`	probability that a subject randomized to control will switch to treatment
`rho1`	probability that a subject randomized to treatment will switch to control (or respond like control subjects)
`ss.ratio`	n1/n0, where n0 (n1) is sample size of control (treatment) group for proposed study
`var.ratio`	σ_1^2/σ_0^2, where σ_0 (σ_1) is standard deviation of control (treatment) group for proposed study
`deltaB`	boundary value between null and alternative hypotheses for one-sided tests (see details)
`sig.level`	significance level (Type I error)

`power`	minimum power that you want the sample size to achieve
`alternative`	One- or two-sided test
`refinement`	More precise sample size estimates for certain types of responses, possible values: NULL (use asymptotic method), ‘Normal’ or ’N’,’Bernoulli’ or ’B’, and ’Fisher.exact’ or ’F’.
`error.fisher`	bound on the abolute error of the power calculations for the Fisher.exact refinement

Details

Calculates the sample sizes for a study designed to test the difference between the means of two groups, where a proportion of those randomized to the control regimin switch to the treatment regimin (rho0) and a proportion of those randomized to the treatment regimin switch to the control regimin (rho1).

You must supply either `mu0` and `mu1` or `delta`=mu0-mu1. The one-sided tests are designed to test either $H_0 : \delta \leq \delta_B$ vs. $H_1 : \delta > \delta_B$ or to test $H_0 : \delta \geq \delta_B$ vs. $H_1 : \delta < \delta_B$. The choice of hypotheses is determined by the value of `delta`; if `delta` $>$ `deltaB` then the former hypotheses are tested, otherwise the latter are.

The responses may be continuous, discrete, or even binary. The value `refinement` defines different methods related to the response type. When refinement=NULL, the method is based on asymptotic normal approximation. For refinement=’Normal’ the method is based on the t-distribution derived when the responses are normal. For refinement=’Bernoulli’ the `sigma0.sq` and `var.ratio` values are defined by `mu0` and `mu1`, and the sample sizes do not use any continuity correction. For refinement=’Fisher.exact’ only the two-sided alternative is allowed and the sample size related to exact powers for the 2X2 Fisher exact test are calculated.

See Fay, Halloran and Follmann (2007) for details.

Value

Object of class "power.htest", a list of the arguments (including the computed sample sizes) augmented with ’METHOD’ and ’NOTE’ elements. The values ’n0’ and ’n1’ are the samples sizes for the two groups, rounded up to the nearest integer.

Note

If you have a two-sided hypotheses with binary data, then refinement=’Fisher.exact’ is more accurate than refinement=’Bernoulli’, but it takes considerably longer to run. If ’Fisher.exact’ is chosen the method uses the `uniroot.integer` function, and if you ‘Esc’ during the calculation, intermediate exact powers will print.

Author(s)

Michael P. Fay

References

Fay, M.P., Halloran, M.E., and Follmann, D.A. (2007). ‘Accounting for Variability in Sample Size Estimation with Applications to Nonadherence and Estimation of Variance and Effect Size’ Biometrics 63: 465-474.

See Also

`uniroot.integer`, `ss.fromdata.neff`, `ss.fromdata.pois`, `ss.fromdata.nvar`

Examples

```
### Suppose treatment will not work on about 30 percent of subjects
### then rho0=.3 and clinically significant difference in means is .5
ss.nonadh(delta=.5,rho0=.3)

## Binary data, quick calculation
ss.nonadh(mu0=.1,mu1=.4,refinement="B")
# Calculate Fisher.exact before packaging, took about 8 seconds
#example.of.Fisher.exact<-ss.nonadh(mu0=.1,mu1=.4,refinement="Fisher.exact")
data(example.of.Fisher.exact)
example.of.Fisher.exact
```

`uniroot.integer` *Find the root of a function to the nearest integer*

Description

Let f be a monotonic function that changes sign within the interval specified. If f(i)=0 for some i within the interval specified (including the ends of the interval), then the root is i. Otherwise if `pos.side`=TRUE (or FALSE) then `uniroot.integer` finds the integer i such that f(i) is closest to the sign change and is positive (or negative).

Usage

```
uniroot.integer(f, interval, lower = min(interval), upper = max(interval),
    step.power = 6, step.up = TRUE, pos.side = FALSE, print.steps = FALSE,
    maxiter = 1000, ...)
```

Arguments

`f`	function for which a root is needed
`interval`	an interval giving minimum and maximum allowable values for root
`lower`	minimum allowable root
`upper`	maximum allowable root
`step.power`	initial step size is $2^{step.power}$
`step.up`	if TRUE steps up from 'lower', if FALSE steps down from 'upper'
`pos.side`	if TRUE finds integer, i, closest to the root such that f(i) $>$ zero
`print.steps`	if TRUE, prints iterations
`maxiter`	maximum number of iterations
`...`	additional arguments to 'f'.

Details

The algorithm evaluates f(i) iteratively, increasing (or decreasing if step.up=FALSE) i by $2^{step.power}$ until either f(i)=0 or f(i) switches sign. If f(i)=0, then stop. If f(i) switches sign, then the change in 'i' is halved each iteration until convergence.

Value

A list with the following elements:

`root`	the integer on the correct side of the root
`f.root`	value of f at root
`iter`	number of times f was evaluated

Note

Unlike `uniroot`, the function is not automatically evaluated at both extremes. This makes `uniroot.integer` an efficient method to use when the calculation time of f(i) increases with the value of 'i'. For an example of the importance of this see `ss.fromdata.pois`.

Author(s)

Michael P. Fay

See Also

`uniroot`, used by `ss.fromdata.neff`, `ss.fromdata.pois`, `ss.nonadh`

Examples

```
root.func<-function(i) i - 500.1
## initial step sizes = 2^2 =4
uniroot.integer(root.func,c(0,Inf),step.power=2)
## more efficient to use bigger initial step sizes = 2^10 =1024
uniroot.integer(root.func,c(0,Inf),step.power=10,print.steps=TRUE)
```

Index