

Package ‘clusterPower’

September 5, 2017

Version 0.6.111

Date 2017-09-03

Title Power Calculations for Cluster-Randomized and Cluster-Randomized Crossover Trials

License GPL (>= 2)

Imports lme4 (>= 1.0)

Description Calculate power for cluster randomized trials (CRTs) that compare two means, two proportions, or two counts using closed-form solutions. In addition, calculate power for cluster randomized crossover trials using Monte Carlo methods. For more information, see Reich et al. (2012) <doi:10.1371/journal.pone.0035564>.

RoxygenNote 6.0.1

NeedsCompilation no

Author Ken Kleinman [aut, cre],
Jon Moyer [aut],
Nicholas Reich [aut],
Daniel Obeng [ctb]

Maintainer Ken Kleinman <ken.kleinman@gmail.com>

Repository CRAN

Date/Publication 2017-09-05 08:32:09 UTC

R topics documented:

clusterPower	2
crtpwr.2mean	2
crtpwr.2prop	3
crtpwr.2rate	4
expit	5
fixed.effect	6
hayes.power.poisson	7
mixed.eff.params	8
power.sim.normal	9

Index	13
--------------	-----------

clusterPower	<i>clusterPower: doing various power analysis calculations for cluster randomized trials.</i>
--------------	---

Description

clusterPower: doing various power analysis calculations for cluster randomized trials.

crtpr.2mean	<i>Power calculations for simple cluster randomized trials, continuous outcome</i>
-------------	--

Description

Compute the power of a simple cluster randomized trial with a continuous outcome, or determine parameters to obtain a target power.

Usage

```
crtpr.2mean(alpha = 0.05, power = 0.8, m = NA, n = NA, cv = 0,
            d = NA, icc = NA, varw = NA, method = c("taylor", "weighted"),
            tol = .Machine$double.eps^0.25)
```

Arguments

alpha	The level of significance of the test, the probability of a Type I error.
power	The power of the test, 1 minus the probability of a Type II error.
m	The number of clusters per condition. It must be greater than 1.
n	The mean of the cluster sizes, or a vector of cluster sizes for one arm.
cv	The coefficient of variation of the cluster sizes. When $cv = 0$, the clusters all have the same size.
d	The difference in condition means.
icc	The intraclass correlation.
varw	The within-cluster variation.
method	The method for calculating variance inflation due to unequal cluster sizes. Either a method based on Taylor approximation of relative efficiency ("taylor"), or weighting by cluster size ("weighted")
tol	Numerical tolerance used in root finding. The default provides at least four significant digits.

Details

Exactly one of alpha, power, m, n, nsd, d, icc, and varw must be passed as NA. Note that alpha, power, and nsd have non-NA defaults, so if those are the parameters of interest they must be explicitly passed as NA.

If n is a vector the values, m and cv will be recalculated using the values in n. If n is a vector and method is "taylor", the exact relative efficiency will be calculated as described in van Breukelen et al (2007).

Value

The computed argument.

Note

'uniroot' is used to solve power equation for unknowns, so you may see errors from it, notably about inability to bracket the root when invalid arguments are given.

Author(s)

Jonathan Moyer (<jon.moyer@gmail.com>)

References

Eldridge SM, Ukoumunne OC, Carlin JB. (2009) The Intra-Cluster Correlation Coefficient in Cluster Randomized Trials: A Review of Definitions. *Int Stat Rev.* 77: 378-394.

Eldridge SM, Ashby D, Kerry S. (2006) Sample size for cluster randomized trials: effect of coefficient of variation of cluster size and analysis method. *Int J Epidemiol.* 35(5):1292-300.

van Breukelen GJP, Candel MJJM, Berger MPF. (2007) Relative efficiency of unequal versus equal cluster sizes in cluster randomized and multicentre trials. *Statist Med.* 26:2589-2603.

Examples

```
# Find the number of clusters per condition needed for a trial with alpha = .05,
# power = 0.8, 10 observations per cluster, no variation in cluster size, a difference
# of 1 unit, icc = 0.1 and a variance of five units.
crtprw.2mean(n=10 ,d=1, icc=.1, varw=5)
#
# The result, showing m of greater than 15, suggests 16 clusters per condition should be used.
```

crtprw.2prop

Power calculations for simple cluster randomized trials, binary outcome

Description

Compute the power of a simple cluster randomized trial with a binary outcome, or determine parameters to obtain a target power.

Usage

```
crtpwr.2prop(alpha = 0.05, power = 0.8, m = NA, n = NA, cv = 0,
  p1 = NA, p2 = NA, icc = NA, pooled = FALSE, plinc = TRUE,
  tol = .Machine$double.eps^0.25)
```

Arguments

alpha	The level of significance of the test, the probability of a Type I error.
power	The power of the test, 1 minus the probability of a Type II error.
m	The number of clusters per condition. It must be greater than 1.
n	The mean of the cluster sizes.
cv	The coefficient of variation of the cluster sizes. When $cv = 0$, the clusters all have the same size.
p1	The expected proportion in the treatment group.
p2	The proportion in the control group.
icc	The intraclass correlation.
pooled	Logical indicating if pooled standard error should be used.
plinc	Logical indicating if p1 is expected to be greater than p2.
tol	Numerical tolerance used in root finding. The default provides at least four significant digits.

Value

The computed argument. #' @examples # Find the number of clusters per condition needed for a trial with $\alpha = .05$, # power = 0.8, 10 observations per cluster, no variation in cluster size, probability in condition 1 of .1 and condition 2 of .2, and $icc = 0.1$. `crtpwr.2prop(n=10 ,p1=.1, p2=.2, icc=.1)` # # The result, showing m of greater than 37, suggests 38 clusters per condition should be used.

Authors

Jonathan Moyer (<jon.moyer@gmail.com>)

crtpwr.2rate

Power calculations for simple cluster randomized trials, count outcome

Description

Compute the power of a simple cluster randomized trial with a count outcome, or determine parameters to obtain a target power.

Usage

```
crtplr.2rate(alpha = 0.05, power = 0.8, m = NA, py = NA, r1 = NA,
             r2 = NA, cvb = NA, tol = .Machine$double.eps^0.25)
```

Arguments

alpha	The level of significance of the test, the probability of a Type I error.
power	The power of the test, 1 minus the probability of a Type II error.
m	The number of clusters per condition. It must be greater than 1.
py	The number of person-years of observation per cluster.
r1	The expected mean event rate per unit time in the treatment group.
r2	The mean event rate per unit time in the control group.
cvb	The between-cluster coefficient of variation.
tol	Numerical tolerance used in root finding. The default provides at least four significant digits.

Value

The computed argument. # # @examples # Find the number of clusters per condition needed for a trial with alpha = .05, # power = 0.8, 10 person-years per cluster, rate in condition 1 of .1 and condition 2 of .2, and cvb = 0.1. crtplr.2rate(py=10, r1=.1, r2=.2, cvb=.1) ## The result, showing m of greater than 24, suggests 25 clusters per condition should be used.

Authors

Jonathan Moyer (<jon.moyer@gmail.com>)

expit

The expit and logit functions

Description

The expit and logit functions are useful shortcuts when using logistic regression models.

Usage

```
expit(x)
```

```
logit(p)
```

Arguments

x	a real number
p	a number between 0 and 1, i.e. a probability

Details

The logit function is defined as $\text{logit}(p) = \log(p)/\log(1-p)$ and can also be described as the log odds of a given probability. The expit is the inverse of the logit function and is defined as $\text{expit}(x) = \exp(x)/(1+\exp(x))$.

Value

A numeric value, possibly vectorized, depending on the input.

Author(s)

Nicholas G. Reich

Examples

```
expit(-2)
curve(expit(x), from=-5, to=5)

logit(.5)
curve(logit(x), from=0, to=1)
```

fixed.effect

Canned estimation functions for the power simulations.

Description

These functions are designed to be used by the power.sim.XXX functions as the functions which estimate the treatment effect. They fit simple fixed and random effects models and return the estimated treatment effect. These functions are not designed to be called directly by the user.

Usage

```
fixed.effect(dat, incl.period.effect, outcome.type, alpha)

fixed.effect.cluster.level(dat, incl.period.effect, outcome.type, alpha)

random.effect(dat, incl.period.effect, outcome.type, alpha)

weighted.crossover.cluster.level(dat, incl.period.effect, outcome.type, alpha)
```

Arguments

dat	observed data as a data.frame with columns named, "y", "trt" and "clust". "per" column is optional if period.var==0.
incl.period.effect	indicator of whether to include a period effect
outcome.type	one of "gaussian", "binomial", "poisson"
alpha	the type I error rate

Details

`random.effect()` relies on a call to `glmer()` from the `lme4` package. `fixed.effect()` relies on a call to `glm()`. `fixed.effect.cluster.level()` will save lots of time if you just want to run a cluster-level analysis and you have lots of observations. `weighted.crossover.cluster.level()` implements methods for fitting a weighted analysis on data from a crossover study (see Turner et al. 1997).

Value

A numeric vector with the following three elements, in order: [1] a point estimate for the treatment effect, [2] lower bound of (1-alpha) confidence interval, [3] lower bound of (1-alpha) confidence interval.

Author(s)

Nicholas G. Reich

See Also

[power.sim.normal](#), [power.sim.binomial](#), [power.sim.poisson](#)

hayes.power.poisson *An implementation of power calculations for cluster-randomized study based on the coefficient of variation.*

Description

This function calculates the power for a specified cluster-randomized study based on the methods described by Hayes et al (1999).

Usage

```
hayes.power.poisson(n.clusters, period.effect, btw.clust.var, at.risk.params,  
  cluster.size, effect.size, alpha = 0.05)
```

Arguments

<code>n.clusters</code>	number of clusters
<code>period.effect</code>	period effect, on the link scale. See details.
<code>btw.clust.var</code>	the between-cluster variance
<code>at.risk.params</code>	the expected at-risk time per individual in the study
<code>cluster.size</code>	the number of individuals in each cluster
<code>effect.size</code>	effect size, specified on the GLM link scale
<code>alpha</code>	desired type I error rate

Details

Calculates, for a specified study design, the power of that study to detect the specified effect size. The model is specified as a Poisson log-linear random effects model (`period.effect` and `btw.clust.var` are parameters from the model specified in Reich et al (2012)). Based on this model specification, the coefficient of variation between cluster-level outcomes is calculated using conditional expectation (see `mixed.eff.params()`) and then the formula from Hayes and Bennett (1999) is implemented.

Value

A numeric vector of length 1, containing the estimated power for the given study specifications.

References

Reich NG et al. PLoS ONE. Empirical Power and Sample Size Calculations for Cluster-Randomized and Cluster-Randomized Crossover Studies. 2012. <http://ow.ly/fEn39>

Hayes RJ and Bennett S. Int J Epi. Simple sample size calculation for cluster-randomized trials. 1999. <http://www.ncbi.nlm.nih.gov/pubmed/10342698>

See Also

[mixed.eff.params](#)

Examples

```
hayes.power.poisson(n.clusters=36, period.effect=log(.015), btw.clust.var=0,
  at.risk.params=20, cluster.size=20, effect.size=log(.7))
```

`mixed.eff.params`

Calculation of variance in Poisson mixed model setting.

Description

This function is designed to calculate the overall variance for cluster-level outcomes in a mixedeffect Poisson model. Conditional expectation calculations are implemented.

Usage

```
mixed.eff.params(pi0, btw.clust.var, Tk)
```

Arguments

<code>pi0</code>	the baseline cluster-level mean on the scale of the link function
<code>btw.clust.var</code>	the between-cluster-variance
<code>Tk</code>	the at-risk time for each cluster

Details

The `mixed.eff.params()` function is used by the `hayes.power.poisson()` function to compute the effective coefficient of variation, or k , for a particular study design.

Value

A numeric vector with the following three named elements, in order: ["expectation"] the overall mean of cluster-level outcomes, ["variance"] the overall variance of cluster-level outcomes, ["hayes.k"] the estimated coefficient of variation.

Author(s)

Nicholas G. Reich

References

Reich NG et al. PLoS ONE. Empirical Power and Sample Size Calculations for Cluster-Randomized and Cluster-Randomized Crossover Studies. 2012. <http://ow.ly/fEn39>

Hayes RJ and Bennett S. Int J Epi. Simple sample size calculation for cluster-randomized trials. 1999. <http://www.ncbi.nlm.nih.gov/pubmed/10342698>

See Also

[hayes.power.poisson](#)

Examples

```
mixed.eff.params(pi0=log(1), btw.clust.var=.5, Tk=100)
```

power.sim.normal

Power simulations for cluster-randomized crossover study designs.

Description

These functions run simulations to calculate power for a given cluster-randomized crossover study design. The user can specify a function which runs the desired method of analysis. The function `make.base.data()` is not meant to be called directly by users, but is used in the data generation algorithms employed by the other functions.

Usage

```
power.sim.normal(n.sim = 10, effect.size, alpha = 0.05, n.clusters,
  n.periods, cluster.size, btw.clust.var, indiv.var = NULL, ICC = NULL,
  period.effect, period.var, estimation.function, permute = FALSE,
  verbose = FALSE)
```

```
power.sim.binomial(n.sim = 10, effect.size, alpha = 0.05, n.clusters,
  n.periods, cluster.size, btw.clust.var, period.effect, period.var,
  estimation.function, permute = FALSE, verbose = FALSE)
```

```
power.sim.poisson(n.sim = 10, effect.size, alpha = 0.05, n.clusters,
  n.periods, cluster.size, btw.clust.var, period.effect, period.var,
  estimation.function, at.risk.params, permute = FALSE, verbose = FALSE)
```

```
make.base.data(n.obs, n.clusters, cluster.size, n.periods)
```

Arguments

<code>n.sim</code>	number of datasets to simulate
<code>effect.size</code>	effect size, specified on the GLM link scale
<code>alpha</code>	desired type I error rate
<code>n.clusters</code>	number of clusters
<code>n.periods</code>	number of periods of study
<code>cluster.size</code>	either a numeric vector length one or of length(<code>n.clusters</code>) defining the number of individuals in each cluster.
<code>btw.clust.var</code>	the between-cluster variance
<code>indiv.var</code>	for normal outcomes only, the individual level variance
<code>ICC</code>	for normal outcomes only, the ICC may be specified instead of <code>indiv.var</code>
<code>period.effect</code>	period effect, on the link scale. See details.
<code>period.var</code>	the period effects are drawn from a normal distribution centered at <code>period.effect</code> with variance <code>period.var</code> . If <code>period.var = 0</code> , period effect is assumed to be the same for all periods.
<code>estimation.function</code>	function to run the data analysis
<code>permute</code>	indicator of whether to run permutation inferences. Defaults to <code>FALSE</code> .
<code>verbose</code>	indicator of whether to print out updates as the simulator is running. Defaults to <code>FALSE</code>
<code>at.risk.params</code>	a numeric vector of length 1 or 2. See details.
<code>n.obs</code>	(for <code>make.base.data()</code> only) - the total number of observations in the data set

Details

Runs the power simulation.

The `period.effect` parameter needs to be specified on the "link function scale". Meaning that if the average baseline risk for a Poisson model is 4/1000, then the `period.effect` should be specified as `log(.004)`. Similarly, the baseline risk in a logistic model should be specified on the logit scale. The period effect can have length of 1, in which case it is treated as the average period effect across all periods, or it can have length equal to `n.periods`, in which case it is assumed that the investigator is specifying exact period effects s/he wishes to simulate.

For the Poisson simulations, at risk time is computed for each individual in the simulation. If `at.risk.time` is specified as a numeric vector of length 1, then the given number is the constant at risk time which every individual is assumed to have. If `length(at.risk.time)==2`, the values are taken as the mean and size parameters of a negative binomial distribution (used as `mu` and `size` in the `rnbinom()` function) from which an at-risk time is drawn for each individual. Specifically, the at risk times are drawn as `at.risk.time = 1 + rnbinom(1, size=at.risk.params[2], mu=at.risk.params[1])`.

Value

A list with the following components

results matrix with columns "dataset", "beta.est", "beta.cil", "beta.cih", "reject.null", "pval.permute"

power numeric, the estimated power

permute.power numeric, the estimated power using the permutation inference

sample.data a data frame containing the final simulated data set from the simulation run

Author(s)

Nicholas G. Reich

References

Turner RM, White IR, Croudace T, PIP Study Group. Analysis of cluster randomized cross-over trial data: a comparison of methods. *Stat Med.* 2007 Jan 30;26(2):274-89.

Examples

```
## Not run:
a <- power.sim.normal(n.sim=10, effect.size=5, alpha=.05, n.clusters=2, n.periods=2,
  cluster.size=20, btw.clust.var=5, ICC=1/20, period.effect=2,
  estimation.function=fixed.effect, verbose=TRUE, period.var=0)

b <- power.sim.binomial(n.sim=10, effect.size=log(.75), alpha=.05, n.clusters=20, n.periods=2,
  cluster.size=50, btw.clust.var=.2, period.effect=logit(.2),
  estimation.function=random.effect, verbose=TRUE, period.var=0)

c <- power.sim.poisson(n.sim=10, effect.size=log(.75), alpha=.05, n.clusters=100, n.periods=2,
  cluster.size=10, btw.clust.var=.4, period.effect=log(.2),
  estimation.function=random.effect, verbose=TRUE, period.var=0, at.risk.params=10)

## End(Not run)
```


Index

`clusterPower`, 2
`clusterPower-package (clusterPower)`, 2
`crtpr.2mean`, 2
`crtpr.2prop`, 3
`crtpr.2rate`, 4

`expit`, 5

`fixed.effect`, 6

`hayes.power.poisson`, 7, 9

`logit (expit)`, 5

`make.base.data (power.sim.normal)`, 9
`mixed.eff.params`, 8, 8

`power.sim.binomial`, 7
`power.sim.binomial (power.sim.normal)`, 9
`power.sim.normal`, 7, 9
`power.sim.poisson`, 7
`power.sim.poisson (power.sim.normal)`, 9

`random.effect (fixed.effect)`, 6

`weighted.crossover.cluster.level
(fixed.effect)`, 6