Package 'clusterPower'

January 28, 2021

Version 0.7.0

Date 2021-01-27

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

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

License GPL (>= 2)

Imports lme4 (>= 1.0),
progress (>= 1.1.2),
dplyr(>= 0.7.5),
tidyr (>= 0.8.1),
R.utils (>= 2.10.1),
car (>= 3.0-4),
lmerTest (>= 3.1-2),
nlme (>= 3.1-149),
foreach (>= 1.5.0),
shiny (>= 1.0.5),
methods (>= 4.0.0),
mathjaxr

Description Calculate power for cluster randomized trials (CRTs) including multi-arm trials, individually randomized group treatment trials (IGRTTs), stepped wedge trials (SWTs) and others using closed-form (analytic) solutions, and estimates power using Monte Carlo methods.

Suggests MASS (<= 7.3-53),
geepack (>= 1.2),
nloptr (>= 1.2.2.2),
optimx (>= 2020-4.2),
doParallel (>= 1.0.15),
shinyBS (>= 0.61),
tidyverse (>= 1.1.1),
DT (>= 0.2),
stringr (>= 1.2.0),
CRTSize,
data.table,
testthat (>= 3.0.0),
knitr,
rmarkdown,
shinycssloaders

RdMacros mathjaxr

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.1.1

Config/testthat/edition 3

R topics documented:

`binCalcICC` *BinCalcICC: calculate ICC values for data from CRTs with binary outcomes.*

Description

BinCalcICC: calculate ICC values for data from CRTs with binary outcomes.

Usage

```
binCalcICC(
  data = NULL,
 method = c("aov", "aovs", "keq", "kpr", "keqs", "kprs", "stab", "ub", "fc", "mak",
    "peq", "pgp", "ppr", "rm", "lin", "sim"),
  ci.type = c("aov", "wal", "fc", "peq", "rm"),
  alpha = 0.05,
  kappa = 0.45,
  nAGQ = 1,
  sim.min = 1,
  sim.max = 100,
  nsim = 1000
)
```

Arguments

`data`	A dataframe of the sort generated by `cps.binary()` or `cps.ma.binary()`; can be generate by using `all.sim.data = TRUE`.
`method`	The method to be used to compute ICC. A single or multiple methods can be used at a time. By default, all 16 methods will be used. See Details for more information.
`ci.type`	The type of confidence interval to be computed. By default all 5 types will be reported. See Details for more information.
`alpha`	The significance level to be used while computing the confidence interval. Default value is 0.05.
`kappa`	Value of Kappa to be used in computing Stabilized ICC when the method stab is chosen. Default value is 0.45.
`nAGQ`	An integer, as in glmer function of package lme4, denoting the number of points per axis for evaluating the adaptive Gauss-Hermite approximation to the log-likelihood. Used when the method lin is chosen. Default value is 1.
`sim.min`	Optional, integer. The number of the first simulation for which ICC will be calculated. Default is 1.
`sim.max`	Optional, integer. The number of the last simulation for which ICC will be calculated. Default is `nsim`.
`nsim`	Number of Monte Carlo replicates used in ICC computation method. `sim`. Default is 1000.

Value

A list with the following components:

estimate A dataframe containing the name of methods used and corresponding estimates of Intra-cluster Correlation coefficients

confidence.intervals A dataframe containing names of confidence interval types and corresponding estimated confidence intervals

Author(s)

Alexandria C. Sakrejda (<acbro0@umass.edu>) and Ken Kleinman (<ken.kleinman@gmail.com>)

Examples

```
## Not run:
bin.ma.rct.unbal <- cps.ma.binary(nsim = 3,
                            nsubjects = list(rep(200, times=15),
                            rep(150, times=15),
                            rep(170, times=15)),
                            narms = 3,
                            nclusters = 15,
                            probs = c(0.15, 0.23, 0.22),
                            sigma_b_sq = c(0.1, 0.1, 0.1),
                            alpha = 0.05, allSimData = TRUE,
                            seed = 123, cores="all")

binCalcICC(data = bin.ma.rct.unbal, nsim = 1000)

## End(Not run)

## Not run:
binary.sim = cps.binary(nsim = 100, nsubjects = 20,
  nclusters = 10, p1 = 0.8,
  p2 = 0.5, sigma_b_sq = 1,
  sigma_b_sq2 = 1.2, alpha = 0.05,
  method = 'glmm', allSimData = TRUE)

binCalcICC(data = binary.sim, nsim = 1000)

## End(Not run)
```

`clusterPower` *clusterPower: power analysis and sample size calculations for cluster randomized trials and related designs.*

Description

The clusterPower package is design for experiments with correlated, A.K.A. clustered observations. It contains many functions for calculating the power, sample size, and parameters necessary for achieving desired power and sample size from analytic equations. It also estimation of power via Monte Carlo methods via very flexible functions for a wide range of methods for which analytic methods or approximations may not exist.

Introduction

Most of the functions in this package are based on the generalized linear mixed model approach to the analysis of cluster randomized trials.

For example, for approximately normal outcomes in a simple parallel design, the data generating model is:

$$y_{ij}|b_i = \beta_0 + \beta_1 x_{ij} + b_i + e_{ij}$$

where i indexes cluster and j indexes subject within cluster. In this case we assume random effect b_i, the cluster means, are distributed Normal $(0, \sigma_b^2)$ and the residual error e_{ij} $N(0, \sigma^2)$. In this special case, we define the "total variance" as $\sigma_b^2 + \sigma^2$ and the Intracluster Correlation coefficient as $\sigma_b^2)/(\sigma_b^2 + \sigma^2)$. The ICC is useful in some special cases as a simplifying statistic, though it has no natural analogue in generalized linear mixed models when the distribution lacks a variance that is independent of the mean.

For example, for dichotomous outcomes, one convenient data generating model is:

$$logit(Pr(y_{ij}|b_i = \beta_0 + \beta_1 x_{ij} + b_i$$

`cpa.binary` *Analytic power calculations for parallel arm cluster-randomized trials with binary outcomes*

Description

Compute the power, number of clusters needed, number of subjects per cluster needed, or other key parameters for a parallel cluster randomized trial with a binary outcome.

Exactly one of `alpha`, `power`, `nclusters`, `nsubjects`, `CV`, `p1`, `p2`, and `ICC` must be passed as `NA`. Note that `alpha`, `power`, and `CV` have non-`NA` defaults, so if those are the parameters of interest they must be explicitly passed as `NA`.

Usage

```
cpa.binary(
  alpha = 0.05,
  power = NA,
  nclusters = NA,
  nsubjects = NA,
  CV = 0,
  p1 = NA,
  p2 = NA,
  ICC = 0.05,
  pooled = FALSE,
  p1inc = TRUE,
  tdist = 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.
nclusters	The number of clusters per condition. It must be greater than 1.
nsubjects	The mean of the cluster sizes.
CV	The coefficient of variation of the cluster sizes. When CV = 0, (default) the clusters all have the same size.
p1	The proportion with the outcome in one of the conditions, a numeric between 0-1.
p2	The proportion with the outcome in the other condition, a numeric between 0-1.
ICC	The intraclass correlation, a numeric between 0-1. (See Details, below.)
pooled	Logical indicating if pooled standard error should be used.
p1inc	Logical indicating if p1 is expected to be greater than p2. Only needed if p1 or p2 is NA.
tdist	If TRUE use t distribution with df equal to (nclusters - 2). Otherwise use the normal distribution. Default is TRUE.
tol	Numerical tolerance used in root finding. The default provides at least four significant digits.

Value

The computed value of the NA parameter (among alpha, power, nclusters, nsubjects, CV, p1, p2, and ICC) needed to satisfy the power and sample size equation.

Authors

Jonathan Moyer (<jon.moyer@gmail.com>), Ken Kleinman (<ken.kleinman@gmail.com>)

Notes

This function implements the approach of Donner and Klar (2000). An estimate for the intracluster correlation coefficient (ICC) is used to calculate a design effect that accounts for variance inflation due to clustering.

There are several ways in which estimates for the ICC for a binary outcome can be calculated, as described by Wu, Crespi, and Wong (2012). The user is advised to exercise caution in estimating, generating, and interpreting ICCs in this setting.

Unlike in the case of normal distributed outcomes (cpa.normal), the ICC refers neither to any natural parameter of a data generating model nor to any function of its parameters. For this reason we do not offer the user a option to input the variance between the clusters. If you prefer to use that input, we suggest using the cps.binary function.

This function was inspired by work from Stephane Champely (pwr.t.test) and Peter Dalgaard (power.t.test). As with those functions, '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. This generally means that no solution exists for which the omitted parameter and the supplied parameters fulfill the equation. In particular, the desired power may not be achievable with any number of subjects or clusters.

Testing details

This function has been verified against reference values from the NIH's GRT Sample Size Calculator, PASS11, `CRTSize::n4props`, and `clusterPower::cps.binary`.

References

Donner A, Klar N. Design and Analysis of Cluster Randomization Trials in Health Research. London; Arnold; 2000.

Wu S, Crespi CM, Wong WK. Comparison of Methods for Estimating Intraclass Correlation Coefficient for Binary Responses in Cancer Prevention Cluster Randomized Trials. Contemp Clin Trials. 2012; 33(5): 869-880. doi:10.1016/j.cct.2012.05.004 London: Arnold; 2000.

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.
## Not run:
cpa.binary(power = 0.08, nsubjects = 10, p1 = 0.1, p2 = 0.2, ICC = 0.1)

## End(Not run)
#
# The result, showing nclusters of greater than 37, suggests 38 clusters per
# condition should be used.

# Find the minimum detectable p2 > p1, given 38 clusters per condition, 10
# observations per cluster no variation in cluster size, ICC of 0.1, and
# probability of .1 in condition 2, with power of .8.
## Not run:
cpa.binary(power = 0.08, nsubjects = 10, nclusters = 38,
  p1 = 0.1, p2 = NA, ICC = 0.1, p1inc = FALSE)

## End(Not run)
# The result shows that p2 greater than 0.198922 can be detected with at
# least 80% power.
```

---

cpa.count — *Analytic power calculations for parallel arm cluster-randomized trials with count outcomes*

Description

Compute the power, number of clusters needed, number of subjects per cluster needed, or other key parameters for a simple parallel cluster randomized trial with a count outcome.

Exactly one of `alpha`, `power`, `nclusters`, `nsubjects`, `r1`, `r2`, and `CVB` must be passed as `NA`. Note that `alpha` and `power` have non-`NA` defaults, so if those are the parameters of interest they must be explicitly passed as `NA`.

Usage

```
cpa.count(
  alpha = 0.05,
  power = 0.8,
  nclusters = NA,
  nsubjects = NA,
  r1 = NA,
  r2 = NA,
  CVB = NA,
  r1inc = 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.
nclusters	The number of clusters per condition. It must be greater than 1
nsubjects	The number of units of person-time of observation per cluster
r1	The mean event rate per unit time in one of the conditions
r2	The mean event rate per unit time in the other condition
CVB	The between-cluster coefficient of variation
r1inc	Logical indicating if r1 is expected to be greater than r2. This is only important to specify if one of r1 or r2 is NA.
tol	Numerical tolerance used in root finding. The default provides at least four significant digits.

Value

The computed value of the NA parameter (among alpha, power, nclusters, nsubjects, r1, r2 and CVB) needed to satisfy the power and sample size equation.

Authors

Jonathan Moyer (<jon.moyer@gmail.com>), Ken Kleinman (<ken.kleinman@gmail.com>)

Note

This function implements the approach of Hayes and Bennet (1999). An estimate for the intracluster correlation coefficient (ICC) is used to calculate a design effect that accounts for variance inflation due to clustering.

The coefficient of variation CVB is the variance of the cluster rates divided by the mean of the cluster rates.

The CVB refers neither to any natural parameter of a data generating model nor to any function of its parameters. For this reason we do not offer the user a option to input the variance between the cluster means. If you prefer to use that input, we suggest using the cps.count function.

This function was inspired by work from Stephane Champely (pwr.t.test) and Peter Dalgaard (power.t.test). As with those functions, '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. This

generally means that no solution exists for which the omitted parameter and the supplied parameters fulfill the equation. In particular, the desired power may not be achievable with any number of subjects or clusters.

Testing details

This function has been verified against reference values from `CRTsize::n4incidence`, and `clusterPower::cps.count`.

References

Donner A, Klar N. Design and Analysis of Cluster Randomization Trials in Health Research. Chichester, UK; 2009.

Hayes JR, Bennett S. Simple sample size calculation for cluster-randomized trials. International Journal of Epidemiology 1999; 28:319-326

Hayes JR, Moulton LH. Cluster Randomized Trials. Boca Raton, FL: CRC Press; 2009.

Examples

```
# Find the number of clusters per condition needed for a trial with alpha = 0.05,
# power = 0.80, 10 person-years per cluster, rate in condition 1 of 0.10
# and condition 2 of 0.20, and CVB = 0.10.

cpa.count(nsubjects=10, r1=0.10, r2=0.20, CVB=0.10)

# The result, showimg nclusters of greater than 24, suggests 25 clusters per
# condition should be used.

# Find the largest CVB compatible with 80% power when there are 25 clusters, 10
# subject-units of time per cluster, and a rate of 0.1 and 0.2 in each condition.

cpa.count(nsubjects=10, nclusters= 25,r1=0.10, r2=0.20, CVB=NA)

# Results show that CVB as high as 0.107 can still yield power this high.
```

---

cpa.did.binary — *Power calculations for difference-in-difference cluster randomized trials, dichotomous outcome*

Description

Compute the power of a difference-in-difference cluster randomized trial design with a binary outcome, or determine parameters to obtain a target power.

Usage

```
cpa.did.binary(
  alpha = 0.05,
  power = 0.8,
  nclusters = NA,
  nsubjects = NA,
  p = NA,
```

```
  d = NA,
  ICC = NA,
  rho_c = NA,
  rho_s = 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.
nclusters	The number of clusters per condition. It must be greater than 1.
nsubjects	The mean of the cluster sizes.
p	The expected mean proportion at the post-test, averaged across both arms.
d	The expected absolute difference.
ICC	The intraclass correlation.
rho_c	The correlation between baseline and post-test outcomes at the cluster level. This value can be used in both cross-sectional and cohort designs. If this quantity is unknown, a value of 0 is a conservative estimate.
rho_s	The correlation between baseline and post-test outcomes at the subject level. This should be used for a cohort design or a mixture of cohort and cross-sectional designs. In a purely cross-sectional design (baseline subjects are completely different from post-test subjects), this value should be 0.
tol	Numerical tolerance used in root finding. The default provides at least four significant digits.

Details

Exactly one of alpha, power, nclusters, nsubjects, p, d, ICC, rho_c, and rho_s must be passed as NA. Note that alpha and power have non-NA defaults, so if those are the parameters of interest they must be explicitly passed as NA.

Value

The computed argument.

Authors

Jonathan Moyer (<jon.moyer@gmail.com>), Ken Kleinman (<ken.kleinman@gmail.com>)

Note

This function was inspired by work from Stephane Champely (pwr.t.test) and Peter Dalgaard (power.t.test). As with those functions, '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.

References

Murray D. Design and Analysis of Group-Randomized Trials. New York, NY: Oxford University Press; 1998.

Examples

```
# Find the number of clusters per condition needed for a trial with alpha = .05,
# power = 0.8, 50 observations per cluster, expected mean post-test proportion of .50,
# expected difference of .1, ICC = 0.05, cluster level correlation of 0.3, and subject level
# correlation of 0.7.
cpa.did.binary(nsubjects=50 ,p=.5, d=.1, ICC=.05, rho_c=.3, rho_s=.7)
#
# The result, showimg nclusters of greater than 32, suggests 33 clusters per
# condition should be used.
```

`cpa.did.count` *This help page is a stub. Equations do not yet exist for this type of analysis and outcome. Try* `cps.did.count()` *instead.*

Description

This help page is a stub. Equations do not yet exist for this type of analysis and outcome. Try `cps.did.count()` instead.

Usage

```
cpa.did.count(...)
```

Arguments

`...`	Any argument passed to the function.

Value

A helpful suggestion to use cps.did.count() instead.

Author(s)

Alexandria C. Sakrejda (<acbro0@umass.edu>

Ken Kleinman (<ken.kleinman@gmail.com>)

`cpa.did.normal` *Power calculations for difference-in-difference cluster randomized trials, continuous outcome*

Description

Compute the power of a difference-in-difference cluster randomized trial design with a continuous outcome, or determine parameters to obtain a target power.

Usage

```
cpa.did.normal(
  alpha = 0.05,
  power = 0.8,
  nclusters = NA,
  nsubjects = NA,
  d = NA,
  ICC = NA,
  rho_c = NA,
  rho_s = NA,
  vart = 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.
nclusters	The number of clusters per condition. It must be greater than 1.
nsubjects	The mean of the cluster sizes, or a vector of cluster sizes for one arm.
d	The difference in mean change between conditions (i.e. "difference-in-difference").
ICC	The intraclass correlation.
rho_c	The correlation between baseline and post-test outcomes at the cluster level. This value can be used in both cross-sectional and cohort designs. If this quantity is unknown, a value of 0 is a conservative estimate.
rho_s	The correlation between baseline and post-test outcomes at the subject level. This should be used for a cohort design or a mixture of cohort and cross-sectional designs. In a purely cross-sectional design (baseline subjects are completely different from post-test subjects), this value should be 0.
vart	The total variation of the outcome (the sum of within- and between-cluster variation).
tol	Numerical tolerance used in root finding. The default provides at least four significant digits.

Details

Exactly one of alpha, power, nclusters, nsubjects, d, ICC, rho_c, rho_s, and vart must be passed as NA. Note that alpha andpower have non-NA defaults, so if those are the parameters of interest they must be explicitly passed as NA.

If nsubjects is a vector the values, nclusters will be recalculated using the values in nsubjects.

Value

The computed argument.

Note

This function was inspired by work from Stephane Champely (pwr.t.test) and Peter Dalgaard (power.t.test). As with those functions, '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.

Authors

Jonathan Moyer (<jon.moyer@gmail.com>), Ken Kleinman (<ken.kleinman@gmail.com>)

References

Rutterford C, Copas A, Eldridge S. (2015) Methods for sample size determination in cluster randomized trials. Int J Epidemiol. 44(3):1051-1067.

Teerenstra S, Eldridge S, Graff M, de Hoop E, Borm, GF. (2012) A simple sample size formula for analysis of covariance in cluster randomized trials. Statist Med. 31:2169-2178

Examples

```
# Find the number of clusters per condition needed for a trial with alpha = 0.05,
# power = 0.80, nsubjects = 100, d = 0.50 units, ICC = 0.05, rho_c = 0.50, rho_s = 0.70,
# and vart = 1 unit.
cpa.did.normal(nsubjects = 100 , d = 0.5, ICC = 0.05, rho_c = 0.50, rho_s = 0.70, vart = 1)
#
# The result, nclusters = 4.683358, suggests 5 clusters per condition should be used.
```

---

cpa.irgtt.binary *Power calculations for individually randomized group treatment trials, binary outcome*

Description

Compute the power of an individually randomized group treatment trial (IRGTT) design with a binary outcome, or determine parameters to obtain a target power.

Usage

```
cpa.irgtt.binary(
  alpha = 0.05,
  power = 0.8,
  nclusters = NA,
  nsubjects = NA,
  ncontrols = NA,
  ICC = NA,
  p2 = NA,
  p1 = NA,
  decrease = 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.
nclusters	The number of clusters in the intervention arm.
nsubjects	The number of subjects in each cluster in the intervention arm.

ncontrols	The number of subjects in the control arm.
ICC	The intracluster correlation coefficient, the correlation in outcome measurements between two individuals from the same cluster in the intervention arm.
p2	The expected probability of the outcome in the intervention arm.
p1	The expected probability of the outcome in the control arm.
decrease	Whether or not the proportion in the intervention arm is expected to be less than the proportion in the control arm. If TRUE it is assumed p2 < p1, while FALSE implies p2 > p1.
tol	Numerical tolerance used in root finding. The default provides at least four significant digits.

Details

Exactly one of alpha, power, nclusters, nsubjects, ncontrols, ICC, p2, and p1 must be passed as NA. Note that alpha and power have non-NA defaults, so if those are the parameters of interest they must be explicitly passed as NA.

Value

The computed argument.

Note

This function was inspired by work from Stephane Champely (pwr.t.test) and Peter Dalgaard (power.t.test). As with those functions, '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.

Authors

Jonathan Moyer (<jon.moyer@gmail.com>), Ken Kleinman (<ken.kleinman@gmail.com>)

References

Moerbeek, M. and Wong, W. K. (2008) Sample size formulae for trials comparing group and individual treatments in a multilevel model. Statist. Med., 27:2850-2864. doi: 10.1002/sim.3115.

Examples

```
# Find the required number of subjects per intervention cluster an IRGTT with alpha = 0.05,
# power = 0.80, nclusters = 23, ncontrols = 146, ICC = 0.05, p2 = 0.397, and p1 = 0.243.

cpa.irgtt.binary(nclusters=23, ncontrols = 146,
  ICC = 0.05, p2 = 0.397, p1 = 0.243, decrease = FALSE)

#
# The result, nsubjects = 7.96624, suggests 8 subjects per cluster
# in the intervention arm should be recruited.
# This means that the total number of subjects in the
# study is nclusters * nsubjects + ncontrols = 23 * 8 + 146 = 330.
```

cpa.irgtt.count — *This help page is a stub. Equations do not yet exist for this type of analysis and outcome. Try* `cps.irgtt.count()` *instead.*

Description

This help page is a stub. Equations do not yet exist for this type of analysis and outcome. Try `cps.irgtt.count()` instead.

Usage

```
cpa.irgtt.count(...)
```

Arguments

`...`	Any argument passed to the function.

Value

A helpful suggestion to use cps.irgtt.count() instead.

Author(s)

Alexandria C. Sakrejda (<acbro0@umass.edu>

Ken Kleinman (<ken.kleinman@gmail.com>)

cpa.irgtt.normal — *Power calculations for individually randomized group treatment trials, continuous outcome*

Description

Compute the power of an individually randomized group treatment trial (IRGTT) design with a continuous outcome, or determine parameters to obtain a target power.

Usage

```
cpa.irgtt.normal(
  alpha = 0.05,
  power = 0.8,
  nclusters = NA,
  nsubjects = NA,
  ncontrols = NA,
  d = NA,
  varu = NA,
  varei = NA,
  varr = 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.
nclusters	The number of clusters in the intervention arm.
nsubjects	The number of subjects in each cluster in the intervention arm.
ncontrols	The number of subjects in the control arm.
d	The expected treatment effect.
varu	The variance of the cluster level random effect for clusters in the intervention arm.
varei	The variance of the subject level random error for individuals in the intervention arm.
varr	The variance of the subject level random error for individuals in the control arm.
tol	Numerical tolerance used in root finding. The default provides at least four significant digits.

Details

Exactly one of alpha, power, nclusters, nsubjects, ncontrols, d, varu, varei, and varr must be passed as NA. Note that alpha and power have non-NA defaults, so if those are the parameters of interest they must be explicitly passed as NA.

Value

The computed argument.

Note

This function was inspired by work from Stephane Champely (pwr.t.test) and Peter Dalgaard (power.t.test). As with those functions, '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.

Authors

Jonathan Moyer (<jon.moyer@gmail.com>), Ken Kleinman (<ken.kleinman@gmail.com>)

References

Moerbeek, M. and Wong, W. K. (2008) Sample size formulae for trials comparing group and individual treatments in a multilevel model. Statist. Med., 27:2850-2864. doi: 10.1002/sim.3115.

Examples

```
# Find the required number of control subjects for an IRGTT with alpha = 0.05, power = 0.80,
# nclusters = 10, nsubjects = 10, d = 0.5 units,
# varu = 0.1, varei = 0.9, varr = 1.
cpa.irgtt.normal(nclusters=10, nsubjects = 10,
  d = 0.5, varu = 0.1, varei = 0.9, varr = 1)
#
# The result, ncontrols = 77.81084, suggests 78 subjects in the control arm should be recruited.
# This means that the total number of subjects in the
# study is nclusters*nsubjects + ncontrols = 10*10 + 78 = 178.
```

`cpa.ma.binary` *This help page is a stub. Equations do not yet exist for this type of analysis and outcome. Try* `cps.did.binary()` *instead.*

Description

This help page is a stub. Equations do not yet exist for this type of analysis and outcome. Try `cps.did.binary()` instead.

Usage

```
cpa.ma.binary(...)
```

Arguments

`...`	Any argument passed to the function.

Value

A helpful suggestion to use cps.ma.binary() instead.

Author(s)

Alexandria C. Sakrejda (<acbro0@umass.edu>

Ken Kleinman (<ken.kleinman@gmail.com>)

`cpa.ma.count` *This help page is a stub. Equations do not yet exist for this type of analysis and outcome. Try* `cps.did.count()` *instead.*

Description

This help page is a stub. Equations do not yet exist for this type of analysis and outcome. Try `cps.did.count()` instead.

Usage

```
cpa.ma.count(...)
```

Arguments

`...`	Any argument passed to the function.

Value

A helpful suggestion to use cps.ma.count() instead.

Author(s)

Alexandria C. Sakrejda (<acbro0@umass.edu>

Ken Kleinman (<ken.kleinman@gmail.com>)

`cpa.ma.normal` *Power calculations for multi-arm cluster randomized trials, continuous outcome*

Description

Compute the power of the overall F-test for a multi-arm cluster randomized trial with a continuous outcome, or determine parameters to obtain a target power.

Usage

```
cpa.ma.normal(
  alpha = 0.05,
  power = 0.8,
  narms = NA,
  nclusters = NA,
  nsubjects = NA,
  vara = NA,
  varc = NA,
  vare = 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.
`narms`	The number of independent arms (conditions). It must be greater than 2.
`nclusters`	The number of clusters per arm. It must be greater than 1.
`nsubjects`	The cluster size.
`vara`	The between-arm variance.
`varc`	The between-cluster variance.
`vare`	The within-cluster variance.
`tol`	Numerical tolerance used in root finding. The default provides at least four significant digits.

Details

Exactly one of `alpha`, `power`, `narms`, `nclusters`, `nsubjects`, `vara`, `varc`, and `vare` must be passed as `NA`. Note that `alpha` and `power` have non-`NA` defaults, so if those are the parameters of interest they must be explicitly passed as `NA`.

Assuming a balanced design, the between-arm variance σ_a^2 (corresponding to the function argument `vara`) can be estimated using the formula:

$$\sigma_a^2 = \sum_{i=1}^{n_a} (\mu_i - \mu)^2 / (n_a - 1)$$

where n_a is the number of arms, μ_i is the estimate of the i-th arm mean, and μ is the estimate of the overall mean of the outcome. This variance can be computed in R using the `var` function and a

vector of arm means. For example, suppose the estimated means for a three-arm trial were 74, 80, and 86 Then the estimate of the between-arm variance could be computed with `var(c(74,80,86))`, yielding a value of 36.

Value

The computed argument.

Note

This function was inspired by work from Stephane Champely (pwr.t.test), Peter Dalgaard (power.t.test), and Claus Ekstrom (power.anova.test). As with those functions, '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.

Authors

Jonathan Moyer (<jon.moyer@gmail.com>), Ken Kleinman (<ken.kleinman@gmail.com>)

References

Murray DM. Design and Analysis of Group-Randomized Trials. New York, NY: Oxford University Press; 1998.

Examples

```
# Suppose we are planning a multi-arm trial composed of a control arm and
# two treatment arms. It is known that each arm will contain 5 clusters. We
# wish to know the minimum number of subjects per cluster necessary to
# attain 80% power at a 5% level of significance. A pilot study was used to
# determine estimates of the between-arm variance, the between-cluster
# variance, and the within-cluster variance. The observed means of each arm
# in the pilot study were 74, 80, and 86, so the  between-arm variance is 36.
# As discussed in the "Details" section above, this can be calculated using
# the command var(c(74,80,86)). The within-cluster and between-cluster
# standard deviations were observed to be 8 and 3, respectively. This means
# the within-cluster and between-cluster variances are 64 and 9, respectively.
# These values are entered into the function as follows:

cpa.ma.normal(narms=3,nclusters=5,vara=36,varc=9,vare=64)
#
# The result, showing nsubjects of greater than 20, suggests 21 subjects per
# cluster should be used.
```

`cpa.normal` *Analytic power calculations for parallel arm cluster-randomized trials with normal outcomes*

Description

Compute the power, number of clusters needed, number of subjects per cluster needed, or other key parameters for a simple parallel cluster randomized trial with a normal outcome.

Exactly one of `alpha`, `power`, `nclusters`, `nsubjects`, `CV`, and `d` must be passed as `NA`. Note that `alpha`, `power`, and `CV` have non-`NA` defaults, so if those are the parameters of interest they must be explicitly passed as `NA`. The user must supply sufficient variance parameters to produce values for both the ICC and the total variance by providing 2 of the following: `ICC`, `vart`, `sigma_b_sq`, or `sigma_sq`.

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

Usage

```
cpa.normal(
  alpha = 0.05,
  power = NA,
  nclusters = NA,
  nsubjects = NA,
  sigma_sq = NA,
  sigma_b_sq = NA,
  CV = 0,
  d = NA,
  ICC = NA,
  vart = 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. Defaults to NA.
`nclusters`	The number of clusters per condition. It must be greater than 1.
`nsubjects`	The mean of the cluster sizes, or a vector of cluster sizes for one arm. When nsubjects is a vector, CV and nclusters are calculated from nsubjects and user-entered CV and nclusters is ignored.
`sigma_sq`	Within-cluster (residual) variance, σ^2.
`sigma_b_sq`	Between-cluster variance σ_b^2.
`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 $\sigma_b^2/(\sigma_b^2 + \sigma^2)$. Accepts a numeric between 0-1.
`vart`	The total variation of the outcome (the sum of within- and between-cluster variation) $\sigma_b^2 + \sigma^2$.
`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"). Default is `"taylor"`.
`tol`	Numerical tolerance used in root finding. The default provides at least four significant digits.

Value

The computed value of the NA parameter (from among alpha, power, nclusters, nsubjects, CV, and d) needed to satisfy the power and sample size equation.

Note

This function was inspired by work from Stephane Champely (pwr.t.test) and Peter Dalgaard (power.t.test). As with those functions, '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. This generally means that no solution exists for which the omitted parameter and the supplied parameters fulfill the equation. In particular, the desired power may not be achievable with any number of subjects or clusters.

Testing details

This function has been verified against reference values from the NIH's GRT Sample Size Calculator, PASS11, CRTsize::n4means, and clusterPower::cps.normal.

Authors

Jonathan Moyer (<jon.moyer@gmail.com>), Alexandria C. Sakrejda (<acbro0@umass.edu>), and Ken Kleinman (<ken.kleinman@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:
## Not run:
cpa.normal(nsubjects = 10, d = 1, ICC = .1, vart = 5)

## End(Not run)
# The result, showing nclusters of greater than 15, suggests 16 clusters per
# condition should be used.

# Find the power achieved with 16 clusters, 10 subjects per cluster,
# difference between condition of 1 unit, ICC = .1, and total variance
# of 5 units:
## Not run:
cpa.normal(power = NA, nclusters = 16, nsubjects = 10, d = 1,
           sigma_b_sq = .5, vart = 5)

## End(Not run)
# The result shows the power is 0.801766.
```

```
# Find the power achieved when each trial arm has 5 clusters of
# sizes 100, 50, 25, 100, and 100. When a vector of cluster sizes
# is provided (as in this example), the "ncluster" argument is ignored.

## Not run:
cpa.normal(alpha = .05, power = NA, nsubjects = c(100, 50, 25, 100, 100),
           d = .2, ICC = .1, sigma_b_sq = .1)

## End(Not run)
# The result shows the power is 0.13315.

# Find the power achieved with 20 clusters per arm, where
# the cluster sizes vary but have a mean size of 100 and coefficient of variation of .5:
## Not run:
cpa.normal(alpha = .05, power = NA, nclusters = 20, nsubjects = 100, CV = .5,
           d = .2, ICC = .1, sigma_b_sq = .1)

## End(Not run)
# The result shows the power is 0.4559881.
```

`cpa.sw.binary` *Power simulations for cluster-randomized trials: Stepped Wedge Design, Binary Outcome*

Description

This function uses a modified Cox method to determine power for stepped wedge cluster-randomized controlled trials. Users can modify a variety of parameters to suit their desired experimental situation.

Usage

```
cpa.sw.binary(
  nclusters = NA,
  steps = NA,
  nsubjects = NA,
  alpha = 0.05,
  timeEffect = 0,
  ICC = NA,
  p0 = NA,
  p1 = NA,
  tol = 1e-05,
  GQ = 100,
  quiet = FALSE
)
```

Arguments

`nclusters`	Number of clusters; accepts non-negative integer scalar (required).
`steps`	Number of crossover steps; Accepts positive scalar indicating the total number of steps, NOT including the baseline (required).

nsubjects	Number of subjects per cluster; accepts a scalar. Equal cluster sizes are assumed (required).
alpha	Significance level (default=0.05).
timeEffect	Expected time effect over the entire study period (assumed to be linear across time steps); accepts numeric (required). Default = 0 (no time effects).
ICC	Intracluster correlation coefficient as defined by Hussey and Hughes (2007) for participants at first time step; accepts numeric (required).
p0	Estimated baseline effect; accepts numeric (required).
p1	Estimated treatment effect; accepts numeric (required).
tol	Machine tolerance. Accepts numeric. Default is 1e-5.
GQ	Number of quadriture points used in Gaussian Legendre integration; accepts a scalar. Default is 100.
quiet	Suppresses the progress bar; logical. Default is FALSE.

Details

The stepped wedge trial design is a type of cross-over design in which clusters change treatments in waves. Initially all the clusters receive the same standard treatment, and at the end of the trial all of the clusters will be receiving the treatment of interest. More than one cluster can change treatments in a wave, but the order in which clusters change treatments is randomly determined. The outcome of interest is assessed in each cluster during each wave.

Users must specify the number of subjects per cluster, number of clusters, the number of time steps, the baseline effect, the expected treatment effect, expected absolute difference between treatment arms, ICC, and time effect.

Value

The estimated power.

Note

Much of the FORTRAN code for this package was kindly provided by Dr. Zhou.

Author(s)

Alexandria C. Sakrejda (<acbro0@umass.edu>)

Ken Kleinman (<ken.kleinman@gmail.com>)

References

Zhou X, Liao X, Kunz LM, Normand ST, Wang M, Spiegelman D. A maximum likelihood approach to power calculations for stepped wedge designs of binary outcomes. Biostatistics. 2020 Jan 1;21(1):102-121. doi: 10.1093/biostatistics/kxy031

Hussey, MA AND Hughes, JP. (2007). Design and analysis of stepped wedge cluster randomized trials. Contemporary Clinical Trials 28, 182–191.

Examples

```
# Estimate power for a trial with 3 steps and 9 clusters at the
# initiation of the study. Those
# clusters have 14 subjects each with no time effects.
# We estimated arm outcome proportions of
# 0.2 (pre-treatment) and 0.31 (post-treatment) and intracluster
# correlation coefficient (ICC) of 0.05.
# The resulting power should be 0.7992842.

## Not run:
sw.bin <- cpa.sw.binary(nclusters = 9,
  steps = 3,
  nsubjects = 14,
  timeEffect = 0,
  ICC = 0.05,
  p1 = 0.31,
  p0 = 0.2)

## End(Not run)
```

---

cpa.sw.count *Power calculations for stepped-wedge trials with a count outcome.*

Description

This function uses the SWSamp package by Gianluca Baio for estimating power based on analytic formula of Hussey and Hughes (2007) where sample size calculations are based on an assumption of a normally-distributed outcome.

Usage

```
cpa.sw.count(
  lambda1,
  RR,
  nclusters,
  steps,
  nsubjects,
  ICC = 0.01,
  alpha = 0.05,
  which.var = "within",
  X = NULL,
  all.returned.objects = FALSE
)
```

Arguments

lambda1	Baseline rate for outcome of interest
RR	Estimated relative risk of the intervention
nclusters	Number of clusters

steps	Number of time steps. Baseline is assumed.
nsubjects	Average size of each cluster
ICC	Intra-class correlation coefficient (default = 0.01)
alpha	Significance level (default=0.05)
which.var	String character specifying which variance to report. Options are the default value 'within' or 'total'.
X	A design matrix indicating the time at which each of the clusters should switch to the intervention arm. Default is NULL and this matrix is automatically computed, but can it can be passed as a user-defined matrix with (nclusters) rows and (steps + 1) columns.
all.returned.objects	Logical. Default = FALSE, indicating that only the estimated power should be returned. When TRUE, all objects (listed below) are returned.

Value

power	The resulting power

When all.returned.objects = TRUE, returned items also include:

sigma.y	The estimated total (marginal) sd for the outcome
sigma.e	The estimated residual sd
sigma.a	The resulting cluster-level sd
setting	A list including the following values: - n.clusters = The number of clusters (nclusters) - n.time.points = The number of steps in the SW design (steps) - avg.cluster.size = The average cluster size (nsubjects) - design.matrix = The design matrix for the SWT under consideration

Author(s)

Alexandria C. Sakrejda (<acbro0@umass.edu>)

Ken Kleinman (<ken.kleinman@gmail.com>)

References

Baio, G; Copas, A; Ambler, G; Hargreaves, J; Beard, E; and Omar, RZ Sample size calculation for a stepped wedge trial. Trials, 16:354. Aug 2015.

Hussey M and Hughes J. Design and analysis of stepped wedge cluster randomized trials. Contemporary Clinical Trials. 28(2):182-91. Epub 2006 Jul 7. Feb 2007

Examples

```
cpa.sw.count(lambda1 = 1.75, RR = 0.9, nclusters = 21, steps = 6, nsubjects = 30, ICC = 0.01)
```

`cpa.sw.normal` *Power calculations for stepped wedge cluster randomized trials, continuous outcome*

Description

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

Exactly one of `alpha`, `power`, `nclusters`, `nsubjects`, `ntimes`, `d`, `ICC`, `rho_c`, `rho_s`, and `vart` must be passed as `NA`. Note that `alpha` and`power` have non-`NA` defaults, so if those are the parameters of interest they must be explicitly passed as `NA`.

The stepped wedge model assumed by Hooper et al (2016) is given below:

$$y_{itjk} = \mu + \beta_t + X_{it}\theta + c_{ij} + (ct)_{itj} + s_{ijk} + e_{itjk}$$

where y_{itjk} is the outcome for individual k in cluster j of arm i at time t. Fixed effects include the overall mean μ and effects for time β_t. The vector X_{it} is 1 if arm i at time t is undergoing the intervention, 0 otherwise. The terms c_{ij}, $(ct)_{itj}$, s_{ijk}, and e_{itjk} correspond to the time invariant cluster random effect, the time-varying cluster random effect, the time invariant subject random effect, and the time-varying subject random effect respectively. Random effects are assumed to be independent and Normally distributed with mean 0 and variances σ^2_C, σ^2_{CT}, σ^2_S, and σ^2_E, respectively.

The total variance of the outcome σ^2 is given by

$$\sigma^2 = \sigma^2_C + \sigma^2_{CT} + \sigma^2_S + \sigma^2_E$$

Let ρ, ρ, and ρ be the intracluster correlation, cluster autocorrelation, and subject autocorrelation, respectively. These parameters are given as follows:

$$\rho = \frac{\sigma^2_C + \sigma^2_{CT}}{\sigma^2_C + \sigma^2_{CT} + \sigma^2_S + \sigma^2_E}$$

$$\rho_C = \frac{\sigma^2_C}{\sigma^2_C + \sigma^2_{CT}}$$

$$\rho_S = \frac{\sigma^2_S}{\sigma^2_S + \sigma^2_E}$$

When $\rho_S = 0$ the design is considered to be a cross-sectional design, with new individuals observed at each time point. When $\rho_S > 0$ the design is a closed cohort, with repeated measurements on the same individuals at each time point.

Usage

```
cpa.sw.normal(
  alpha = 0.05,
  power = 0.8,
  nclusters = NA,
  nsubjects = NA,
  ntimes = NA,
  d = NA,
  ICC = NA,
  rho_c = NA,
  rho_s = NA,
  vart = 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.
nclusters	The number of clusters switching to the intervention condition at each time point.
nsubjects	The number of subjects in each cluster.
ntimes	The number of time points in the trial (not including baseline).
d	The expected treatment effect.
ICC	The intra-cluster correlation, the correlation in outcome measurements between two individuals from the same cluster.
rho_c	The cluster autocorrelation, the correlation between two population means from the same cluster at different times. This value can be used in both cross-sectional and cohort designs.
rho_s	The individual autocorrelation, the correlation between two outcome measurements in the same individual at different times. In a purely cross-sectional design (new subjects are obtained at each time point), this value should be 0. For a cohort design (the same subjects are observed at each time point), this value will be greater than 0.
vart	The total variation of the outcome (the sum of variances from cluster and individual level random effects).
tol	Numerical tolerance used in root finding. The default provides at least four significant digits.

Value

The computed argument.

Note

This function was inspired by work from Stephane Champely (pwr.t.test) and Peter Dalgaard (power.t.test). As with those functions, '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.

Authors

Jonathan Moyer (<jon.moyer@gmail.com>), Alexandria C. Sakrejda (<acbro0@gmail.com>), Ken Kleinman (<ken.kleinman@gmail.com>)

References

Baio G, Copas A, Ambler G, et al., 2015. Sample size calculation for a stepped wedge trial. Trials. 16:354.

Hooper, R., Teerenstra, S., Hoop, E., and Eldridge, S. (2016) Sample size calculation for stepped wedge and other longitudinal cluster randomised trials. Statist. Med., 35: 4718-4728. doi: 10.1002/sim.7028.

Hussey, M. and Hughes, J. (2007) Design and analysis of stepped wedge cluster randomized trials. Contemp. Clin. Trials, 28: 182-191. doi: 10.1016/j.cct.2006.05.007.

Examples

```
# Find the required number of clusters switching to intervention at each time point for a trial
# with alpha = 0.05, power = 0.80, nsubjects = 50, ntimes = 5, d = 1.5 units, ICC = 0.2,
# rho_c = 0.80, rho_s = 0, and vart = 16 square-units. Note that because rho_s = 0, this is a
# cross-sectional design.
cpa.sw.normal(nsubjects = 50, ntimes = 5, d = 1.5, ICC = 0.2, rho_c = 0.80, rho_s = 0, vart = 16)
#
# The result, nclusters = 1.288772, suggests 2 clusters switching per time point
# should be used. This means that the total number of clusters in the study is
# nclusters * ntimes = 2 * 5 = 10.
```

---

cps.binary *Power simulations for cluster-randomized trials: Parallel Designs, Binary Outcome*

Description

This function uses Monte Carlo methods (simulations) to estimate power for cluster-randomized trials. Users can modify a variety of parameters to suit the simulations to their desired experimental situation.

Users must specify the desired number of simulations, number of subjects per cluster, number of clusters per arm, and two of the following three parameters: expected probability of the outcome in one group, expected probability of the outcome in the second group, and expected difference in probabilities between groups. Default values are provided for significance level, analytic method, progress updates, and whether the simulated data sets are retained.

Usage

```
cps.binary(
  nsim = NULL,
  nsubjects = NULL,
  nclusters = NULL,
  p1 = NULL,
  p2 = NULL,
  sigma_b_sq = NULL,
  sigma_b_sq2 = NULL,
```

```
  alpha = 0.05,
  method = "glmm",
  quiet = FALSE,
  allSimData = FALSE,
  seed = NA,
  nofit = FALSE,
  poorFitOverride = FALSE,
  lowPowerOverride = FALSE,
  timelimitOverride = TRUE,
  irgtt = FALSE
)
```

Arguments

nsim	Number of datasets to simulate; accepts integer. Required.
nsubjects	Number of subjects per cluster; accepts either a scalar (implying equal cluster sizes for the two groups), a vector of length two (equal cluster sizes within arm), or a vector of length sum(nclusters) (unequal cluster sizes within arm). Required.
nclusters	Number of clusters per treatment group; accepts a single integer (if there are the same number of clusters in each arm) or a vector of 2 integers (if nsubjects differs between arms). If a vector of cluster sizes >2 is provided in nsubjects, sum(nclusters) must match the nsubjects vector length. Required.
p1	Expected probability of outcome in first group.
p2	Expected probability of outcome in second group.
sigma_b_sq	Between-cluster variance; if sigma_b_sq2 is not specified, between-cluster variances are assumed to be equal in the two arms. Accepts numeric. Required.
sigma_b_sq2	Between-cluster variance for clusters in second group. Only required if between-cluster variances differ between treatment arms.
alpha	Significance level; default = 0.05.
method	Data analysis method, either generalized linear mixed effects model (GLMM) or generalized estimating equations (GEE). Accepts c('glmm', 'gee'); default = 'glmm'. Required.
quiet	When set to FALSE, displays simulation progress and estimated completion time, default = TRUE.
allSimData	Option to output list of all simulated datasets; default = FALSE.
seed	Option to set the seed. Default is NA.
nofit	Option to skip model fitting and analysis and only return the simulated data. Default = FALSE.
poorFitOverride	Option to override stop() if more than 25% of fits fail to converge.
lowPowerOverride	Option to override stop() if the power is less than 0.5 after the first 50 simulations and every ten simulations thereafter. On function execution stop, the actual power is printed in the stop message. Default = FALSE. When TRUE, this check is ignored and the calculated power is returned regardless of value.
timelimitOverride	Logical. When FALSE, stops execution if the estimated completion time is more than 2 minutes. Defaults to TRUE.

`irgtt` Logical. Default = FALSE. Is the experimental design an individually randomized group treatment trial? For details, see ?cps.irgtt.binary.

Details

The data generating model for observation j in cluster i is:

$$y_{ij} \sim \texttt{Bernoulli}(\frac{e^{p_1+b_i}}{1+e^{p_1+b_i}})$$

for the first group or arm, where $b_i \sim N(0, \sigma_b^2)$, while for the second group,

$$y_{ij} \sim \texttt{Bernoulli}(\frac{e^{p_2+b_i}}{1+e^{p_2+b_i}})$$

where $b_i \sim N(0, \sigma_{b_2}^2)$; if $\sigma_{b_2}^2$ is not used, then the second group uses $b_i \sim N(0, \sigma_b^2)$.

All random terms are generated independent of one another.

Non-convergent models are not included in the calculation of exact confidence intervals.

Value

If `nofit = F`, a list with the following components:

- Character string indicating total number of simulations, simulation type, and number of convergent models
- Number of simulations
- Data frame with columns "Power" (estimated statistical power), "lower.95.ci" (lower 95 "upper.95.ci" (upper 95 "Alpha" (probability of committing a Type I error or rejecting a true null), "Beta" (probability of committing a Type II error or failing to reject a false null). Note that non-convergent models are returned for review, but not included in this calculation.
- Analytic method used for power estimation
- Significance level
- Vector containing user-defined cluster sizes
- Vector containing user-defined number of clusters
- Data frame reporting sigma_b_sq for each group
- Vector containing user-supplied outcome probability and estimated odds ratio
- Data frame containing three estimates of ICC
- Data frame with columns: "Estimate" (Estimate of treatment effect for a given simulation), "Std.err" (Standard error for treatment effect estimate), "Test.statistic" (z-value (for GLMM) or Wald statistic (for GEE)), "p.value", "converge" (Did simulated model converge?)
- If allSimData = TRUE, list of data frames, each containing: "y" (Simulated response value), "trt" (Indicator for treatment group), "clust" (Indicator for cluster)
- List of warning messages produced by non-convergent models; Includes model number for cross-referencing against `model.estimates`
- Logical vector reporting whether models converged.

If `nofit = T`, a data frame of the simulated data sets, containing:

- "arm" (Indicator for treatment arm)
- "cluster" (Indicator for cluster)
- "y1" ... "yn" (Simulated response value for each of the `nsim` data sets).

Testing details

This function has been verified against reference values from the NIH's GRT Sample Size Calculator, PASS11, `CRTsize::n4prop`, and `clusterPower::cpa.binary`.

Author(s)

Alexander R. Bogdan, Alexandria C. Sakrejda (<acbro0@umass.edu>), and Ken Kleinman (<ken.kleinman@gmail.com
#'

References

Elridge, S., Ukoumunne, O. & Carlin, J. The Intra-Cluster Correlation Coefficient in Cluster Randomized Trials: A Review of Definitions. International Statistical Review (2009), 77, 3, 378-394. doi: 10.1111/j.1751-5823.2009.00092.x

Snjiders, T. & Bosker, R. Multilevel Analysis: an Introduction to Basic and Advanced Multilevel Modelling. London, 1999: Sage.

Wu S, Crespi CM, Wong WK. Comparison of Methods for Estimating Intraclass Correlation Coefficient for Binary Responses in Cancer Prevention Cluster Randomized Trials. Contemp Clin Trials. 2012; 33(5): 869-880. doi:10.1016/j.cct.2012.05.004 London: Arnold; 2000.

See Also

An intracluster correlation coefficient (ICC) for binary outcome data is neither a natural parameter of the data generating model nor a function of its parameters. Several methods for calculation have been suggested (Wu, Crespi, and Wong, 2012). We provide several versions of ICCs for comparison. These can be accessed in the `bincalcICC()` function.

Examples

```
# Estimate power for a trial with 10 clusters in each arm, 20 subjects in
# each cluster, with a probability of 0.8 in the first arm and 0.5 in the
# second arm, with a sigma_b_sq = 1 in the first arm sigma_b_sq = 1.2 in
# the second arm.

## Not run:
binary.sim = cps.binary(nsim = 100, nsubjects = 20,
  nclusters = 10, p1 = 0.8,
  p2 = 0.5, sigma_b_sq = 1,
  sigma_b_sq2 = 1.2, alpha = 0.05,
  method = 'glmm', allSimData = FALSE)

## End(Not run)

# Estimate power for a trial just as above, except that in the first arm,
# the clusters have 10 subjects in 9 of the 10 clusters and 100 in the tenth
# cluster, while in the second arm all clusters have 20 subjects.

## Not run:
binary.sim2 = cps.binary(nsim = 100,
  nsubjects = c(c(rep(10,9),100), rep(20,10)),
  nclusters = 10, p1 = 0.8,
  p2 = 0.5, sigma_b_sq = 1,
  sigma_b_sq2 = 1.2, alpha = 0.05,
```

```
  method = 'gee', allSimData = FALSE)

## End(Not run)
```

---

cps.count *Simulation-based power estimation for cluster-randomized trials: Parallel Designs, Count Outcome*

Description

This function uses Monte Carlo methods (simulations) to estimate power for cluster-randomized trials with integer-valued outcomes. Users can modify a variety of parameters to suit the simulations to their desired experimental situation.

Users must specify the desired number of simulations, number of subjects per cluster, number of clusters per treatment arm, between-cluster variance, and two of the following three parameters: mean event rate per unit time in one group, the mean event rate per unit time in the second group, and/or the mean difference in event rates between groups. Default values are provided for significance level, analytic method, whether progress updates are displayed, and whether the simulated data sets are retained.

Note that if all units have the same observation time, you can use the mean count instead of the "mean event per unit time" in the preceding paragraph.

Usage

```
cps.count(
  nsim = NULL,
  nsubjects = NULL,
  nclusters = NULL,
  c1 = NULL,
  c2 = NULL,
  cDiff = NULL,
  sigma_b_sq = NULL,
  sigma_b_sq2 = NULL,
  family = "poisson",
  negBinomSize = 1,
  analysis = "poisson",
  method = "glmm",
  alpha = 0.05,
  quiet = FALSE,
  allSimData = FALSE,
  irgtt = FALSE,
  seed = NA,
  nofit = FALSE,
  poorFitOverride = FALSE,
  lowPowerOverride = FALSE,
  timelimitOverride = TRUE,
  optmethod = "Nelder_Mead"
)
```

Arguments

nsim	Number of datasets to simulate; accepts integer. Required.
nsubjects	Number of subjects per cluster; accepts either a scalar (implying equal cluster sizes for the two groups), a vector of length two (equal cluster sizes within arm), or a vector of length sum(nclusters) (unequal cluster sizes within arm). If a vector of > 2 is provided in nsubjects, sum(nclusters) must match the nsubjects vector length. Required.
nclusters	Number of clusters per treatment group; accepts a single integer (if there are the same number of clusters in each arm) or a vector of 2 integers (if there are not). Required.
	At least 2 of the following 3 arguments must be specified:
c1	The mean event rate per unit time in the first arm.
c2	The mean event rate per unit time in the second arm.
cDiff	Expected difference in mean event rates between groups, defined as cDiff = c1 -c2.
sigma_b_sq	Between-cluster variance; if sigma_b_sq2 is not specified, between-cluster variances are assumed to be equal in the two arms. Accepts numeric. Required.
sigma_b_sq2	Between-cluster variance for clusters in the second arm. Only required if between-cluster variances differ between treatment arms.
family	Distribution from which responses are simulated. Accepts Poisson ('poisson') or negative binomial ('neg.binom'); default = 'poisson'. Required.
negBinomSize	Only used when generating simulated data from the negative binomial (family = 'neg.binom'), this is the target for number of successful trials, or the dispersion parameter (the shape parameter of the gamma mixing distribution). Must be strictly positive but need not be integer. Defaults to 1.
analysis	Family used for data analysis; currently only applicable when method = 'glmm'. Accepts c('poisson', 'neg.binom'); default = 'poisson'. Required.
method	Data analysis method, either generalized linear mixed effects model (GLMM) or generalized estimating equations (GEE). Accepts c('glmm', 'gee'); default = 'glmm'. Required.
alpha	The level of significance of the test, the probability of a Type I error. Default = 0.05.
quiet	When set to FALSE, displays simulation progress and estimated completion time. Default = FALSE.
allSimData	Option to include a list of all simulated datasets in the output object. Default = FALSE.
irgtt	Logical. Default = FALSE. Is the experimental design an individually randomized group treatment trial? For details, see ?cps.irgtt.count.
seed	Option to set the seed. Default is NA.
nofit	Option to skip model fitting and analysis and instead return a dataframe with the simulated datasets. Default = FALSE.
poorFitOverride	Option to override stop() if more than 25% of fits fail to converge.
lowPowerOverride	Option to override stop() if the power is less than 0.5 after the first 50 simulations and every ten simulations thereafter. On function execution stop, the actual power is printed in the stop message. Default = FALSE. When TRUE, this check is ignored and the calculated power is returned regardless of value.

`timelimitOverride`
: Logical. When FALSE, stops execution if the estimated completion time is more than 2 minutes. Defaults to TRUE.

`optmethod`
: Option to fit with a different optimizer. Defaults to `Nelder_Mead`.

Details

If `family = 'poisson'`, the data generating model is:

$$y_{ij} \sim \texttt{Poisson}(e^{c_1+b_i})$$

for the first group or arm, where $b_i \sim N(0, \sigma_b^2)$, while for the second group,

$$y_{ij} \sim \texttt{Poisson}(e^{c_2+b_i})$$

where $b_i \sim N(0, \sigma_{b_2}^2)$; if $\sigma_{b_2}^2$ is not specified, then the second group uses $b_i \sim N(0, \sigma_b^2)$.

If `family = 'neg.bin'`, the data generating model, using the alternative parameterization of the negative binomial distribution detailed in `stats::rnbinom`, is:

$$y_{ij} \sim \texttt{NB}(\mu = e^{c_1+b_i}, \texttt{size} = 1)$$

for the first group or arm, where $b_i \sim N(0, \sigma_b^2)$, while for the second group,

$$y_{ij} \sim \texttt{NB}(\mu = e^{c_2+b_i}, \texttt{size} = 1)$$

where $b_i \sim N(0, \sigma_{b_2}^2)$; if $\sigma_{b_2}^2$ is not specified, then the second group uses $b_i \sim N(0, \sigma_b^2)$.

Non-convergent models are not included in the calculation of exact confidence intervals.

Value

If `nofit = F`, a list with the following components

- Character string indicating total number of simulations, distribution of simulated data, and regression family
- Number of simulations
- Data frame with columns "Power" (Estimated statistical power), "lower.95.ci" (Lower 95 "upper.95.ci" (Upper 95 Note that non-convergent models are returned for review, but not included in this calculation.
- Analytic method used for power estimation
- Data frame containing families for distribution and analysis of simulated data
- Significance level
- Vector containing user-defined cluster sizes
- Vector containing user-defined number of clusters
- Data frame reporting σ_b^2 for each group
- Vector containing expected events per unit time and risk ratios based on user inputs
- Data frame with columns: "Estimate" (Estimate of treatment effect for a given simulation), "Std.err" (Standard error for treatment effect estimate), "Test.statistic" (z-value (for GLMM) or Wald statistic (for GEE)), "p.value", "converge" (Did model converge for that set of simulated data?)

- If `allSimData = TRUE`, a list of data frames, each containing: "y" (Simulated response value), "trt" (Indicator for treatment arm), "clust" (Indicator for cluster)
- Logical vector reporting whether models converged.

If `nofit = T`, a data frame of the simulated data sets, containing:

- "arm" (Indicator for treatment arm)
- "cluster" (Indicator for cluster)
- "y1" ... "yn" (Simulated response value for each of the `nsim` data sets).

Testing details

This function has been verified, where possible, against reference values from PASS11, `CRTsize::n4incidence`, `clusterPower::cps.count`, and `clusterPower::cpa.count`.

Author(s)

Alexander R. Bogdan, Alexandria C. Sakrejda (<acbro0@umass.edu>), and Ken Kleinman (<ken.kleinman@gmail.com

Examples

```
# Estimate power for a trial with 10 clusters in each arm with 20 subjects each,
# with sigma_b_sq = 0.1 in both arms. We expect mean event rates per unit time of
# 20 and 30 in the first and second arms, respectively, and we use 100 simulated
# data sets analyzed by the GEE method.

## Not run:
count.sim = cps.count(nsim = 100, nsubjects = 20, nclusters = 10,
                      c1 = 20, c2 = 30, sigma_b_sq = 0.1,
                      family = 'poisson', analysis = 'poisson',
                      method = 'gee', alpha = 0.05, quiet = FALSE,
                      allSimData = FALSE, seed = 123)

## End(Not run)
# The resulting estimated power (if you set seed = 123) should be about 0.8.



# Estimate power for a trial with 10 clusters and 10 subjects per cluster in the
# first arm, 20 clusters and 20 subjects per cluster in the second, and
# sigma_b_sq = 0.1 in both arms. We expect mean event rates per unit time of
# 20 and 30 in the first and second arms, respectively, and we use 100 simulated
# data sets analyzed by the GLMM method.

## Not run:
count.sim = cps.count(nsim = 100, nsubjects = c(10,20), nclusters = c(10,10),
                      c1 = 20, c2 = 30, sigma_b_sq = 0.1,
                      family = 'poisson', analysis = 'poisson',
                      method = 'glmm', alpha = 0.05, quiet = FALSE,
                      allSimData = FALSE, seed = 123)

## End(Not run)
# The resulting estimated power (if you set seed = 123) should be about 0.85.
```

```
# Estimate power for a trial with 5 clusters in the first arm, those clusters having
# 4, 5, 6, 7, and 7 subjects each, and 10 clusters in the second arm, those
# clusters having 5 subjects each, with sigma_b_sq = 0.1 in the first arm and
# sigma_b_sq2 = .05 in the second arm. We expect mean event rates per unit time
# of 20 and 30 in the first and second arms, respectively, and we use 100 simulated
# data sets analyzed by the GLMM method.

## Not run:
count.sim = cps.count(nsim = 100, nsubjects = c(4, 5, 6, 7, 7, rep(5, times = 10)),
                      nclusters = c(5,10),
                      c1 = 20, c2 = 30,
                      sigma_b_sq = 0.1, sigma_b_sq2 = 0.05,
                      family = 'poisson', analysis = 'poisson',
                      method = 'glmm', alpha = 0.05, quiet = FALSE,
                      allSimData = FALSE, seed = 123)

## End(Not run)
# The resulting estimated power (if you set seed = 123) should be about 0.75.
```

`cps.did.binary` *Power simulations for cluster-randomized trials: Difference in Difference, Binary Outcome.*

Description

This function utilizes iterative simulations to determine approximate power for cluster-randomized controlled trials. Users can modify a variety of parameters to suit the simulations to their desired experimental situation.

Runs the power simulation for difference in difference RCTs with binary outcomes.

Users must specify the desired number of simulations, number of subjects per cluster, number of clusters per arm, pre-treatment between-cluster variance, and two of the following three terms: expected probability of outcome in arm 1, expected probability of outcome in arm 2, expected difference in probabilities between groups ; post-treatment between-cluster variance, significance level, analytic method, progress updates, and simulated data set output may also be specified.

The following equations are used to estimate intra-cluster correlation coefficients:

P_h:

$$ICC = \frac{\sigma_b}{\sigma_b + \pi^2/3}$$

P_c:

$$ICC = \frac{P(Y_{ij} = 1, Y_{ih} = 1) - \pi_j \pi_h}{\sqrt{\pi_j(1 - \pi_j)\pi_h(1 - \pi_h)}}$$

P_lmer:

$$ICC = \frac{\sigma_b}{\sigma_b + \sigma_w}$$

Usage

```
cps.did.binary(
  nsim = NULL,
  nsubjects = NULL,
  nclusters = NULL,
  p.diff = NULL,
  p1t0 = 0,
  p2t0 = NULL,
  p1t1 = NULL,
  p2t1 = NULL,
  or1 = NULL,
  or2 = NULL,
  or.diff = NULL,
  sigma_b_sq0 = NULL,
  sigma_b_sq1 = NULL,
  alpha = 0.05,
  method = "glmm",
  quiet = TRUE,
  allSimData = FALSE,
  poorFitOverride = FALSE,
  lowPowerOverride = FALSE,
  timelimitOverride = TRUE,
  seed = NA,
  nofit = FALSE
)
```

Arguments

nsim	Number of datasets to simulate; accepts integer (required).
nsubjects	Number of subjects per cluster; accepts integer (required).
nclusters	Number of clusters per arm; accepts integer (required).
p.diff	Optional if p1t1 and p2t0 are provided. Expected difference in outcome proportion between groups, defined as p.diff = (p1t1 - p1t0) - (p2t1 - p2t0). At least 2 of the following 3 arguments must be specified when using expected odds ratios:
p1t0	Required. Expected outcome proportion in arm 1 at baseline. Default is 0.
p2t0	Optional. Expected outcome proportion in arm 2 at baseline. If no quantity is provided, p2t0 = p1t0 is assumed.
p1t1	Optional. Expected outcome proportion in arm 1 at follow-up. If no quantity is provided, p1t1 = p1t0 is assumed.
p2t1	Required. Expected outcome proportion in arm 2 at follow-up.
or1	Expected odds ratio for outcome in arm 1
or2	Expected odds ratio for outcome in arm 2
or.diff	Expected difference in odds ratio for outcome between groups, defined as or.diff = or1 - or2.
sigma_b_sq0	Pre-treatment (time == 0) between-cluster variance; accepts numeric scalar (indicating equal between-cluster variances for both arms) or a vector of length 2 specifying treatment-specific between-cluster variances.

sigma_b_sq1	Post-treatment (time == 1) between-cluster variance; accepts numeric scalar (indicating equal between-cluster variances for both arms) or a vector of length 2 specifying treatment-specific between-cluster variances. If not provided by the user, sigma_b_sq1 = sigma_b_sq0.
alpha	Significance level. Default = 0.05
method	Analytical method, either Generalized Linear Mixed Effects Model (GLMM) or Generalized Estimating Equation (GEE). Accepts c('glmm', 'gee') (required); default = 'glmm'.
quiet	When set to FALSE, displays simulation start time and completion time. Default is TRUE.
allSimData	Option to output list of all simulated datasets. Default = FALSE.
poorFitOverride	Option to override stop() if more than 25% of fits fail to converge; default = FALSE.
lowPowerOverride	Option to override stop() if the power is less than 0.5 after the first 50 simulations and every ten simulations thereafter. On function execution stop, the actual power is printed in the stop message. Default = FALSE. When TRUE, this check is ignored and the calculated power is returned regardless of value.
timelimitOverride	Logical. When FALSE, stops execution if the estimated completion time is more than 2 minutes. Defaults to TRUE.
seed	Option to set the seed. Default is NA.
nofit	Option to skip model fitting and analysis and only return the simulated data. Default = FALSE.

Value

A list with the following components

- Character string indicating total number of simulations, simulation type, and number of convergent models
- Number of simulations
- Data frame with columns "Power" (Estimated statistical power), "lower.95.ci" (Lower 95 "upper.95.ci" (Upper 95
- Analytic method used for power estimation
- Significance level
- Vector containing user-defined cluster sizes
- Vector containing user-defined number of clusters
- Data frame reporting sigma_b_sq for each group at each time point
- Vector containing expected difference in probabilities based on user inputs
- Data frame with columns: "Period" (Pre/Post-treatment indicator), "Arm" (Arm indicator), "Value" (Mean response value)
- Data frame containing three estimates of ICC
- Data frame with columns: "Estimate" (Estimate of treatment effect for a given simulation), "Std.err" (Standard error for treatment effect estimate), "Test.statistic" (z-value (for GLMM) or Wald statistic (for GEE)), "p.value", "converge" (Did simulated model converge?), "sig.val" (Is p-value less than alpha?)

- If `allSimData = TRUE`, a list of data frames, each containing: "y" (Simulated response value), "trt" (Indicator for arm), "clust" (Indicator for cluster), "period" (Indicator for time point)
- List of warning messages produced by non-convergent models. Includes model number for cross-referencing against `model.estimates`

If `nofit = T`, a data frame of the simulated data sets, containing:

- "arm" (Indicator for treatment arm)
- "cluster" (Indicator for cluster)
- "y1" ... "yn" (Simulated response value for each of the `nsim` data sets).

Author(s)

Alexander R. Bogdan

Alexandria C. Sakrejda (<acbro0@umass.edu>

Ken Kleinman (<ken.kleinman@gmail.com>)

References

Snjiders, T. & Bosker, R. Multilevel Analysis: an Introduction to Basic and Advanced Multilevel Modelling. London, 1999: Sage.

Elridge, S., Ukoumunne, O. & Carlin, J. The Intra-Cluster Correlation Coefficient in Cluster Randomized Trials: A Review of Definitions. International Statistical Review (2009), 77, 3, 378-394. doi: 10.1111/j.1751-5823.2009.00092.x

Examples

```
# Estimate power for a trial with 10 clusters in both arms, those clusters having
# 20 subjects each, with sigma_b_sq0 = 1. We have estimated arm proportions of 0.2
# and 0.3 in the first and second arms, respectively, and we use
# 100 simulated data sets analyzed by the GLMM method. The resulting estimated power
# (if you set seed = 123) should be about 0.78.

## Not run:
did.binary.sim = cps.did.binary(nsim = 100, nsubjects = 20, nclusters = 10,
                                p1t0 = 0.1, p2t0 = 0.1,
                                p1t1 = 0.2, p2t1 = 0.45, sigma_b_sq0 = 1,
                                sigma_b_sq1 = 1, alpha = 0.05,
                                method = 'glmm', allSimData = FALSE, seed = 123)

## End(Not run)
```

`cps.did.count` *Power simulations for cluster-randomized trials: Difference in Difference, Count Outcome.*

Description

This function utilizes iterative simulations to determine approximate power for cluster-randomized controlled trials. Users can modify a variety of parameters to suit the simulations to their desired experimental situation.

Runs power simulations for difference in difference cluster randomized control trials using count outcomes

Users must specify the desired number of simulations, number of subjects per cluster, number of clusters per arm, between-cluster variance, two of the following: expected count in arm 1, expected count in arm 2, difference in counts between groups; significance level, analytic method, and whether or not progress updates should be displayed while the function is running.

Usage

```
cps.did.count(
  nsim = NULL,
  nsubjects = NULL,
  nclusters = NULL,
  c1t0 = 0,
  c2t0 = NULL,
  c1t1 = NULL,
  c2t1 = NULL,
  c.diff = NULL,
  sigma_b_sq0 = NULL,
  sigma_b_sq1 = 0,
  family = "poisson",
  analysis = "poisson",
  negBinomSize = 1,
  method = "glmm",
  alpha = 0.05,
  quiet = FALSE,
  allSimData = FALSE,
  poorFitOverride = FALSE,
  lowPowerOverride = FALSE,
  timelimitOverride = TRUE,
  seed = NA,
  nofit = FALSE
)
```

Arguments

nsim	Number of datasets to simulate; accepts integer (required).
nsubjects	Number of subjects per cluster; accepts integer (required).
nclusters	Number of clusters per arm; accepts integer (required). At least 2 of the following 3 arguments must be specified:
c1t0	Required. Expected outcome count in arm 1 at baseline. Default is 0.
c2t0	Optional. Expected outcome count in arm 2 at baseline. If no quantity is provided, c2t0 = c1t0 is assumed.
c1t1	Optional. Expected outcome count in arm 1 at follow-up. If no quantity is provided, c1t1 = c1t0 is assumed.
c2t1	Required. Expected outcome count in arm 2 at follow-up.

`c.diff` Optional if c1t1 and c2t0 are provided. Expected difference in outcome count between groups, defined as c.diff = (c1t1 - c1t0) - (c2t1 - c2t0).

`sigma_b_sq0` Pre-treatment (time == 0) between-cluster variance; accepts numeric scalar (indicating equal between-cluster variances for both arm) or a vector of length 2 specifying treatment-specific between-cluster variances

`sigma_b_sq1` Post-treatment (time == 1) between-cluster variance; accepts numeric scalar (indicating equal between-cluster variances for both arm) or a vector of length 2 specifying treatment-specific between-cluster variances. For data simulation, sigma_b_sq1 is added to sigma_b_sq0, such that if sigma_b_sq0 = 5 and sigma_b_sq1 = 2, the between-cluster variance at time == 1 equals 7. Default = 0.

`family` Distribution from which responses are simulated. Accepts Poisson ('poisson') or negative binomial ('neg.binom') (required); default = 'poisson'

`analysis` Family used for regression; currently only applicable for GLMM. Accepts c('poisson', 'neg.binom') (required); default = 'poisson'

`negBinomSize` Only used when generating simulated data from the negative binomial (family = 'neg.binom'), this is the target for number of successful trials, or the dispersion parameter (the shape parameter of the gamma mixing distribution). Must be strictly positive but need not be integer. Defaults to 1.

`method` Analytical method, either Generalized Linear Mixed Effects Model (GLMM) or Generalized Estimating Equation (GEE). Accepts c('glmm', 'gee') (required); default = 'glmm'

`alpha` Significance level for power estimation, accepts value between 0 - 1; default = 0.05

`quiet` When set to FALSE, displays simulation progress and estimated completion time. Default = FALSE.

`allSimData` Option to output list of all simulated datasets. Default = FALSE

`poorFitOverride` Option to override `stop()` if more than 25% of fits fail to converge; default = FALSE.

`lowPowerOverride` Option to override `stop()` if the power is less than 0.5 after the first 50 simulations and every ten simulations thereafter. On function execution stop, the actual power is printed in the stop message. Default = FALSE. When TRUE, this check is ignored and the calculated power is returned regardless of value.

`timelimitOverride` Logical. When FALSE, stops execution if the estimated completion time is more than 2 minutes. Defaults to TRUE.

`seed` Option to set the seed. Default is NA.

`nofit` Option to skip model fitting and analysis and only return the simulated data. Default = `FALSE`.

Value

A list with the following components:

- Character string indicating total number of simulations, distribution of simulated data, and regression family
- Number of simulations

- Data frame with columns 'Power' (Estimated statistical power), 'lower.95.ci' (Lower 95 'upper.95.ci' (Upper 95
- Analytic method used for power estimation
- Data frame containing families for distribution and analysis of simulated data
- Significance level
- Vector containing user-defined cluster sizes
- Vector containing user-defined number of clusters
- Data frame reporting between-cluster variances at each time point for each arm
- Vector containing expected counts and risk ratios based on user inputs
- Data frame with columns: 'Period' (Pre/Post-treatment indicator), 'Arm.2' (Arm indicator), 'Value' (Mean response value)
- Data frame with columns: 'Estimate' (Estimate of treatment effect for a given simulation), 'Std.Err' (Standard error for treatment effect estimate), 'Test.statistic' (z-value (for GLMM) or Wald statistic (for GEE)), 'p.value', 'converge' (Did simulated model converge?), 'sig.val' (Is p-value less than alpha?)
- If `allSimData = TRUE`, a list of data frames, each containing: 'y' (Simulated response value), 'trt' (Indicator for arm), 'clust' (Indicator for cluster), 'period' (Indicator for time point)

If `nofit = T`, a data frame of the simulated data sets, containing:

- "arm" (Indicator for treatment arm)
- "cluster" (Indicator for cluster)
- "y1" ... "yn" (Simulated response value for each of the `nsim` data sets).

Author(s)

Alexandria C. Sakrejda (<acbro0@umass.edu>

Alexander R. Bogdan

Ken Kleinman (<ken.kleinman@gmail.com>)

References

Snjiders, T. & Bosker, R. Multilevel Analysis: an Introduction to Basic and Advanced Multilevel Modelling. London, 1999: Sage.

Elridge, S., Ukoumunne, O. & Carlin, J. The Intra-Cluster Correlation Coefficient in Cluster Randomized Trials: A Review of Definitions. International Statistical Review (2009), 77, 3, 378-394. doi: 10.1111/j.1751-5823.2009.00092.x

Examples

```
# Estimate power for a trial with 7 clusters in both arms, those clusters having
# 9 subjects each, with sigma_b_sq0 = 0.1 in the first arm and 0.5 in the second arm.
# We have estimated arm counts of 5 and 3 in the first and second arms, respectively,
# and we use 100 simulated data sets analyzed by the GLMM method. The resulting
# estimated power (if you set seed = 123) should be 0.86.

## Not run:
did.count.sim = cps.did.count(nsim = 100, nsubjects = 9, nclusters = 7,
                              c1t0 = 5, c1t1 = 5, c2t0 = 5, c2t1 = 8,
```

```
                                sigma_b_sq0 = c(1, 0.5), sigma_b_sq1 = c(0.5, 0.8),
                                family = 'poisson', analysis = 'poisson',
                                method = 'glmm', seed = 123)

## End(Not run)
```

`cps.did.normal` *Power simulations for cluster-randomized trials: Difference in Difference Design, Continuous Outcome.*

Description

This set of functions utilize iterative simulations to determine approximate power for difference in difference cluster-randomized controlled trials. Users can modify a variety of parameters to suit the simulations to their desired experimental situation.

Runs the power simulation for difference in difference (DID) cluster-randomized controlled trial.

Users must specify the desired number of simulations, number of subjects per cluster, number of clusters per arm, expected absolute difference between arms, two of the following: ICC, within-cluster variance, or between-cluster variance; significance level, analytic method, progress updates, and simulated data set output may also be specified.

Usage

```
cps.did.normal(
  nsim = NULL,
  nsubjects = NULL,
  nclusters = NULL,
  delta_mu = 0,
  delta_mu2 = NULL,
  sigma_sq = NULL,
  sigma_b_sq0 = NULL,
  sigma_b_sq1 = 0,
  alpha = 0.05,
  method = "glmm",
  poorFitOverride = FALSE,
  lowPowerOverride = FALSE,
  timelimitOverride = TRUE,
  quiet = FALSE,
  allSimData = FALSE,
  seed = NA,
  nofit = FALSE
)
```

Arguments

`nsim`	Number of datasets to simulate; accepts integer (required).
`nsubjects`	Number of subjects per arm; accepts either a scalar (equal cluster sizes, both groups), a vector of length two (equal cluster sizes within groups), or a vector of length `sum(nclusters)` (unequal cluster sizes within groups) (required).

nclusters	Number of clusters per group; accepts integer scalar or vector of length 2 for unequal number of clusters per arm (required)
delta_mu	Default = 0. Reference arm expected change between from baseline to followup.
delta_mu2	Expected change in tretament arm at follow-up; accepts numeric (required).
sigma_sq	Within-cluster variance; accepts numeric scalar (indicating equal within-cluster variances for both arms at both time points) or vector of length 4 specifying within-cluster variance for each arm at each time point.
sigma_b_sq0	Pre-treatment (time == 0) between-cluster variance; accepts numeric scalar (indicating equal between-cluster variances for both arms) or a vector of length 2 specifying treatment-specific between-cluster variances
sigma_b_sq1	Post-treatment (time == 1) between-cluster variance; accepts numeric scalar (indicating equal between-cluster variances for both arm) or a vector of length 2 specifying treatment-specific between-cluster variances. For data simulation, sigma_b_sq1 is added to sigma_b_sq0, such that if sigma_b_sq0 = 5 and sigma_b_sq1 = 2, the between-cluster variance at time == 1 equals 7. Default = 0.
alpha	Significance level. Default = 0.05.
method	Analytical method, either Generalized Linear Mixed Effects Model (GLMM) or Generalized Estimating Equation (GEE). Accepts c('glmm', 'gee') (required); default = 'glmm'.
poorFitOverride	Option to override stop() if more than 25% of fits fail to converge; default = FALSE.
lowPowerOverride	Option to override stop() if the power is less than 0.5 after the first 50 simulations and every ten simulations thereafter. On function execution stop, the actual power is printed in the stop message. Default = FALSE. When TRUE, this check is ignored and the calculated power is returned regardless of value.
timelimitOverride	Logical. When FALSE, stops execution if the estimated completion time is more than 2 minutes. Defaults to TRUE.
quiet	When set to FALSE, displays simulation progress and estimated completion time; default is FALSE.
allSimData	Option to output list of all simulated datasets; default = FALSE.
seed	Option to set the seed. Default is NA.
nofit	Option to skip model fitting and analysis and only return the simulated data. Default = FALSE.

Value

A list with the following components:

- Character string indicating total number of simulations and simulation type
- Number of simulations
- Data frame with columns "Power" (Estimated statistical power), "lower.95.ci" (Lower 95 "upper.95.ci" (Upper 95
- Analytic method used for power estimation
- Significance level
- Vector containing user-defined cluster sizes

- Vector containing user-defined number of clusters
- Data frame reporting ICC, within & between cluster variances for both arms at each time point
- Vector containing expected difference between groups based on user inputs
- Data frame with columns: "Period" (Pre/Post-treatment indicator), "Arm.2" (arm indicator), "Value" (Mean response value)
- Data frame with columns: "Estimate" (Estimate of treatment effect for a given simulation), "Std.err" (Standard error for treatment effect estimate), "Test.statistic" (z-value (for GLMM) or Wald statistic (for GEE)), "p.value", "sig.val" (Is p-value less than alpha?)
- If `allSimData = TRUE`, a list of data frames, each containing: "y" (Simulated response value), "trt" (Indicator for arm), "clust" (Indicator for cluster), "period" (Indicator for time point)

If `nofit = T`, a data frame of the simulated data sets, containing:

- "arm" (Indicator for treatment arm)
- "cluster" (Indicator for cluster)
- "y1" ... "yn" (Simulated response value for each of the `nsim` data sets).

Author(s)

Alexander R. Bogdan

Alexandria C. Sakrejda (<acbro0@umass.edu>)

Ken Kleinman (<ken.kleinman@gmail.com>)

Examples

```
# Estimate power for a trial with 6 clusters in arm 1 and 6 clusters in arm 2,
# those clusters having 120 subjects each, with sigma_sq = 1. Estimated
# arm mean changes are 0 and 0.48 in the first and second arms, respectively, and we use
# 100 simulated data sets analyzed by the GLMM method. The resulting estimated
# power (for seed = 123) should be 0.81.

## Not run:
normal.did.rct = cps.did.normal(nsim = 100, nsubjects = 120, nclusters = 6,
                                delta_mu = 0, delta_mu2 = 0.48, sigma_sq = 1, alpha = 0.05,
                                  sigma_b_sq0 = 0.1, method = 'glmm', seed = 123)

## End(Not run)
```

`cps.irgtt.binary` *Simulation-based power estimation for binary outcome individually randomized group treatment trials.*

Description

This function utilizes iterative simulations to determine approximate power for individually randomized group treatment trials. Users can modify a variety of parameters to suit the simulations to their desired experimental situation.

Usage

```
cps.irgtt.binary(
  nsim = NULL,
  nsubjects = NULL,
  nclusters = NULL,
  p1 = NULL,
  p2 = NULL,
  sigma_b_sq = 0,
  sigma_b_sq2 = 0,
  alpha = 0.05,
  quiet = TRUE,
  allSimData = FALSE,
  poorFitOverride = FALSE,
  lowPowerOverride = FALSE,
  timelimitOverride = TRUE,
  nofit = FALSE,
  seed = NA
)
```

Arguments

nsim	Number of datasets to simulate; accepts integer (required).
nsubjects	Number of subjects per cluster in the clustered (arm 2) group; accepts integer (required).
nclusters	Number of clusters per arm; accepts integer (required).
p1	Expected probability of outcome in arm 1 (required)
p2	Expected probability of outcome in arm 2 (required)
sigma_b_sq	Between-cluster variance; defaults to 0. Accepts numeric.
sigma_b_sq2	Between-cluster variance for clusters in arm 2.
alpha	Significance level; default = 0.05
quiet	When set to FALSE, displays simulation progress and estimated completion time, default is TRUE.
allSimData	Option to output list of all simulated datasets; default = FALSE.
poorFitOverride	Option to override stop() if more than 25% of fits fail to converge; default = FALSE.
lowPowerOverride	Option to override stop() if the power is less than 0.5 after the first 50 simulations and every ten simulations thereafter. On function execution stop, the actual power is printed in the stop message. Default = FALSE. When TRUE, this check is ignored and the calculated power is returned regardless of value.
timelimitOverride	Logical. When FALSE, stops execution if the estimated completion time is more than 2 minutes. Defaults to TRUE.
nofit	Option to skip model fitting and analysis and return the simulated data. Defaults to FALSE.
seed	Option to set seed. Default is NA.

Details

Runs the power simulation for binary outcomes.

Users must specify the desired number of simulations, number of subjects per cluster, number of clusters per arm, two of the following three terms: expected probability of outcome in arm 1, expected probability of outcome in arm 2, expected difference in probabilities between groups; significance level, progress updates, and simulated data set output may also be specified.

Value

A list with the following components

- Character string indicating total number of simulations, simulation type, and number of convergent models
- Number of simulations
- Data frame with columns "Power" (Estimated statistical power), "lower.95.ci" (Lower 95 confidence interval bound)
- Analytic method used for power estimation
- Significance level
- Vector containing user-defined cluster sizes
- Vector containing user-defined number of clusters
- Data frame reporting sigma_b_sq for each group
- Vector containing expected difference in probabilities based on user inputs
- Data frame with columns: "Estimate" (Estimate of treatment effect for a given simulation), "Std.err" (Standard error for treatment effect estimate), "Test.statistic" (z-value (for GLMM) or Wald statistic (for GEE)), "p.value", "converge" (Did simulated model converge?), "sig.val" (Is p-value less than alpha?)
- If `allSimData = TRUE`, a list of data frames, each containing: "y" (Simulated response value), "trt" (Indicator for arm), "clust" (Indicator for cluster)
- List of warning messages produced by non-convergent models; Includes model number for cross-referencing against `model.estimates`

If `nofit = T`, a data frame of the simulated data sets, containing:

- "arm" (Indicator for treatment arm)
- "cluster" (Indicator for cluster)
- "y1" ... "yn" (Simulated response value for each of the `nsim` data sets).

Author(s)

Alexandria C. Sakrejda (<acbro0@umass.edu>)

Alexander R. Bogdan

Ken Kleinman (<ken.kleinman@gmail.com>)

References

Snjiders, T. & Bosker, R. Multilevel Analysis: an Introduction to Basic and Advanced Multilevel Modelling. London, 1999: Sage.

Eldridge, S., Ukoumunne, O. & Carlin, J. The Intra-Cluster Correlation Coefficient in Cluster Randomized Trials: A Review of Definitions. International Statistical Review (2009), 77, 3, 378-394. doi: 10.1111/j.1751-5823.2009.00092.x

Examples

```
## Not run:
irgtt.binary.sim <- cps.irgtt.binary(nsim = 100, nsubjects = c(150, 30),
                        nclusters = 10, p1 = 0.44,
                        p2 = 0.2, sigma_b_sq2 = 1, alpha = 0.05,
                        allSimData = FALSE)

## End(Not run)
```

`cps.irgtt.count` *Power simulations for cluster-randomized trials: Individually randomized group treatment trial designs, count outcome.*

Description

This function utilizes iterative simulations to determine approximate power for cluster-randomized controlled trials. Users can modify a variety of parameters to suit the simulations to their desired experimental situation.

Usage

```
cps.irgtt.count(
  nsim = NULL,
  nsubjects = NULL,
  nclusters = NULL,
  c1 = NULL,
  c2 = NULL,
  sigma_b_sq = 0,
  sigma_b_sq2 = 0,
  family = "poisson",
  analysis = "poisson",
  negBinomSize = 1,
  alpha = 0.05,
  quiet = FALSE,
  allSimData = FALSE,
  poorFitOverride = FALSE,
  lowPowerOverride = FALSE,
  timelimitOverride = TRUE,
  nofit = FALSE,
  seed = NA,
  optmethod = "Nelder_Mead"
)
```

Arguments

`nsim`	Number of datasets to simulate; accepts integer (required).
`nsubjects`	Number of subjects per cluster; accepts integer (required).
`nclusters`	Number of clusters in the arm; accepts integer (required). Arm 1 cluster size defaults to 1. At least 2 of the following 3 arguments must be specified:

c1	Expected outcome count in arm 1
c2	Expected outcome count in arm 2
sigma_b_sq	Between-cluster variance; defaults to 0. Accepts numeric. If between cluster variances differ between arms, the following must also be specified:
sigma_b_sq2	Between-cluster variance for clusters in arm 2
family	Distribution from which responses are simulated. Accepts Poisson ('poisson') or negative binomial ('neg.binom') (required); default = 'poisson'
analysis	Family used for regression; currently only applicable for GLMM. Accepts 'poisson' or 'neg.binom' (required); default = 'poisson'
negBinomSize	Only used when generating simulated data from the negative binomial (family = 'neg.binom'), this is the target for number of successful trials, or the dispersion parameter (the shape parameter of the gamma mixing distribution). Must be strictly positive but need not be integer. Defaults to 1.
alpha	Significance level. Default = 0.05.
quiet	When set to FALSE, displays simulation progress and estimated completion time. Default = FALSE.
allSimData	Option to output list of all simulated datasets. Default = FALSE.
poorFitOverride	Option to override stop() if more than 25% of fits fail to converge; default = FALSE.
lowPowerOverride	Option to override stop() if the power is less than 0.5 after the first 50 simulations and every ten simulations thereafter. On function execution stop, the actual power is printed in the stop message. Default = FALSE. When TRUE, this check is ignored and the calculated power is returned regardless of value.
timelimitOverride	Logical. When FALSE, stops execution if the estimated completion time is more than 2 minutes. Defaults to TRUE.
nofit	Option to skip model fitting and analysis and return the simulated data. Defaults to FALSE.
seed	Option to set seed. Default is NA.
optmethod	Option to fit with a different optimizer. Defaults to Nelder_Mead.

Details

Runs the power simulation for count outcomes.

Users must specify the desired number of simulations, number of subjects per cluster, number of clusters per arm, between-cluster variance, two of the following: expected count in arm 1 (no clusters), expected count in arm 2 (clustered arm), expected difference in counts between arms; significance level, and whether or not progress updates should be displayed while the function is running.

Value

A list with the following components

- Character string indicating total number of simulations, distribution of simulated data, and regression family
- Number of simulations

- Data frame with columns "Power" (Estimated statistical power), "lower.95.ci" (Lower 95 "upper.95.ci" (Upper 95
- Analytic method used for power estimation
- Data frame containing families for distribution and analysis of simulated data
- Significance level
- Vector containing user-defined cluster sizes
- Vector containing user-defined number of clusters
- Data frame reporting between-cluster variances for both arms
- Vector containing expected counts and risk ratios based on user inputs
- Data frame with columns: "Estimate" (Estimate of treatment effect for a given simulation), "Std.err" (Standard error for treatment effect estimate), "Test.statistic" (z-value (for GLMM) or Wald statistic (for GEE)), "p.value", "sig.val" (Is p-value less than alpha?)
- If `allSimData = TRUE`, a list of data frames, each containing: "y" (Simulated response value), "trt" (Indicator for arm), "clust" (Indicator for cluster)

If `nofit = T`, a data frame of the simulated data sets, containing:

- "arm" (Indicator for treatment arm)
- "cluster" (Indicator for cluster)
- "y1" ... "yn" (Simulated response value for each of the `nsim` data sets).

Author(s)

Alexandria C. Sakrejda (<acbro0@umass.edu>

Alexander R. Bogdan

Ken Kleinman (<ken.kleinman@gmail.com>)

Examples

```
## Not run:
irgtt.count.sim <- cps.irgtt.count(nsim = 100, nsubjects = c(200, 10), nclusters = 10,
                            c1 = 5, c2 = 7, sigma_b_sq2 = 0.1,
                            family = 'poisson', analysis = 'poisson',
                            alpha = 0.05, quiet = FALSE, allSimData = FALSE)

## End(Not run)
```

`cps.irgtt.normal` *Simulation-based power estimation for continuous outcome individually randomized group treatment trials.*

Description

This function uses iterative simulations to determine approximate power for individually randomized group treatment trials with a normally-distributed outcome of interest. Users can modify a variety of parameters to suit the simulations to their desired experimental situation. This function returns the summary power values for each arm.

Usage

```
cps.irgtt.normal(
  nsim = NA,
  nsubjects = NA,
  nclusters = NA,
  mu = NA,
  mu2 = NA,
  sigma_sq = NA,
  sigma_b_sq = 0,
  ICC2 = NA,
  sigma_sq2 = NA,
  sigma_b_sq2 = 0,
  alpha = 0.05,
  quiet = FALSE,
  allSimData = FALSE,
  nofit = FALSE,
  seed = NA,
  poorFitOverride = FALSE,
  lowPowerOverride = FALSE,
  timelimitOverride = TRUE
)
```

Arguments

nsim	Number of datasets to simulate; accepts integer (required).
nsubjects	Number of subjects per cluster in each arm; accepts either a scalar (equal cluster sizes, both groups), a vector of length two (equal cluster sizes within groups), or a vector of length sum(nclusters) (unequal cluster sizes within groups) (required).
nclusters	Number of clusters in the clustered group; accepts a scalar (required)
mu	Expected mean of arm 1; accepts numeric (required).
mu2	Expected mean of arm 2; accepts numeric (required).
sigma_sq	Within-cluster variance; accepts numeric
sigma_b_sq	Between-cluster variance for clusters in arm 2. Defaults to 0.
ICC2	Intra-cluster correlation coefficient for clusters in arm 2
sigma_sq2	Within-cluster variance for clusters in arm 2
sigma_b_sq2	Between-cluster variance for clusters in arm 2.
alpha	Significance level; default = 0.05.
quiet	When set to FALSE, displays simulation progress and estimated completion time; default is FALSE.
allSimData	Option to output list of all simulated datasets; default = FALSE.
nofit	Option to skip model fitting and analysis and return the simulated data. Defaults to FALSE.
	At least 2 of the following must be specified:
seed	Option to set seed. Default is NA.
poorFitOverride	Option to override stop() if more than 25% of fits fail to converge; default = FALSE.

`lowPowerOverride`
: Option to override `stop()` if the power is less than 0.5 after the first 50 simulations and every ten simulations thereafter. On function execution stop, the actual power is printed in the stop message. Default = FALSE. When TRUE, this check is ignored and the calculated power is returned regardless of value.

`timelimitOverride`
: Logical. When FALSE, stops execution if the estimated completion time is more than 2 minutes. Defaults to TRUE.

Details

Runs the power simulation.

Users must specify the desired number of simulations, number of subjects per cluster, number of clusters per arm, expected means for the arm 1 and arm 2 (respectively), two of the following: ICC, within-cluster variance, or between-cluster variance; significance level, progress updates, and simulated data set output may also be specified.

Value

A list with the following components:

- Character string indicating total number of simulations and simulation type
- Number of simulations
- Data frame with columns "Power" (Estimated statistical power), "lower.95.ci" (Lower 95 "upper.95.ci" (Upper 95
- Analytic method used for power estimation
- Significance level
- Vector containing user-defined cluster sizes
- Vector containing user-defined number of clusters in each treatment group
- Data frame reporting ICC for Treatment/Non-Treatment groups
- Vector containing expected group means based on user inputs
- Data frame with columns: "Estimate" (Estimate of treatment effect for a given simulation), "Std.err" (Standard error for treatment effect estimate), "Test.statistic" (z-value (for GLMM) or Wald statistic (for GEE)), "p.value", "sig.val" (Is p-value less than alpha?)
- If `allSimData = TRUE`, a list of data frames, each containing: "y" (Simulated response value), "trt" (Indicator for treatment group), "clust" (Indicator for cluster)

If `nofit = T`, a data frame of the simulated data sets, containing:

- "arm" (Indicator for treatment arm)
- "cluster" (Indicator for cluster)
- "y1" ... "yn" (Simulated response value for each of the `nsim` data sets).

Author(s)

Alexandria C. Sakrejda (<acbro0@umass.edu>)

Alexander R. Bogdan

Ken Kleinman (<ken.kleinman@gmail.com>)

Examples

```
## Not run:
irgtt.normal.sim <- cps.irgtt.normal(nsim = 100, nsubjects = c(100, 10),
                       nclusters = 8, mu = 1.1, mu2 = 1.5,
                        sigma_sq = 0.1, sigma_sq2 = 0.2,
                        sigma_b_sq2 = 0.1, alpha = 0.05,
                        quiet = FALSE, allSimData = TRUE, seed = 123)

## End(Not run)
```

`cps.ma.binary` *Simulation-based power estimation for binary outcome multi-arm cluster-randomized trials.*

Description

This function uses iterative simulations to determine approximate power for multi-arm cluster-randomized controlled trials with binary outcomes of interest. Users can modify a variety of parameters to suit the simulations to their desired experimental situation. This function validates the user's input and passes the necessary arguments to an internal function, which performs the simulations. This function returns the summary power values for each treatment arm.

Usage

```
cps.ma.binary(
  nsim = 1000,
  nsubjects = NULL,
  narms = NULL,
  nclusters = NULL,
  probs = NULL,
  sigma_b_sq = NULL,
  alpha = 0.05,
  quiet = FALSE,
  method = "glmm",
  multi_p_method = "bonferroni",
  allSimData = FALSE,
  seed = NULL,
  cores = NA,
  tdist = FALSE,
  poorFitOverride = FALSE,
  lowPowerOverride = FALSE,
  timelimitOverride = TRUE,
  nofit = FALSE,
  optmethod = "Nelder-Mead",
  return.all.models = FALSE
)
```

Arguments

`nsim`	Number of datasets to simulate; accepts integer (required).

`nsubjects` Number of subjects per cluster (required); accepts an integer if all are equal and `narms` and `nclusters` are provided. Alternately, the user can supply a list with one entry per arm if the cluster sizes are the same within the arm, or, if they are not the same within the arms, the user can supply a list of vectors where each vector represents an arm and each entry in the vector is the number of subjects per cluster.

`narms` Integer value representing the number of trial arms.

`nclusters` An integer or vector of integers representing the number of clusters in each arm.

`probs` Expected absolute treatment effect probabilities for each arm; accepts a scalar or a vector of length `narms` (required).

`sigma_b_sq` Between-cluster variance; accepts a vector of length `narms` (required).

`alpha` Significance level; default = 0.05.

`quiet` When set to FALSE, displays simulation progress and estimated completion time; default is FALSE.

`method` Analytical method, either Generalized Linear Mixed Effects Model (GLMM) or Generalized Estimating Equation (GEE). Accepts c('glmm', 'gee') (required); default = 'glmm'.

`multi_p_method` A string indicating the method to use for adjusting p-values for multiple comparisons. Choose one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none". The default is "bonferroni". See `?p.adjust` for additional details.

`allSimData` Option to output list of all simulated datasets; default = FALSE.

`seed` Option to set.seed. Default is NULL.

`cores` String ("all"), NA, or scalar value indicating the number of cores to be used for parallel computing. Default = NA (no parallel computing).

`tdist` Logical value indicating whether simulated data should be drawn from a t-distribution rather than the normal distribution. Default = FALSE.

`poorFitOverride`
Option to override `stop()` if more than 25% of fits fail to converge or power<0.5 after 50 iterations; default = FALSE.

`lowPowerOverride`
Option to override `stop()` if the power is less than 0.5 after the first 50 simulations and every ten simulations thereafter. On function execution stop, the actual power is printed in the stop message. Default = FALSE. When TRUE, this check is ignored and the calculated power is returned regardless of value.

`timelimitOverride`
Logical. When FALSE, stops execution if the estimated completion time is more than 2 minutes. Defaults to TRUE.

`nofit` Option to skip model fitting and analysis and return the simulated data. Defaults to `FALSE`.

`optmethod` User-specified optimizer method. Accepts `bobyqa`, `Nelder_Mead` (default), and optimizers wrapped in the `optimx` package.

`return.all.models`
Logical; Returns all of the fitted models and the simulated data. Defaults to FALSE.

Details

Users must specify the desired number of simulations, number of subjects per cluster, number of clusters per treatment arm, group probabilities, and the between-cluster variance. Significance level, analytic method, progress updates, poor/singular fit override, and whether or not to return the simulated data may also be specified. The internal function can be called directly by the user to return the fitted models rather than the power summaries (see `?cps.ma.normal.internal` for details).

Because the model for binary outcomes may be slower to fit than those for other distributions, this function may be slower than its normal or count-distributed counterparts. Users can spread the simulated data generation and model fitting tasks across multiple cores using the `cores` argument. Users should expect that parallel computing may make model fitting faster than using a single core for more complicated models. For simpler models, users may prefer to use single thread computing (`cores=1`), as the processes involved in allocating memory and copying data across cores also may take some time. For time-savings, this function stops execution early if estimated power < 0.5 or more than 25% of models produce a singular fit or non-convergence warning message, unless `poorFitOverride = TRUE`.

Non-convergent models are not included in the calculation of exact confidence intervals.

Value

A list with the following components:

power Data frame with columns "power" (Estimated statistical power), "lower.95.ci" (Lower 95% confidence interval bound), "upper.95.ci" (Upper 95% confidence interval bound).

model.estimates Data frame with columns corresponding to each arm with descriptive suffixes as follows: ".Estimate" (Estimate of treatment effect for a given simulation), "Std.Err" (Standard error for treatment effect estimate), ".zval" (for GLMM) | ".wald" (for GEE), and ".pval" (the p-value estimate).

overall.power Table of F-test (when method="glmm") or chi^2 (when method="gee") significance test results.

overall.power.summary Summary overall power of treatment model compared to the null model.

sim.data Produced when allSimData==TRUE. List of `nsim` data frames, each containing: "y" (simulated response value), "trt" (indicator for treatment group or arm), and "clust" (indicator for cluster).

model.fit.warning.percent Character string containing the percent of `nsim` in which the glmm fit was singular or failed to converge, produced only when method = "glmm" & allSimData = FALSE.

model.fit.warning.incidence Vector of length `nsim` denoting whether or not a simulation glmm fit triggered a "singular fit" or "non-convergence" error, produced only when method = "glmm" & allSimData=TRUE.

If `nofit = T`, a data frame of the simulated data sets, containing:

- "arm" (Indicator for treatment arm)
- "cluster" (Indicator for cluster)
- "y1" ... "yn" (Simulated response value for each of the `nsim` data sets).

Testing details

This function has been verified, where possible, against reference values from the NIH's GRT Sample Size Calculator, PASS11, `CRTsize::n4prop`, `clusterPower::cps.binary`, and `clusterPower::cpa.binary`.

Author(s)

Alexandria C. Sakrejda (<acbro0@umass.edu>), Alexander R. Bogdan, and Ken Kleinman (<ken.kleinman@gmail.com

Examples

```
## Not run:
bin.ma.rct.unbal <- cps.ma.binary(nsim = 12,
                            nsubjects = list(rep(20, times=15),
                            rep(15, times=15),
                            rep(17, times=15)),
                            narms = 3,
                            nclusters = 15,
                            probs = c(0.35, 0.43, 0.50),
                            sigma_b_sq = c(0.1, 0.1, 0.1),
                            alpha = 0.05, allSimData = TRUE,
                            seed = 123, cores="all")

bin.ma.rct.bal <- cps.ma.binary(nsim = 100, nsubjects = 50, narms=3,
                            nclusters=8,
                            probs = c(0.35, 0.4, 0.5),
                            sigma_b_sq = 0.1, alpha = 0.05,
                            quiet = FALSE, method = 'glmm',
                            allSimData = FALSE,
                            multi_p_method="none",
                            seed = 123, cores="all",
                            poorFitOverride = FALSE)

## End(Not run)
```

---

cps.ma.count *Simulation-based power estimation for cluster-randomized trials: Parallel Designs, Count Outcome with multiple arms*

Description

This function uses Monte Carlo methods (simulations) to estimate power for cluster-randomized trials for integer-valued outcomes with two or more trial conditions. Users can modify a variety of parameters to suit the simulations to their desired experimental situation.

Users must specify the desired number of simulations, number of subjects per cluster, number of clusters per treatment arm, between-cluster variance, and two of the following three parameters: mean event rate per unit time in one group, the mean event rate per unit time in the second group, and/or the mean difference in event rates between groups. Default values are provided for significance level, analytic method, progress updates, and whether the simulated data sets are retained.

Note that if all units have the same observation time, you can use the mean count instead of the "mean event per unit time" in the preceding paragraph.

Users must specify the desired number of simulations, number of subjects per cluster, number of clusters per treatment arm, group probabilities, and the between-cluster variance. Significance level, analytic method, whether progress updates are displayed, poor/singular fit override, and whether or not to return the simulated data may also be specified.

This user-friendly function calls an internal function; the internal function can be called directly by the user to return the fitted models rather than the power summaries (see ?cps.ma.count.internal for details).

Users can spread the simulated data generation and model fitting tasks across multiple cores using the `cores` argument. Users should expect that parallel computing may make model fitting faster than using a single core for more complicated models. For simpler models, users may prefer to use single thread computing (`cores=1`), as the processes involved in allocating memory and copying data across cores also may take some time. For time-savings, this function stops execution early if estimated power < 0.5 or more than 25% of models produce a singular fit or non-convergence warning message, unless `poorFitOverride = TRUE`.

Usage

```
cps.ma.count(
  nsim = 1000,
  nsubjects = NULL,
  narms = NULL,
  nclusters = NULL,
  counts = NULL,
  family = "poisson",
  analysis = "poisson",
  negBinomSize = 1,
  sigma_b_sq = NULL,
  alpha = 0.05,
  quiet = FALSE,
  method = "glmm",
  multi_p_method = "bonferroni",
  allSimData = FALSE,
  seed = NA,
  cores = NA,
  tdist = FALSE,
  poorFitOverride = FALSE,
  lowPowerOverride = FALSE,
  timelimitOverride = TRUE,
  return.all.models = FALSE,
  nofit = FALSE,
  opt = "NLOPT_LN_BOBYQA"
)
```

Arguments

`nsim`	Number of datasets to simulate; accepts integer. Required.
`nsubjects`	Number of subjects per cluster; accepts an integer (implying equal cluster sizes in all arms) if `narms` and `nclusters` are provided. Alternately, a list with one integer per arm (if the cluster sizes are the same within the arm), or a list of vectors where each vector represents an arm and each entry in the vector is the number of subjects per cluster (if the cluster sizes are not the same within the arms). Required.
`narms`	Number of trial arms; accepts integer. Required.
`nclusters`	Number of clusters per treatment group; accepts a single integer (if there are the same number of clusters in each arm) or a vector of integers representing the number of clusters in each arm (if nsubjects differs between arms). If a list of vectors of cluster sizes is provided in `nsubjects`, then the vector of cluster counts must match the length of the `nsubjects` vectors. Required.

counts — Mean event per unit time for each arm; accepts a scalar (if all arms have the same event rate) or a vector of length narms. Required.

family — Distribution from which responses are simulated. Accepts Poisson ('poisson') or negative binomial ('neg.binom'); default = 'poisson'. Required.

analysis — Family used for data analysis; currently only applicable when method = 'glmm'. Accepts c('poisson', 'neg.binom'); default = 'poisson'. Required.

negBinomSize — Only used when generating simulated data from the negative binomial (family = 'neg.binom'), this is the target for number of successful trials, or the dispersion parameter (the shape parameter of the gamma mixing distribution). Must be positive and defaults to 1. Required when family = 'neg.binom'.

sigma_b_sq — Between-cluster variance for each arm; accepts a scalar (if all arms have the same between-cluster variance) or a vector of length narms. Required.

alpha — The level of significance of the test, the probability of a Type I error. Default = 0.05.

quiet — When set to FALSE, displays simulation progress and estimated completion time. Default = FALSE.

method — Data analysis method, either generalized linear mixed effects model (GLMM) or generalized estimating equations (GEE). Accepts c('glmm','gee'); default = 'glmm'. Required.

multi_p_method — A string indicating the method to use for adjusting p-values for multiple comparisons. Choose one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none". The default is "bonferroni". See ?p.adjust for additional details.

allSimData — Option to include a list of all simulated datasets in the output object. Default = FALSE.

seed — Option to set the seed. Default is NULL.

cores — Number of cores to be used for parallel computing. Accepts a string ("all"), NA (no parallel computing), or scalar value indicating the number of CPUs to use. Default = NA.

tdist — Logical value indicating whether cluster-level random effects should be drawn from a t distribution rather than a normal distribution. Default = FALSE.

poorFitOverride — Option to override stop() if more than 25% of fits fail to converge; default = FALSE.

lowPowerOverride — Option to override stop() if the power is less than 0.5 after the first 50 simulations and every ten simulations thereafter. On stop, the power calculated from the completed simulations is printed in the stop message. Default = FALSE. When TRUE, this check is ignored and the calculated power is returned regardless of value.

timelimitOverride — Logical. When FALSE, stops execution if the estimated completion time is more than 2 minutes. Defaults to TRUE.

return.all.models — Logical; Returns all of the fitted models, the simulated data, the overall model comparisons, and the convergence report vector. This is equivalent to the output of cps.ma.count.internal(). See ?cps.ma.count.internal() for details.

nofit — Option to skip model fitting and analysis and return the simulated data. Defaults to FALSE.

opt — Optimizer for model fitting, from the package optimx or nloptwrap. Default is 'NLOPT_LN_BOBYQA'.

Details

If family = 'poisson', the data generating model is:

$$y_{ijk} \sim \text{Poisson}(e^{c_k + b_{jk}})$$

for observation i, in cluster j, in treatment arm k, where $b_{jk} \sim N(0, \sigma^2_{b_k})$.

If family = 'neg.bin', the data generating model, using the alternative parameterization of the negative binomial distribution detailed in stats::rnbinom, is:

$$y_{ijk} \sim \text{NB}(\mu = e^{c_k + b_{jk}}, \text{size} = 1)$$

for observation i, in cluster j, in treatment arm k, where $b_{jk} \sim N(0, \sigma^2_{b_k})$.

Non-convergent models are not included in the calculation of exact confidence intervals.

For complicated models, we recommend using parallel processing with the cores="all" argument. For simpler models, users may prefer to use single thread computing (cores=1), as the processes involved in allocating memory and copying data across cores also may take some time.

By default, this function stops execution early if estimated power < 0.5 or if more than 25% of models produce a singular fit or non-convergence warning. In some cases, users may want to ignore singularity warnings (see ?isSingular) by setting poorFitOverride = TRUE.

Value

A list with the following components:

power Data frame with columns "power" (Estimated statistical power), "lower.95.ci" (Lower 95% confidence interval bound), "upper.95.ci" (Upper 95% confidence interval bound).

model.estimates Data frame with columns corresponding to each arm with descriptive suffixes as follows: ".Estimate" (Estimate of treatment effect for a given simulation), "Std.Err" (Standard error for treatment effect estimate), ".zval" (for GLMM) | ".wald" (for GEE), and ".pval" (the p-value estimate).

overall.power Table of F-test (when method="glmm") or chi^2 (when method="gee") significance test results.

overall.power.summary Summary overall power of treatment model compared to the null model.

sim.data Produced when allSimData==TRUE. List of nsim data frames, each containing: "y" (simulated response value), "trt" (indicator for treatment group or arm), and "clust" (indicator for cluster).

model.fit.warning.percent Character string containing the percent of nsim in which the glmm fit was singular or failed to converge, produced only when method = "glmm" & allSimData = FALSE.

model.fit.warning.incidence Vector of length nsim denoting whether or not a simulation glmm fit triggered a "singular fit" or "non-convergence" error, produced only when method = "glmm" & allSimData=TRUE.

If nofit = T, a data frame of the simulated data sets, containing:

- "arm" (Indicator for treatment arm)
- "cluster" (Indicator for cluster)
- "y1" ... "yn" (Simulated response value for each of the nsim data sets).

Author(s)

Alexandria C. Sakrejda (<acbro0@umass.edu>)

Alexander R. Bogdan

Ken Kleinman (<ken.kleinman@gmail.com>)

Examples

```
# For a 3-arm trial with 4, 4, and 5 clusters in each arm, respectively,
# specify the number of subjects in each cluster with 3 vectors in a list,
# each vector representing a study arm. For each cluster, in no particular
# order, denote the number of subjects. In this example, the first arm
# contains 150, 200, 50, and 100 subjects in each of the 4 clusters. The second
# arm contains 50, 150, 210, and 100 subjects in each of 4 clusters, while
# the third arm contains 70, 200, 150, 50, and 100 subjects in each of 5
# clusters. The expected outcomes for each arm are 10, 55, and 65, and
# the sigma_b_sq values are 1, 1, and 2, respectively. Assuming
# seed = 123, the overall power for this trial should be 0.81.

## Not run:
nsubjects.example <- list(c(150, 200, 50, 100), c(50, 150, 210, 100),
                      c(70, 200, 150, 50, 100))
counts.example <- c(10, 55, 65)
sigma_b_sq.example <- c(1, 1, 2)

count.ma.rct.unbal <- cps.ma.count(nsim = 100,
                           narms = 3,
                           nsubjects = nsubjects.example,
                           counts = counts.example,
                           sigma_b_sq = sigma_b_sq.example,
                           alpha = 0.05, seed = 123)

## End(Not run)

# For a different trial with 4 arms, each arm has 4 clusters which
# each contain 100 subjects. Expected counts for each arm are 30
# for the first arm, 35 for the second, 70 for the third, and 40
# for the fourth. Similarly, sigma_b_sq for each arm are 1
# for the first arm, 1.2 for the second, 1 for the third, and 0.9
# for the fourth. Assuming seed = 123, the overall power for this
# trial should be 0.84

## Not run:
count.ma.rct.bal <- cps.ma.count(nsim = 10, nsubjects = 100, narms = 4,
                           nclusters = 25, counts = c(30, 35, 70, 40),
                           sigma_b_sq = c(1, 1.2, 1, 0.9), seed = 123)

## End(Not run)
```

`cps.ma.normal` *Simulation-based power estimation for continuous outcome multi-arm cluster-randomized trials.*

Description

This function uses iterative simulations to determine approximate power for multi-arm cluster-randomized controlled trials with a normally-distributed outcome of interest. Users can modify a variety of parameters to suit the simulations to their desired experimental situation. This function returns the summary power values for each arm.

Usage

```
cps.ma.normal(
  nsim = 1000,
  nsubjects = NULL,
  narms = NULL,
  nclusters = NULL,
  means = NULL,
  sigma_sq = NULL,
  sigma_b_sq = NULL,
  alpha = 0.05,
  quiet = FALSE,
  ICC = NULL,
  method = "glmm",
  multi_p_method = "bonferroni",
  allSimData = FALSE,
  seed = NA,
  cores = NULL,
  poorFitOverride = FALSE,
  lowPowerOverride = FALSE,
  tdist = FALSE,
  return.all.models = FALSE,
  optmethod = "nlminb",
  nofit = FALSE,
  timelimitOverride = TRUE
)
```

Arguments

nsim	Number of datasets to simulate; accepts integer (required).
nsubjects	Number of subjects per cluster (required); accepts an integer if all are equal and narms and nclusters are provided. Alternately, the user can supply a list with one entry per arm if the cluster sizes are the same within the arm, or, if they are not the same within the arms, the user can supply a list of vectors where each vector represents an arm and each entry in the vector is the number of subjects per cluster.
narms	Integer value representing the number of arms.
nclusters	An integer or vector of integers representing the number of clusters in each arm.
means	Expected absolute treatment effect for each arm; accepts a vector of length narms (required).
sigma_sq	Within-cluster variance; accepts a vector of length narms (required).
sigma_b_sq	Between-cluster variance; accepts a vector of length narms (required).
alpha	Significance level; default = 0.05.

quiet	When set to FALSE, displays simulation progress and estimated completion time; default is FALSE.
ICC	The intra-cluster correlation coefficient
method	Analytical method, either Generalized Linear Mixed Effects Model (GLMM) or Generalized Estimating Equation (GEE). Accepts c('glmm', 'gee') (required); default = 'glmm'.
multi_p_method	A string indicating the method to use for adjusting p-values for multiple comparisons. Choose one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", or "none" to leave p-values unadjusted. The default is "bonferroni". See ?p.adjust for additional details.
allSimData	Option to output list of all simulated datasets; default = FALSE.
seed	Option to set.seed. Default is NULL.
cores	a string ("all") or numeric value indicating the number of cores to be used for parallel computing.
poorFitOverride	Option to override stop() if more than 25% of fits fail to converge; default = FALSE.
lowPowerOverride	Option to override stop() if the power is less than 0.5 after the first 50 simulations and every ten simulations thereafter. On function execution stop, the actual power is printed in the stop message. Default = FALSE. When TRUE, this check is ignored and the calculated power is returned regardless of value.
tdist	Logical; use t-distribution instead of normal distribution for simulation values, default = FALSE.
return.all.models	Logical; Returns all of the fitted models, the simulated data, the overall model comparisons, and the convergence report vector. This is equivalent to the output of cps.ma.normal.internal(). See ?cps.ma.normal.internal() for details.
optmethod	Option to fit with a different optimizer. Default is 'nlminb', but some incompatible model types will trigger a list of compatible optimizer options.
nofit	Option to skip model fitting and analysis and return the simulated data. Defaults to FALSE.
timelimitOverride	Logical. When FALSE, stops execution if the estimated completion time is more than 2 minutes. Defaults to TRUE.

Details

Users must specify the desired number of simulations, the group/arm means, and two of the following: ICC, within-cluster variance, or between-cluster variance. Significance level, analytic method, progress updates, poor/singular fit override, and simulated data set output may also be specified. This function validates the user's input and passes the necessary arguments to an internal function, which performs the simulations. The internal function can be called directly by the user to return the fitted models rather than the power summaries (see ?cps.ma.normal.internal for details).

Users must also supply the number of arms, the subjects per cluster, and the number of clusters per arm. For a balanced design, users can provide these values with the arguments narms, nsubjects, and nclusters, respectively. For unbalanced designs, the user may provide a list of vectors with one vector per arm, with each vector containing the number of subjects per cluster. See the examples provided below for a demonstration of the various input options.

Non-convergent models are not included in the calculation of exact confidence intervals.

Value

A list with the following components:

power Data frame with columns "Power" (Estimated statistical power), "lower.95.ci" (Lower 95% confidence interval bound), "upper.95.ci" (Upper 95% confidence interval bound). Note that non-convergent models are returned for review, but not included in this calculation.

model.estimates Produced only when allSimData=TRUE, data frame with columns corresponding to each arm with the suffixes as follows: ".Estimate" (Estimate of treatment effect for a given simulation), "Std.Err" (Standard error for treatment effect estimate), ".tval" (for GLMM) | ".wald" (for GEE), ".pval"

overall.power.table Produced only when allSimData=TRUE, table of F-test (when method="glmm") or chi-squared (when method="gee") significance test results.

overall.power.summary Overall power of model compared to H0. Omits non-convergent models.

simulated.data If `allSimData = TRUE`, a list of `nsim` data frames, each containing: "y" (Simulated response value), "trt" (Indicator for arm), "clust" (Indicator for cluster).

model.fit.warning.percent Character string containing the percent of `nsim` in which the glmm fit was singular or failed to converge, produced only when method == "glmm" & allSimData==FALSE.

model.fit.warning.incidence Vector of length `nsim` denoting whether or not a simulation glmm fit triggered a "singular fit" or "non-convergence" error, produced only when method = "glmm" & allSimData=TRUE.

If `nofit = T`, a data frame of the simulated data sets, containing:

- "arm" (Indicator for treatment arm)
- "cluster" (Indicator for cluster)
- "y1" ... "yn" (Simulated response value for each of the `nsim` data sets).

Testing details

This function has been verified, where possible, against reference values from the NIH's GRT Sample Size Calculator, PASS11, `CRTsize::n4means`, `clusterPower::cps.normal`, and `clusterPower::cpa.normal`.

Author(s)

Alexandria C. Sakrejda (<acbro0@umass.edu>), Alexander R. Bogdan, and Ken Kleinman (<ken.kleinman@gmail.com

Examples

```
## Not run:
nsubjects.example <- list(c(20,20,20,25), c(15, 20, 20, 21), c(17, 20, 21))
means.example <- c(22, 21, 21.5)
sigma_sq.example <- c(1, 1, 0.9)
sigma_b_sq.example <- c(0.1, 0.15, 0.1)

multi.cps.normal.models <- cps.ma.normal(nsim = 100,
                              narms = 3,
                              nsubjects = nsubjects.example,
                              means = means.example,
                              sigma_sq = sigma_sq.example,
                              sigma_b_sq = sigma_b_sq.example,
                              alpha = 0.05,
```

```
                                quiet = FALSE, method = 'glmm',
                                seed = 123, cores = "all",
                                lowPowerOverride = FALSE,
                                poorFitOverride = FALSE,
                                optmethod = "nlm")

 multi.cps.normal <- cps.ma.normal(nsim = 100, narms = 3,
                                    nclusters = c(10,11,10), nsubjects = 25,
                                    means = c(1, 0.25, 1.75),
                                    sigma_sq = c(1.2, 1, 1.9),
                                    sigma_b_sq = c(0.5, 1, 0.75),
                                    quiet = FALSE, ICC=NULL, method = 'glmm',
                                    allSimData = FALSE, seed = 123,
                                    poorFitOverride = TRUE,
                                    cores = NULL,
                                    optmethod = "nlminb")

multi.cps.normal.simple <- cps.ma.normal(nsim = 100, narms = 3,
                                    nclusters = 5, nsubjects = 10,
                                    means = c(22.0, 22.5, 22.9),
                                    sigma_sq = 0.2,
                                    sigma_b_sq = 0.2, alpha = 0.05,
                                    quiet = FALSE, ICC=NULL, method = 'glmm',
                                    allSimData = FALSE, seed = 123,
                                    poorFitOverride = TRUE, cores="all",
                                    optmethod = "NLOPT_LN_NELDERMEAD")

## End(Not run)
```

`cps.normal` *Power simulations for cluster-randomized trials: Parallel Designs, Normal Outcome*

Description

This function uses Monte Carlo methods (simulations) to estimate power for parallel design cluster-randomized trials with normal outcomes. Users can modify a variety of parameters to suit the simulations to their desired experimental situation.

Users must specify the desired number of simulations, number of subjects per cluster, number of clusters per arm, expected means of the arms, and two of the following: ICC, within-cluster variance, or between-cluster variance. Defaults are provided for significance level, analytic method, progress updates, and whether the simulated data sets are retained.

Users have the option of specifying different variance parameters for each arm, different numbers of clusters for each treatment group, and different numbers of units within each cluster.

Non-convergent models are not included in the calculation of exact confidence intervals.

Usage

```
cps.normal(
  nsim = NA,
  nclusters = NA,
  nsubjects = NA,
```

```
  mu = 0,
  mu2 = NA,
  ICC = NA,
  sigma_sq = NA,
  sigma_b_sq = NA,
  ICC2 = NA,
  sigma_sq2 = NA,
  sigma_b_sq2 = NA,
  alpha = 0.05,
  method = "glmm",
  quiet = FALSE,
  allSimData = FALSE,
  seed = NA,
  poorFitOverride = FALSE,
  timelimitOverride = TRUE,
  lowPowerOverride = FALSE,
  irgtt = FALSE,
  nofit = FALSE
)
```

Arguments

nsim	Number of datasets to simulate; accepts integer. Required.
nclusters	Number of clusters per condition; accepts single integer (implying equal numbers of clusters in the two groups) or vector of length 2 (unequal number of clusters per arm). Required.
nsubjects	Number of subjects per cluster; accepts either a scalar (implying equal cluster sizes for the two groups), a vector of length two (equal cluster sizes within arm), or a vector of length sum(nclusters) (unequal cluster sizes within arm). Required.
mu	Mean in the first arm; accepts numeric, default 0. Required..
mu2	Mean in the second arm; accepts numeric. Required.
	At least 2 of the following must be specified:
ICC	Intra-cluster correlation coefficient; accepts a value between 0 and 1.
sigma_sq	Within-cluster variance; accepts numeric.
sigma_b_sq	Between-cluster variance; accepts numeric.
	The defaults for the following are all NA, implying equal variance parameters for the two groups. If one of the following is given, variance parameters differ between treatment groups, and at least 2 of the following must be specified:
ICC2	Intra-cluster correlation coefficient for clusters in the second arm.
sigma_sq2	Within-cluster variance for clusters in the second arm.
sigma_b_sq2	Between-cluster variance for clusters in the second arm.
	Optional parameters:
alpha	Significance level; default = 0.05.
method	Analytical method, either Generalized Linear Mixed Effects Model (GLMM, default) or Generalized Estimating Equation (GEE). Accepts c('glmm', 'gee').
quiet	When set to FALSE, displays simulation progress and estimated completion time; default is FALSE.

`allSimData` Option to include a list of all simulated datasets in the output object. Default = `FALSE`.

`seed` Option to set the seed. Default, NA, selects a seed based on the system clock.

`poorFitOverride`
Option to override `stop()` if more than 25% of fits fail to converge.

`timelimitOverride`
Logical. When FALSE, stops execution if the estimated completion time is more than 2 minutes. Defaults to TRUE.

`lowPowerOverride`
Option to override `stop()` if the power is less than 0.5 after the first 50 simulations and every ten simulations thereafter. On function execution stop, the actual power is printed in the stop message. Default = FALSE. When TRUE, this check is ignored and the calculated power is returned regardless of value.

`irgtt` Logical. Default = FALSE. Is the experimental design an individually randomized group treatment trial? For details, see ?cps.irgtt.normal.

`nofit` Option to skip model fitting and analysis and instead return a dataframe with the simulated datasets. Default = `FALSE`.

Details

The data generating model for observation i in cluster j is:

$$y_{ij} \sim N(\mu + b_i, \sigma^2)$$

for the first group or arm, where $b_i \sim N(0, \sigma_b^2)$, while for the second group,

$$y_{ij} \sim N(\mu_2 + b_i, \sigma_2^2)$$

where $b_i \sim N(0, \sigma_{b_2}^2)$; if none of $\sigma_2^2, \sigma_{b_2}^2$ or `ICC2` are used, then the second group uses $b_i \sim N(0, \sigma_b^2)$ and $y_{ij} \sim N(\mu_2 + b_i, \sigma^2)$.

All random terms are generated independent of one another.

For calls without $\sigma_2^2, \sigma_{b_2}^2$ or `ICC2`, and using `method="glmm"` the fitted model is:

$$y_{ij}|b_i = \mu + \beta_1 x_{ij} + b_i + e_{ij}$$

with $\beta_1 = \mu_2 - \mu$, treatment group indicator $x_{ij} = 0$ for the first group, with $b_i \sim N(0, \sigma_b^2)$ and $e_{ij} \sim N(0, \sigma^2)$. In this case, both the random effects distribution and the residual distribution are the same for both conditions.

Otherwise, for `method="glmm"` the fitted model is:

$$y_{ij}|b_i = \mu + \beta_1 x_{ij} + b_i I(x_{ij} = 0) + e_{ij} I(x_{ij} = 0) + g_i I(x_{ij} = 1) + f_{ij} I(x_{ij} = 1)$$

with β_1, x_{ij}, b_i, and e_{ij} as above, with $g_i \sim N(0, \sigma_{b_2}^2)$ and $f \sim N(0, \sigma_2^2)$, the random effects and residual distribution in the second group.

Value

If `nofit = F`, a list with the following components:

- Character string indicating total number of simulations and simulation type
- Number of simulations

- Data frame with columns "Power" (Estimated statistical power), "lower.95.ci" (Lower 95% confidence interval bound), "upper.95.ci" (Upper 95% confidence interval bound), "Alpha" (Probability of committing a type I or α error or rejecting a true null), "Beta" (Probability of committing a type II error or failing to reject a false null). Note that non-convergent models are returned for review, but not included in this calculation.
- Analytic method used for power estimation
- Significance level
- Vector containing user-defined cluster sizes
- Vector containing user-defined number of clusters in each arm
- Data frame reporting ICC, variance parameters, and means for each arm
- Vector containing expected group means based on user inputs
- Data frame with columns: "Estimate" (Estimate of treatment effect for a given simulation), "Std.err" (Standard error for treatment effect estimate), "Test.statistic" (z-value (for GLMM) or Wald statistic (for GEE)), "p.value", "converge", (Did the model converge?)
- If `allSimData = TRUE`, a list of data frames, each containing: "y" (Simulated response value), "trt" (Indicator for arm), "clust" (Indicator for cluster)

If `nofit = T`, a data frame of the simulated data sets, containing:

- "arm" (Indicator for treatment arm)
- "clust" (Indicator for cluster)
- "y1" ... "yn" (Simulated response value for each of the `nsim` data sets).

Testing details

This function has been verified, where possible, against reference values from the NIH's GRT Sample Size Calculator, PASS11, `CRTsize::n4means`, and `clusterPower::cpa.normal`.

Author(s)

Alexander R. Bogdan, Alexandria C. Sakrejda (<acbro0@umass.edu>), and Ken Kleinman (<ken.kleinman@gmail.com

Examples

```
# Estimate power for a trial with 10 clusters in each arm and 25 subjects in each
# cluster, with an ICC of .3, sigma squared of 20 (implying sigma_b^2 of 8.57143)
# in each group, with arm means of 1 and 4.75 in the two groups, using 100 simulated
# data sets. The resulting estimated power should be 0.78.

## Not run:

normal.sim = cps.normal(nsim = 100, nsubjects = 25, nclusters = 10, mu = 1,
  mu2 = 4.75, ICC = 0.3, sigma_sq = 20, seed = 123)

## End(Not run)



# Estimate power for a trial with 5 clusters in one arm, those clusters having 25 subjects
# each, 25 clusters in the other arm, those clusters having 5 subjects each, the first arm
# having a sigma squared of 20 and sigma_b squared of 8.57143, and the second a sigma squared
```

```
# of 9 and a sigma_b squared of 1, with estimated arm means of 1 and 4.75 in the first and
# second groups, respectively, using 100 simulated data sets analyzed by the GEE method.
# The estimated power should be 0.79, assuming seed = 123.

## Not run:
normal.sim2 = cps.normal(nsim = 100, nclusters = c(5,25), nsubjects = c(25,5), mu = 1,
  mu2 = 4.75, sigma_sq = 20,sigma_b_sq = 8.8571429, sigma_sq2 = 9, sigma_b_sq2 = 1,
  method = "gee", seed = 123)

## End(Not run)


# Estimate power for a trial with 5 clusters in one arm, those clusters having
# 4, 5, 6, 7, 7, and 7 subjects each, and 10 clusters in the other arm,
# those clusters having 5 subjects each, with sigma_b_sq = .3 and and ICC of .3 in both arms.
# We have estimated arm means of 1 and 2 in the first and second arms, respectively, and we use
# 100 simulated data sets analyzed by the GLMM method.

## Not run:
normal.sim2 = cps.normal(nsim = 100, nclusters = c(6,10),
  nsubjects = list(c(4, 5, 6, 7, 7, 7), rep(5, times = 10)),
  mu = 1, mu2 = 2, sigma_b_sq = .3, ICC = .3, method = "glmm",
  seed = 1)

## End(Not run)

# The resulting estimated power (if you set seed = 1) should be about 0.76.

# Estimate power for a trial with 3 clusters in one arm,
# those clusters having 25, 35, and 45 subjects each, and 10 clusters
# in the other arm, those clusters having 5 subjects each, the first arm
# having a sigma squared of 20 and sigma_b squared of 8.57143, and the second a sigma squared
# of 9 and a sigma_b squared of 1, with estimated arm means of 1 and 4.75 in the first and
# second groups, respectively, using 100 simulated data sets analyzed by the GLMM method.

## Not run:

normal.sim2 <- cps.normal(nsim = 100, nclusters = c(3,10),
  nsubjects = c(25, 35, 45, rep(5, times = 10)),
  mu = 1, mu2 = 4.75, sigma_sq = 20, sigma_b_sq = 8.8571429,
  sigma_sq2 = 9, sigma_b_sq2 = 1, method = "glmm")

## End(Not run)

# The resulting estimated power (if you set seed = 1) should be about 0.71.
```

`cps.sw.binary` *Power simulations for cluster-randomized trials: Stepped Wedge Design, Binary Outcome*

Description

This set of functions utilize iterative simulations to determine approximate power for stepped wedge cluster-randomized controlled trials. Users can modify a variety of parameters to suit the simulations to their desired experimental situation.

Usage

```
cps.sw.binary(
  nsim = NULL,
  nsubjects = NULL,
  nclusters = NULL,
  p0 = NULL,
  p1 = NULL,
  steps = NULL,
  sigma_b_sq = NULL,
  alpha = 0.05,
  method = "glmm",
  quiet = FALSE,
  allSimData = FALSE,
  poorFitOverride = FALSE,
  lowPowerOverride = FALSE,
  timelimitOverride = TRUE,
  optmethod = "L-BFGS-B",
  seed = NULL
)
```

Arguments

`nsim`	Number of datasets to simulate; accepts integer (required).
`nsubjects`	Number of subjects per cluster; accepts either a scalar (equal cluster sizes) or a vector of length `nclusters` (user-defined size for each cluster) (required).
`nclusters`	Number of clusters; accepts non-negative integer scalar (required).
`p0`	Expected probability of outcome in arm 1. Accepts scalar between 0 - 1 (required).
`p1`	Expected probability of outcome in arm 2. Accepts scalar between 0 - 1 (required).
`steps`	Number of crossover steps; a baseline step (all clusters in arm 1) is assumed. Accepts positive scalar (indicating the total number of steps; clusters per step is obtained by `nclusters / steps`) or a vector of non-negative integers corresponding either to the number of clusters to be crossed over at each time point (e.g c(2,4,4,2); nclusters = 10) or the cumulative number of clusters crossed over by a given time point (e.g. c(2,4,8,10); nclusters = 10) (required).
`sigma_b_sq`	Between-cluster variance; accepts non-negative numeric scalar (indicating equal between-cluster variances for both arms) or a vector of length 2 specifying treatment-specific between-cluster variances (required).
`alpha`	Significance level. Default = 0.05.
`method`	Analytical method, either Generalized Linear Mixed Effects Model (GLMM) or Generalized Estimating Equation (GEE). Accepts c('glmm', 'gee') (required); default = 'glmm'.

`quiet` When set to FALSE, displays simulation progress and estimated completion time; default is FALSE.

`allSimData` Option to output list of all simulated datasets; default = FALSE.

`poorFitOverride`
Option to override `stop()` if more than 25% of fits fail to converge; default = FALSE.

`lowPowerOverride`
Option to override `stop()` if the power is less than 0.5 after the first 50 simulations and every ten simulations thereafter. On function execution stop, the actual power is printed in the stop message. Default = FALSE. When TRUE, this check is ignored and the calculated power is returned regardless of value.

`timelimitOverride`
Logical. When FALSE, stops execution if the estimated completion time is more than 2 minutes. Defaults to TRUE.

`optmethod` Option to fit with a different optimizer method (using the package `optimx`). Default is 'L-BFGS-B'.

`seed` Option to set.seed. Default is NULL.

Details

Runs power simulations for stepped wedge cluster-randomized controlled trials with a binary outcome. The stepped wedge trial design is a type of cross-over design in which clusters change treatments in waves. Initially all the clusters recieve the same standard treatment, and at the end of the trial all of the clusters will be recieving the treatment of interest. More than one cluster can change treatments in a wave, but the order in which clusters change treatments is randomly determined. The outcome of interest is assessed in each cluster during each wave.

Users must specify the desired number of simulations, number of subjects per cluster, number of clusters per arm, expected absolute difference between arms, within-cluster variance, between-cluster variance, significance level, analytic method, progress updates, and simulated data set output may also be specified.

Value

A list with the following components

- Character string indicating total number of simulations and simulation type
- Number of simulations
- Data frame with columns "Power" (Estimated statistical power), "lower.95.ci" (Lower 95 "upper.95.ci" (Upper 95
- Analytic method used for power estimation
- Significance level
- Vector containing user-defined cluster sizes
- Vector containing user-defined number of clusters
- Data frame reporting ICC, within & between cluster variances for both arms at each time point
- Vector containing expected difference between groups based on user inputs
- Data frame containing mean response values for each arm at each time point
- Matrix showing cluster crossover at each time point

- Data frame with columns: "Estimate" (Estimate of treatment effect for a given simulation), "Std.err" (Standard error for treatment effect estimate), "Test.statistic" (z-value (for GLMM) or Wald statistic (for GEE)), "p.value", "sig.val" (Is p-value less than alpha?)
- If `allSimData = TRUE`, a list of data frames, each containing: "y" (Simulated response value), "trt" (Indicator for arm), "time.point" (Indicator for step; "t1" = time point 0) "clust" (Indicator for cluster), "period" (Indicator for at which step a cluster crosses over)

If `nofit = T`, a data frame of the simulated data sets, containing:

- "arm" (Indicator for treatment arm)
- "clust" (Indicator for cluster)
- "y1" ... "yn" (Simulated response value for each of the `nsim` data sets).

Author(s)

Alexandria C. Sakrejda (<acbro0@umass.edu>)

Alexander R. Bogdan

Ken Kleinman (<ken.kleinman@gmail.com>)

Examples

```
# Estimate power for a trial with 3 steps and 12 clusters in arm 1
# (often the standard-of-care or 'control' arm) at the initiation of the study.
# Those clusters have 50 subjects each, with sigma_b_sq = 1.
# We have estimated arm outcome proportions of 0.1 and 0.2 in the first and second arms,
# respectively, and 100 simulated data sets analyzed by the GLMM method. Using seed = 123,
# the resulting power should be 0.8.

## Not run:
binary.sw.rct = cps.sw.binary(nsim = 100, nsubjects = 50, nclusters = 12,
                              p0 = 0.1, p1 = 0.2, steps = 3,
                              sigma_b_sq = 1, alpha = 0.05, method = 'glmm',
                              quiet = FALSE, allSimData = FALSE, seed = 123)

## End(Not run)
```

---

cps.sw.count — *Power simulations for cluster-randomized trials: Stepped Wedge Design, Count Outcome*

Description

This set of functions utilize iterative simulations to determine approximate power for stepped wedge cluster-randomized controlled trials. Users can modify a variety of parameters to suit the simulations to their desired experimental situation.

Usage

```
cps.sw.count(
  nsim = NULL,
  nsubjects = NULL,
  nclusters = NULL,
  c0 = NULL,
  c1 = NULL,
  steps = NULL,
  sigma_b_sq = NULL,
  alpha = 0.05,
  family = "poisson",
  analysis = "poisson",
  negBinomSize = 1,
  method = "glmm",
  quiet = FALSE,
  allSimData = FALSE,
  poorFitOverride = FALSE,
  lowPowerOverride = FALSE,
  timelimitOverride = TRUE,
  opt = "L-BFGS-B",
  seed = NULL
)
```

Arguments

nsim	Number of datasets to simulate; accepts integer (required).
nsubjects	Number of subjects per cluster; accepts either a scalar (equal cluster sizes) or a vector of length nclusters (user-defined size for each cluster) (required).
nclusters	Number of clusters; accepts non-negative integer scalar (required).
c0	Expected outcome count in arm 1. Accepts scalar (required).
c1	Expected outcome count in arm 2. Accepts scalar (required).
steps	Number of crossover steps; a baseline step (all clusters in arm 1) is assumed. Accepts positive scalar (indicating the total number of steps; clusters per step is obtained by nclusters / steps) or a vector of non-negative integers corresponding either to the number of clusters to be crossed over at each time point (e.g c(2,4,4,2); nclusters = 10) or the cumulative number of clusters crossed over by a given time point (e.g. c(2,4,8,10); nclusters = 10) (required).
sigma_b_sq	Between-cluster variance; accepts non-negative numeric scalar (indicating equal between-cluster variances for both arms) or a vector of length 2 specifying treatment-specific between-cluster variances (required).
alpha	Significance level. Default = 0.05.
family	Distribution from which responses are simulated. Accepts Poisson ('poisson') or negative binomial ('neg.binom') (required); default = 'poisson'
analysis	Family used for regression; currently only applicable for GLMM. Accepts c('poisson', 'neg.binom') (required); default = 'poisson'
negBinomSize	Only used when generating simulated data from the negative binomial (family = 'neg.binom'), this is the target for number of successful trials, or the dispersion parameter (the shape parameter of the gamma mixing distribution). Must be strictly positive but need not be integer. Defaults to 1.

`method` Analytical method, either Generalized Linear Mixed Effects Model (GLMM) or Generalized Estimating Equation (GEE). Accepts c('glmm', 'gee') (required); default = 'glmm'.

`quiet` When set to FALSE, displays simulation progress and estimated completion time; default is FALSE.

`allSimData` Option to output list of all simulated datasets; default = FALSE.

`poorFitOverride` Option to override `stop()` if more than 25% of fits fail to converge; default = FALSE.

`lowPowerOverride` Option to override `stop()` if the power is less than 0.5 after the first 50 simulations and every ten simulations thereafter. On function execution stop, the actual power is printed in the stop message. Default = FALSE. When TRUE, this check is ignored and the calculated power is returned regardless of value.

`timelimitOverride` Logical. When FALSE, stops execution if the estimated completion time is more than 2 minutes. Defaults to TRUE.

`opt` Option to fit with a different optimizer (using the package `optimx`). Default is 'L-BFGS-B'.

`seed` Option to set.seed. Default is NULL.

Details

Runs power simulations for stepped wedge cluster-randomized controlled trials with a count outcome. The stepped wedge trial design is a type of cross-over design in which clusters change treatments in waves. Initially all the clusters recieve the same standard treatment, and at the end of the trial all of the clusters will be recieving the treatment of interest. More than one cluster can change treatments in a wave, but the order in which clusters change treatments is randomly determined. The outcome of interest is assessed in each cluster during each wave.

Users must specify the desired number of simulations, number of subjects per cluster, number of clusters per arm, expected absolute difference between arms, between-cluster variance; significance level, distributional family for data generation, analytic method, progress updates, and simulated data set output may also be specified.

Value

A list with the following components

- Character string indicating total number of simulations and simulation type
- Number of simulations
- Data frame with columns "Power" (Estimated statistical power), "lower.95.ci" (Lower 95 "upper.95.ci" (Upper 95
- Analytic method used for power estimation
- Significance level
- Vector containing user-defined cluster sizes
- Vector containing user-defined number of clusters
- Data frame reporting ICC, within & between cluster variances for both arms at each time point
- Vector containing expected difference between groups based on user inputs
- Data frame containing mean response values for each arm at each time point

- Matrix showing cluster crossover at each time point
- Data frame with columns: "Estimate" (Estimate of treatment effect for a given simulation), "Std.err" (Standard error for treatment effect estimate), "Test.statistic" (z-value (for GLMM) or Wald statistic (for GEE)), "p.value", "sig.val" (Is p-value less than alpha?)
- If `allSimData = TRUE`, a list of data frames, each containing: "y" (Simulated response value), "trt" (Indicator for arm), "time.point" (Indicator for step; "t1" = time point 0) "clust" (Indicator for cluster), "period" (Indicator for at which step a cluster crosses over)

If `nofit = T`, a data frame of the simulated data sets, containing:

- "arm" (Indicator for treatment arm)
- "cluster" (Indicator for cluster)
- "y1" ... "yn" (Simulated response value for each of the `nsim` data sets).

Author(s)

Alexander R. Bogdan

Alexandria C. Sakrejda (<acbro0@umass.edu>)

Ken Kleinman (<ken.kleinman@gmail.com>)

Examples

```
# Estimate power for a trial with 3 steps and 12 clusters in arm 1
# (often the standard-of-care or 'control' arm) at the initiation
#of the study. Those clusters have 15 subjects each, with sigma_b_sq = 1.
# We have estimated arm outcome counts of 4 and 5 in the first and second arms,
# respectively, and 100 simulated data sets analyzed by the GLMM method. Using seed = 123,
# the resulting power should be 0.85.

## Not run:
count.sw.rct = cps.sw.count(nsim = 100, nsubjects = 15, nclusters = 12,
                            c0 = 4, c1 = 5, steps = 3, sigma_b_sq = 1,
                            alpha = 0.05, family = 'poisson', analysis = 'poisson',
                            method = 'glmm', seed = 123)

## End(Not run)
```

`cps.sw.normal` *Power simulations for cluster-randomized trials: Stepped Wedge Design, Continuous Outcome.*

Description

This set of functions utilize iterative simulations to determine approximate power for stepped wedge cluster-randomized controlled trials. Users can modify a variety of parameters to suit the simulations to their desired experimental situation.

Usage

```
cps.sw.normal(
  nsim = NULL,
  nsubjects = NULL,
  nclusters = NULL,
  mu0 = 0,
  mu1 = NULL,
  steps = NULL,
  sigma_sq = NULL,
  sigma_b_sq = NULL,
  alpha = 0.05,
  method = "glmm",
  quiet = FALSE,
  allSimData = FALSE,
  poorFitOverride = FALSE,
  lowPowerOverride = FALSE,
  timelimitOverride = TRUE,
  seed = NULL
)
```

Arguments

`nsim`	Number of datasets to simulate; accepts integer (required).
`nsubjects`	Number of subjects per cluster; accepts either a scalar (equal cluster sizes) or a vector of length `nclusters` (user-defined size for each cluster) (required).
`nclusters`	Number of clusters; accepts non-negative integer scalar (required).
`mu0`	Expected baseline mean; accepts numeric, default 0. Required..
`mu1`	Expected post-treatment mean; accepts numeric. Required.
`steps`	Number of crossover steps; a baseline step (all clusters in non-treatment group) is assumed. Accepts positive scalar (indicating the total number of steps; clusters per step is obtained by `nclusters / steps`) or a vector of non-negative integers corresponding either to the number of clusters to be crossed over at each time point (e.g c(2,4,4,2); nclusters = 10) or the cumulative number of clusters crossed over by a given time point (e.g. c(2,4,8,10); nclusters = 10) (required).
`sigma_sq`	Within-cluster variance; accepts non-negative numeric scalar (indicating equal within-cluster variances for both treatment groups) or a vector of length 2 specifying within-cluster variances for the non-treatment and treatment groups, respectively (required).
`sigma_b_sq`	Between-cluster variance; accepts non-negative numeric scalar (indicating equal between-cluster variances for both treatment groups) or a vector of length 2 specifying treatment-specific between-cluster variances (required).
`alpha`	Significance level. Default = 0.05.
`method`	Analytical method, either Generalized Linear Mixed Effects Model (GLMM) or Generalized Estimating Equation (GEE). Accepts c('glmm', 'gee') (required); default = 'glmm'.
`quiet`	When set to FALSE, displays simulation progress and estimated completion time; default is FALSE.
`allSimData`	Option to output list of all simulated datasets; default = FALSE.

`poorFitOverride`
: Option to override `stop()` if more than 25% of fits fail to converge; default = FALSE.

`lowPowerOverride`
: Option to override `stop()` if the power is less than 0.5 after the first 50 simulations and every ten simulations thereafter. On function execution stop, the actual power is printed in the stop message. Default = FALSE. When TRUE, this check is ignored and the calculated power is returned regardless of value.

`timelimitOverride`
: Logical. When FALSE, stops execution if the estimated completion time is more than 2 minutes. Defaults to TRUE.

`seed`
: Option to set.seed. Default is NULL.

Details

Runs power simulations for stepped wedge cluster-randomized controlled trials with continuous outcome. The stepped wedge trial design is a type of cross-over design in which clusters change treatments in waves. Initially all the clusters recieve the same standard treatment, and at the end of the trial all of the clusters will be recieving the treatment of interest. More than one cluster can change treatments in a wave, but the order in which clusters change treatments is randomly determined. The outcome of interest is assessed in each cluster during each wave.

Users must specify the desired number of simulations, number of subjects per cluster, number of clusters per arm, expected means for each arm, within-cluster variance, between-cluster variance, significance level, analytic method, progress updates, and simulated data set output may also be specified.

Value

A list with the following components

- Character string indicating total number of simulations and simulation type
- Number of simulations
- Data frame with columns "Power" (Estimated statistical power), "lower.95.ci" (Lower 95 "upper.95.ci" (Upper 95
- Analytic method used for power estimation
- Significance level
- Vector containing user-defined cluster sizes
- Vector containing user-defined number of clusters
- Data frame reporting ICC, within & between cluster variances for Treatment/Non-Treatment groups at each time point
- Vector containing expected means for each arm based on user inputs
- Data frame containing mean response values for each treatment group at each time point
- Matrix showing cluster crossover at each time point
- Data frame with columns: "Estimate" (Estimate of treatment effect for a given simulation), "Std.err" (Standard error for treatment effect estimate), "Test.statistic" (z-value (for GLMM) or Wald statistic (for GEE)), "p.value", "sig.val" (Is p-value less than alpha?)
- If `allSimData = TRUE`, a list of data frames, each containing: "y" (Simulated response value), "trt" (Indicator for treatment group), "time.point" (Indicator for step; "t1" = time point 0) "clust" (Indicator for cluster), "period" (Indicator for at which step a cluster crosses over)

If nofit = T, a data frame of the simulated data sets, containing:

- "arm" (Indicator for treatment arm)
- "cluster" (Indicator for cluster)
- "y1" ... "yn" (Simulated response value for each of the nsim data sets).

Author(s)

Alexander R. Bogdan

Alexandria C. Sakrejda (<acbro0@umass.edu>)

Ken Kleinman (<ken.kleinman@gmail.com>)

Examples

```
# Estimate power for a trial with 3 steps and 9 clusters in arm 1 (often the
# standard-of-care or 'control' arm) at the initiation of the study. Those
# clusters have 14 subjects each, with sigma_b = 1 and sigma_b_sq = 1. We
# have estimated arm outcome means of 1 and 2.1 in the first and second arms,
# respectively, and 100 simulated data sets analyzed by the GLMM method. Using seed = 123,
# the resulting power should be 0.82.

## Not run:
normal.sw.rct = cps.sw.normal(nsim = 100, nsubjects = 14, nclusters = 9,
                              mu0 = 1, mu1 = 2.1, steps = 3, sigma_sq = 1,
                              sigma_b_sq = 1, alpha = 0.05, method = 'glmm',
                              seed = 123)

## End(Not run)
```

---

package_map_helper *Look up which internal functions are called by exported functions.*

Description

Look up which internal functions are called by exported functions.

Usage

```
package_map_helper(packageName = "clusterPower")
```

Arguments

packageName	The name of the package in quotes. Defaults to "clusterPower".

Value

List of internal functions and the line numbers in which they appear inside external functions.

Author(s)

Alexandria Sakrejda (<acbro0@umass.edu>)

`runExample` *Run a Shiny app for power analysis functions*

Description

This function runs the clusterPower Shiny apps.

Usage

```
runExample(appname = "main")
```

Arguments

`appname`	which app should be launched. Choices are either "main" or "analytic". Default is "main".

Author(s)

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

Alexandria Sakrejda (<acbro0@umass.edu>)

Index