Package ‘clusterPower’

August 29, 2016

Version 0.5

Date 2013-12-13

Title Power calculations for cluster-randomized and cluster-randomized crossover trials

Author Nicholas G. Reich <nick@schoolph.umass.edu>, Daniel Obeng <dobeng@gmail.com>

Maintainer Nicholas G. Reich <nick@schoolph.umass.edu>

License GPL (>= 2)

Imports lme4 (>= 1.0)

Description This package enables researchers to calculate power for cluster-randomized crossover trials by employing a simulation-based approach. A particular study design is specified, with fixed sample sizes for all clusters and an assumed treatment effect, and the empirical power for that study design is calculated by simulating hypothetical datasets.

NeedsCompilation no

Repository CRAN

Date/Publication 2013-12-13 19:52:21

R topics documented:

  expit ................................................................. 2
  fixed.effect ......................................................... 3
  hayes.power.poisson ............................................. 4
  mixed.eff.params .................................................. 5
  power.sim.normal .................................................. 6

Index 9
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 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 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)
random.effect(dat, incl.period.effect, outcome.type, alpha)
fixed.effect.cluster.level(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 1 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**


**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 риск.params, cluster.size, effect.size, alpha=.05)

Arguments

n.clusters  number of clusters
period.effect  period effect, on the link scale. See details.
btw.clust.var  the between-cluster variance
at риск.params  the expected at-risk time per individual in the study
cluster.size  the number of individuals in each cluster
effect.size  effect size, specified on the GLM link scale
alpha  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.

Author(s)

Nicholas G. Reich

References

mixed.eff.params

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))

Description

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

Usage

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

Arguments

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

Details

mixed.eff.params() 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

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

```r
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)
```

```r
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

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

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


Examples

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

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

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

# End(Not run)
```
Index

+ Topic **logistic**
  expit, 2
+ Topic **permutation**
  power.sim.normal, 6
+ Topic **power**
  hayes.power.poisson, 4
  power.sim.normal, 6
+ Topic **random effects models**
  hayes.power.poisson, 4
  mixed.eff.params, 5
+ Topic **regression**
  fixed.effect, 3
+ Topic **simulation**
  power.sim.normal, 6

expit, 2

fixed.effect, 3

hayes.power.poisson, 4, 6

logit (expit), 2

make.base.data (power.sim.normal), 6
mixed.eff.params, 5, 5

power.sim.binomial, 3
power.sim.binomial (power.sim.normal), 6
power.sim.normal, 3, 6
power.sim.poisson, 3
power.sim.poisson (power.sim.normal), 6

random.effect (fixed.effect), 3

weighted.crossover.cluster.level
  (fixed.effect), 3