Package ‘odr’

March 13, 2020

Type Package

Title Optimal Design and Statistical Power of Multilevel Randomized Trials

Version 1.0.2

Date 2020-3-13

Description Calculate the optimal sample allocation that produces smallest variance of a treatment effect or the highest statistical power
for experimental studies under a budget constraint,
perform power analyses with and without accommodating cost structures of sampling, and calculate
the relative efficiency between two sample allocations.
The references
for the proposed methods are:
(1) Shen, Z. (2019). Optimal sample allocation in multilevel Experiments.
(Doctoral dissertation). University of Cincinnati, Cincinnati, OH.
(2) Shen, Z., & Kelcey, B. (in press).
Optimal sample allocation under unequal costs in
cluster-randomized trials.
Journal of Educational and Behavioral Statistics.
(3) Champely., S. (2018). pwr: Basic functions for power analysis
(Version 1.2-2) [Software]. Available from
<https://CRAN.R-project.org/package=pwr>.

Depends R (>= 3.3.0), stats (>= 3.0.0), graphics (>= 3.0.0), base(>= 3.0.0)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Suggests knitr, rmarkdown

VignetteBuilder rmarkdown, knitr

NeedsCompilation no

Author Zuchao Shen [aut, cre],
Ben Kelcey [aut]

Maintainer Zuchao Shen <zuchao.shen@gmail.com>

Repository CRAN

Date/Publication 2020-03-13 11:30:02 UTC

R topics documented:

`odr-package` *Optimal design and statistical power of experimental studies assessing main and mediation effects*

Description

This package is to help researchers design cost-efficient experimental studies assessing main treatment effects with adequate statistical precision by (a) solving optimal sample allocations, (b) comparing design precision and efficiency between different sample allocations, and (c) explicitly accommodating costs and budget in power analyses.

Details

The package covers seven types of experiments aiming to detect main effects on continuous outcomes. These experiments are individual randomized controlled trials (RCTs), two-, three-, and four-level cluster-randomized trials (CRTs), and two-, three-, and four-level multisite randomized trials (MRTs). There are two categorical functions for each type of experiments and a uniform function for all types of experiments. The two categorical functions are 'od' and 'power'. The 'od' function can calculate the optimal sample allocation with and without constraint for each type of experiments. The 'power' function by default can calculate required budget (and required sample size) for desired power, minimum detectable effect size (MDES) under a fixed budget, statistical power under a fixed budget. The 'power' function also can perform conventional power analyses (e.g., required sample size, power, MDES calculation). The uniform function 're' (or 'rpe') is to compare the relative (precision and) efficiency between two designs with different sample allocations.

Author(s)

Zuchao Shen, Ben Kelcey

Maintainer: Zuchao Shen zuchao.shen@gmail.com

---

od.1 *Optimal sample allocation calculation for individual randomized controlled trials*

Description

The optimal design of individual randomized controlled trials (RCTs) is to choose the sample allocation that minimizes the variance of treatment effect under fixed budget and cost structure. The optimal design parameter is the proportion of individuals to be assigned to treatment (`p`).

Usage

```
od.1(p = NULL, r12 = NULL, c1 = NULL, c1t = NULL, m = NULL,
  plots = TRUE, plim = NULL, varlim = NULL, plab = NULL,
  varlab = NULL, vartitle = NULL, verbose = TRUE)
```

Arguments

`p`	the proportion of individuals to be assigned to treatment.
`r12`	the proportion of outcome variance explained by covariates.
`c1`	the cost of sampling one unit in control condition.
`c1t`	the cost of sampling one unit in treatment condition.
`m`	total budget, default value is the total costs of sampling 60 individuals across treatment conditions.
`plots`	logical, provide variance plots if TRUE, otherwise not; default value is TRUE.
`plim`	the plot range for p, default value is c(0, 1).
`varlim`	the plot range for variance, default value is c(0, 0.05).
`plab`	the plot label for `p` , default value is "Proportion of Individuals in Treatment: p"
`varlab`	the plot label for variance, default value is "Variance".
`vartitle`	the title of variance plot, default value is NULL.
`verbose`	logical; print the value of `p` if TRUE, otherwise not; default value is TRUE.

Value

unconstrained or constrained optimal sample allocation (p). The function also returns the variance of treatment effect, function name, design type, and parameters used in the calculation.

References

Shen, Z. (in progress). Using optimal sample allocation to improve statistical precision and design efficiency for multilevel randomized trials (Unpublished doctoral dissertation). University of Cincinnati, Cincinnati, OH.

Examples

```
# unconstrained optimal design #---------
  myod1 <- od.1(r12 = 0.5, c1 = 1, c1t = 5, varlim = c(0, 0.2))
  myod1$out # output

# constrained p, no calculation performed #---------
  myod2 <- od.1(r12 = 0.5, c1 = 1, c1t = 5, varlim = c(0, 0.2), p = 0.5)
  myod2$out
# relative efficiency (RE)
  myre <- re(od = myod1, subod= myod2)
  myre$re # RE = 0.87

# when sampling costs are equal, a balanced  design with p = 0.5 is the best #---------
  myod3 <- od.1(r12 = 0.5, c1 = 1, c1t = 1, varlim = c(0, 0.2))
  myod3$out # output
```

`od.2` *Optimal sample allocation calculation for two-level CRTs*

Description

The optimal design of two-level cluster randomized trials (CRTs) is to choose the sample allocation that minimizes the variance of treatment effect under fixed budget and cost structure. The optimal design parameters include the level-1 sample size per level-2 unit (n) and the proportion of level-2 clusters/groups to be assigned to treatment (p). This function solves the optimal n and/or p with and without constraints.

Usage

```
od.2(n = NULL, p = NULL, icc = NULL, r12 = NULL, r22 = NULL,
  c1 = NULL, c2 = NULL, c1t = NULL, c2t = NULL, m = NULL,
  plots = TRUE, plot.by = NULL, nlim = NULL, plim = NULL,
  varlim = NULL, nlab = NULL, plab = NULL, varlab = NULL,
  vartitle = NULL, verbose = TRUE)
```

Arguments

`n`	the level-1 sample size per level-2 unit.
`p`	the proportion of level-2 clusters/units to be assigned to treatment.
`icc`	the unconditional intraclass correlation coefficient (ICC) in population or in each treatment condition.
`r12`	the proportion of level-1 variance explained by covariates.
`r22`	the proportion of level-2 variance explained by covariates.
`c1`	the cost of sampling one level-1 unit in control condition.
`c2`	the cost of sampling one level-2 unit in control condition.
`c1t`	the cost of sampling one level-1 unit in treatment condition.
`c2t`	the cost of sampling one level-2 unit in treatment condition.
`m`	total budget, default value is the total costs of sampling 60 level-2 units across treatment conditions.
`plots`	logical, provide variance plots if TRUE, otherwise not; default value is TRUE.
`plot.by`	specify variance plot by `n` and/or `p`; default value is plot.by = list(n = "n", p = "p").
`nlim`	the plot range for n, default value is c(2, 50).
`plim`	the plot range for p, default value is c(0, 1).
`varlim`	the plot range for variance, default value is c(0, 0.05).
`nlab`	the plot label for `n`, default value is "Level-1 Sample Size: n".
`plab`	the plot label for `p`, default value is "Proportion Level-2 Units in Treatment: p"
`varlab`	the plot label for variance, default value is "Variance".
`vartitle`	the title of variance plot, default value is NULL.
`verbose`	logical; print the values of `n` and `p` if TRUE, otherwise not; default value is TRUE.

Value

unconstrained or constrained optimal sample allocation (`n` and `p`). The function also returns the variance of treatment effect, function name, design type, and parameters used in the calculation.

References

Shen, Z., & Kelcey, B. (2018, April). Optimal design of cluster randomized trials under condition- and unit-specific cost structures. Roundtable discussion presented at American Educational Research Association (AERA) annual conference, New York City, NY;

Shen, Z., & Kelcey, B. (revise & resubmit). Optimal sample allocation under unequal costs in cluster-randomized trials. Journal of Educational and Behavioral Statistics.

Shen, Z. (in progress). Using optimal sample allocation to improve statistical precision and design efficiency for multilevel randomized trials (Unpublished doctoral dissertation). University of Cincinnati, Cincinnati, OH.

Examples

```
# unconstrained optimal design #---------
  myod1 <- od.2(icc = 0.2, r12 = 0.5, r22 = 0.5, c1 = 1, c2 = 5, c1t = 1, c2t = 50,
              varlim = c(0.01, 0.02))
  myod1$out # output
# plot by p
  myod1 <- od.2(icc = 0.2, r12 = 0.5, r22 = 0.5, c1 = 1, c2 = 5, c1t = 1, c2t = 50,
              varlim = c(0.01, 0.02), plot.by = list(p = 'p'))

# constrained optimal design with n = 20 #---------
  myod2 <- od.2(icc = 0.2, r12 = 0.5, r22 = 0.5, c1 = 1, c2 = 5, c1t = 1, c2t = 50,
              n = 20, varlim = c(0.005, 0.025))
  myod2$out
# relative efficiency (RE)
  myre <- re(od = myod1, subod= myod2)
  myre$re # RE = 0.88

# constrained optimal design with p = 0.5 #---------
  myod3 <- od.2(icc = 0.2, r12 = 0.5, r22 = 0.5, c1 = 1, c2 = 5, c1t = 1, c2t = 50,
             p = 0.5, varlim = c(0.005, 0.025))
  myod3$out
# relative efficiency (RE)
  myre <- re(od = myod1, subod= myod3)
  myre$re # RE = 0.90

# constrained n and p, no calculation performed #---------
  myod4 <- od.2(icc = 0.2, r12 = 0.5, r22 = 0.5, c1 = 1, c2 = 5, c1t = 1, c2t = 50,
              n = 20, p = 0.5, varlim = c(0.005, 0.025))
  myod4$out
# relative efficiency (RE)
  myre <- re(od = myod1, subod= myod4)
  myre$re # RE = 0.83
```

`od.2m` *Optimal sample allocation calculation for two-level multisite randomized trials*

Description

The optimal design of two-level multisite randomized trials (MRTs) is to choose the sample allocation that minimizes the variance of a treatment effect under a fixed budget and cost structure. The optimal design parameters include the level-one sample size per site (n) and the proportion of level-one unit to be assigned to treatment (p). This function solves the optimal n and/or p with and without constraints.

Usage

```
od.2m(n = NULL, p = NULL, icc = NULL, r12 = NULL, r22m = NULL,
```

```
c1 = NULL, c2 = NULL, c1t = NULL, omega = NULL, m = NULL,
plots = TRUE, plot.by = NULL, nlim = NULL, plim = NULL,
varlim = NULL, nlab = NULL, plab = NULL, varlab = NULL,
vartitle = NULL, verbose = TRUE, iter = 100, tol = 1e-10)
```

Arguments

`n`	the level-1 sample size per level-2 unit.
`p`	the proportion of level-4 clusters/units to be assigned to treatment.
`icc`	the unconditional intraclass correlation coefficient (ICC) in population or in each treatment condition.
`r12`	the proportion of level-1 variance explained by covariates.
`r22m`	the proportion of variance of site-specific treatment effect explained by covariates.
`c1`	the cost of sampling one level-1 unit in control condition.
`c2`	the cost of sampling one level-2 unit in control condition.
`c1t`	the cost of sampling one level-1 unit in treatment condition.
`omega`	the standardized variance of site-specific treatment effect
`m`	total budget, default is the total costs of sampling 60 sites.
`plots`	logical, provide variance plots if TRUE, otherwise not; default value is TRUE.
`plot.by`	specify variance plot by `n` and/or `p`; default value is plot.by = list(n = "n", p = "p").
`nlim`	the plot range for n, default value is c(2, 50).
`plim`	the plot range for p, default value is c(0, 1).
`varlim`	the plot range for variance, default value is c(0, 0.05).
`nlab`	the plot label for `n`, default value is "Level-1 Sample Size: n".
`plab`	the plot label for `p`, default value is "Proportion Level-1 Units in Treatment: p".
`varlab`	the plot label for variance, default value is "Variance".
`vartitle`	the title of variance plot, default value is NULL.
`verbose`	logical; print the values of `n` and `p` if TRUE, otherwise not; default value is TRUE.
`iter`	number of iterations; default value is 100.
`tol`	tolerance for convergence; default value is 1e-10.

Value

unconstrained or constrained optimal sample allocation (`n` and `p`). The function also returns the variance of treatment effect, function name, design type, and parameters used in the calculation.

Examples

```
# unconstrained optimal design #---------
  myod1 <- od.2m(icc = 0.2, omega = 0.02, r12 = 0.5, r22m = 0.5,
              c1 = 1, c2 = 10, c1t = 10,
              varlim = c(0, 0.005))
  myod1$out # n = 20, p =0.37
# plots by p
  myod1 <- od.2m(icc = 0.2, omega = 0.02,
              r12 = 0.5, r22m = 0.5,
              c1 = 1, c2 = 10, c1t = 10,
              varlim = c(0, 0.005), plot.by = list(p = 'p'))

# constrained optimal design with p = 0.5 #---------
  myod2 <- od.2m(icc = 0.2, omega = 0.02,
              r12 = 0.5, r22m = 0.5,
              c1 = 1, c2 = 10, c1t = 10,
              varlim = c(0, 0.005), p = 0.5)
  myod2$out
# relative efficiency (RE)
  myre <- re(od = myod1, subod= myod2)
  myre$re # RE = 0.86

# constrained optimal design with n = 5 #---------
  myod3 <- od.2m(icc = 0.2, omega = 0.02,
              r12 = 0.5, r22m = 0.5, c1 = 1, c2 = 10,
              c1t = 10, varlim = c(0, 0.005), n = 5)
  myod3$out
# relative efficiency (RE)
  myre <- re(od = myod1, subod= myod3)
  myre$re # RE = 0.79

# constrained n and p, no calculation performed #---------
  myod4 <- od.2m(icc = 0.2, omega = 0.02, r12 = 0.5, r22m = 0.5,
              c1 = 1, c2 = 10, c1t = 10,
              varlim = c(0, 0.005), p = 0.5, n = 10)
  myod4$out
# relative efficiency (RE)
  myre <- re(od = myod1, subod= myod4)
  myre$re # RE = 0.84
```

`od.3` *Optimal sample allocation calculation for three-level CRTs*

Description

The optimal design of three-level cluster randomized trials (CRTs) is to choose the sample allocation that minimizes the variance of treatment effect under fixed budget and cost structure. The optimal design parameters include the level-1 sample size per level-2 unit (`n`), the level-2 sample size per level-3 unit (`J`), and the proportion of level-3 clusters/groups to be assigned to treatment (`p`). This function solves the optimal `n`, `J` and/or `p` with and without constraints.

Usage

```
od.3(n = NULL, J = NULL, p = NULL, icc2 = NULL, icc3 = NULL,
  r12 = NULL, r22 = NULL, r32 = NULL, c1 = NULL, c2 = NULL,
  c3 = NULL, c1t = NULL, c2t = NULL, c3t = NULL, m = NULL,
  plots = TRUE, plot.by = NULL, nlim = NULL, Jlim = NULL,
  plim = NULL, varlim = NULL, nlab = NULL, Jlab = NULL,
  plab = NULL, varlab = NULL, vartitle = NULL, verbose = TRUE,
  iter = 100, tol = 1e-10)
```

Arguments

`n`	the level-1 sample size per level-2 unit.
`J`	the level-2 sample size per level-3 unit.
`p`	the proportion of level-3 clusters/units assigned to treatment.
`icc2`	the unconditional intraclass correlation coefficient (ICC) at level 2.
`icc3`	the unconditional intraclass correlation coefficient (ICC) at level 3.
`r12`	the proportion of level-1 variance explained by covariates.
`r22`	the proportion of level-2 variance explained by covariates.
`r32`	the proportion of level-3 variance explained by covariates.
`c1`	the cost of sampling one level-1 unit in control condition.
`c2`	the cost of sampling one level-2 unit in control condition.
`c3`	the cost of sampling one level-3 unit in control condition.
`c1t`	the cost of sampling one level-1 unit in treatment condition.
`c2t`	the cost of sampling one level-2 unit in treatment condition.
`c3t`	the cost of sampling one level-3 unit in treatment condition.
`m`	total budget, default is the total costs of sampling 60 level-3 units across treatment conditions.
`plots`	logical, provide variance plots if TRUE, otherwise not; default value is TRUE.
`plot.by`	specify variance plot by `n`, `J` and/or `p`; default is plot.by = list(n = "n", J = "J", p = "p").
`nlim`	the plot range for n, default value is c(2, 50).
`Jlim`	the plot range for J, default value is c(2, 50).
`plim`	the plot range for p, default value is c(0, 1).
`varlim`	the plot range for variance, default value is c(0, 0.05).
`nlab`	the plot label for `n`, default value is "Level-1 Sample Size: n".
`Jlab`	the plot label for `J`, default value is "Level-2 Sample Size: J".
`plab`	the plot label for p, default is "Proportion Level-3 Units in Treatment: p"
`varlab`	the plot label for variance, default value is "Variance".
`vartitle`	the title of variance plot, default value is NULL.
`verbose`	logical; print the values of `n`, `J`, and `p` if TRUE, otherwise not; default is TRUE.
`iter`	number of iterations; default value is 100.
`tol`	tolerance for convergence; default value is 1e-10.

Value

unconstrained or constrained optimal sample allocation (n, J, and p). The function also returns the variance of treatment effect, function name, design type, and parameters used in the calculation.

References

Shen, Z., & Kelcey, B. (revise & resubmit). Optimal sample allocation under unequal costs in cluster-randomized trials. Journal of Educational and Behavioral Statistics.

Shen, Z. (in progress). Using optimal sample allocation to improve statistical precision and design efficiency for multilevel randomized trials (Unpublished doctoral dissertation). University of Cincinnati, Cincinnati, OH.

Examples

```
# unconstrained optimal design #---------
  myod1 <- od.3(icc2 = 0.2, icc3 = 0.1, r12 = 0.5, r22 = 0.5, r32 = 0.5,
             c1 = 1, c2 = 5, c3 = 25, c1t = 1, c2t = 50, c3t = 250,
             varlim = c(0.005, 0.025))
  myod1$out # output
# plots by p and J
  myod1 <- od.3(icc2 = 0.2, icc3 = 0.1, r12 = 0.5, r22 = 0.5, r32 = 0.5,
             c1 = 1, c2 = 5, c3 = 25, c1t = 1, c2t = 50, c3t = 250,
             varlim = c(0.005, 0.025), plot.by = list(p = 'p', J = 'J'))

# constrained optimal design with J = 20 #---------
  myod2 <- od.3(icc2 = 0.2, icc3 = 0.1, r12 = 0.5, r22 = 0.5, r32 = 0.5, J = 20,
             c1 = 1, c2 = 5, c3 = 25, c1t = 1, c2t = 50, c3t = 250,
             varlim = c(0, 0.025))
  myod2$out
# relative efficiency (RE)
  myre <- re(od = myod1, subod= myod2)
  myre$re # RE = 0.53

# constrained optimal design with p = 0.5 #---------
  myod3 <- od.3(icc2 = 0.2, icc3 = 0.1, r12 = 0.5, r22 = 0.5, r32 = 0.5, p = 0.5,
             c1 = 1, c2 = 5, c3 = 25, c1t = 1, c2t = 50, c3t = 250,
             varlim = c(0.005, 0.025))
  myod3$out
# relative efficiency (RE)
  myre <- re(od = myod1, subod= myod3)
  myre$re # RE = 0.84

# constrained n, J and p, no calculation performed #---------
 myod4 <- od.3(icc2 = 0.2, icc3 = 0.1, r12 = 0.5, r22 = 0.5, r32 = 0.5, n = 10, J = 10, p = 0.5,
             c1 = 1, c2 = 5, c3 = 25, c1t = 1, c2t = 50, c3t = 250,
             varlim = c(0, 0.025))
  myod4$out
# relative efficiency (RE)
  myre <- re(od = myod1, subod= myod4)
  myre$re # RE = 0.61
```

od.3m — *Optimal sample allocation calculation for three-level multisite randomized trials*

Description

The optimal design of three-level multisite randomized trials (MRTs) is to choose the sample allocation that minimizes the variance of treatment effect under fixed budget and cost structure. The optimal design parameters include the level-1 sample size per level-2 unit (`n`), the level-2 sample size per level-3 unit (`J`), and the proportion of level-2 unit to be assigned to treatment (`p`). This function solves the optimal `n`, `J` and/or `p` with and without constraints.

Usage

```
od.3m(n = NULL, J = NULL, p = NULL, icc2 = NULL, icc3 = NULL,
  r12 = NULL, r22 = NULL, r32m = NULL, c1 = NULL, c2 = NULL,
  c3 = NULL, c1t = NULL, c2t = NULL, omega = NULL, m = NULL,
  plots = TRUE, plot.by = NULL, nlim = NULL, Jlim = NULL,
  plim = NULL, varlim = NULL, nlab = NULL, Jlab = NULL,
  plab = NULL, varlab = NULL, vartitle = NULL, verbose = TRUE,
  iter = 100, tol = 1e-10)
```

Arguments

`n`	the level-1 sample size per level-2 unit.
`J`	the level-2 sample size per level-3 unit.
`p`	the proportion of level-4 clusters/units to be assigned to treatment.
`icc2`	the unconditional intraclass correlation coefficient (ICC) at level 2.
`icc3`	the unconditional intraclass correlation coefficient (ICC) at level 3.
`r12`	the proportion of level-1 variance explained by covariates.
`r22`	the proportion of level-2 variance explained by covariates.
`r32m`	the proportion of variance of site-specific treatment effect explained by covariates.
`c1`	the cost of sampling one level-1 unit in control condition.
`c2`	the cost of sampling one level-2 unit in control condition.
`c3`	the cost of sampling one level-3 unit in control condition.
`c1t`	the cost of sampling one level-1 unit in treatment condition.
`c2t`	the cost of sampling one level-2 unit in treatment condition.
`omega`	the standardized variance of site-specific treatment effect
`m`	total budget, default is the total costs of sampling 60 level-3 units.
`plots`	logical, provide variance plots if TRUE, otherwise not; default value is TRUE.

plot.by	specify variance plot by n, J and/or p; default value is plot.by = list(n = "n", J = "J", p = "p").
nlim	the plot range for n, default value is c(2, 50).
Jlim	the plot range for J, default value is c(2, 50).
plim	the plot range for p, default value is c(0, 1).
varlim	the plot range for variance, default value is c(0, 0.05).
nlab	the plot label for n, default value is "Level-1 Sample Size: n".
Jlab	the plot label for J, default value is "Level-2 Sample Size: J".
plab	the plot label for p, default value is "Proportion Level-2 Units in Treatment: p".
varlab	the plot label for variance, default value is "Variance".
vartitle	the title of variance plot, default value is NULL.
verbose	logical; print the values of n, J, and p if TRUE, otherwise not; default value is TRUE.
iter	number of iterations; default value is 100.
tol	tolerance for convergence; default value is 1e-10.

Value

unconstrained or constrained optimal sample allocation (n, J, and p). The function also returns the variance of treatment effect, function name, design type, and parameters used in the calculation.

References

Shen, Z. (in progress). Using optimal sample allocation to improve statistical precision and design efficiency for multilevel randomized trials (Unpublished doctoral dissertation). University of Cincinnati, Cincinnati, OH.

Examples

```
# unconstrained optimal design #---------
  myod1 <- od.3m(icc2 = 0.2, icc3 = 0.1, omega = 0.02,
              r12 = 0.5, r22 = 0.5, r32m = 0.5,
              c1 = 1, c2 = 5,
              c1t = 1, c2t = 200, c3 = 200,
              varlim = c(0, 0.005))
  myod1$out # output
# plots by p and J
  myod1 <- od.3m(icc2 = 0.2, icc3 = 0.1, omega = 0.02,
              r12 = 0.5, r22 = 0.5, r32m = 0.5,
              c1 = 1, c2 = 5,
              c1t = 1, c2t = 200, c3 = 200,
              varlim = c(0, 0.005), plot.by = list(p = 'p', J = 'J'))

# constrained optimal design with p = 0.5 #---------
  myod2 <- od.3m(icc2 = 0.2, icc3 = 0.1, omega = 0.02,
              r12 = 0.5, r22 = 0.5, r32m = 0.5,
              c1 = 1, c2 = 5,
```

```
              c1t = 1, c2t = 200, c3 = 200,
              varlim = c(0, 0.005), p = 0.5)
  myod2$out
# relative efficiency (RE)
  myre <- re(od = myod1, subod= myod2)
  myre$re # RE = 0.81

# constrained optimal design with n = 5 #---------
  myod3 <- od.3m(icc2 = 0.2, icc3 = 0.1, omega = 0.02,
              r12 = 0.5, r22 = 0.5, r32m = 0.5,
              c1 = 1, c2 = 5,
              c1t = 1, c2t = 200, c3 = 200,
              varlim = c(0, 0.005), n = 5)
  myod3$out
# relative efficiency (RE)
  myre <- re(od = myod1, subod= myod3)
  myre$re # RE = 0.89

# constrained n, J and p, no calculation performed #---------
  myod4 <- od.3m(icc2 = 0.2, icc3 = 0.1, omega = 0.02,
              r12 = 0.5, r22 = 0.5, r32m = 0.5,
              c1 = 1, c2 = 5,
              c1t = 1, c2t = 200, c3 = 200,
              varlim = c(0, 0.005), p = 0.5, n = 15, J = 20)
  myod4$out
# relative efficiency (RE)
  myre <- re(od = myod1, subod= myod4)
  myre$re # RE = 0.75
```

`od.4` *Optimal sample allocation calculation for four-level CRTs*

Description

The optimal design of four-level cluster randomized trials (CRTs) is to choose the sample allocation that minimizes the variance of treatment effect under fixed budget and cost structure. The optimal design parameters include the level-1 sample size per level-2 unit (`n`), the level-2 sample size per level-3 unit (`J`), the level-3 sample size per level-4 unit (`K`), and the proportion of level-4 clusters/groups to be assigned to treatment (`p`). This function solves the optimal `n`, `J`, `K` and/or `p` with and without constraints.

Usage

```
od.4(n = NULL, J = NULL, K = NULL, p = NULL, icc2 = NULL,
  icc3 = NULL, icc4 = NULL, r12 = NULL, r22 = NULL, r32 = NULL,
  r42 = NULL, c1 = NULL, c2 = NULL, c3 = NULL, c4 = NULL,
  c1t = NULL, c2t = NULL, c3t = NULL, c4t = NULL, m = NULL,
  plots = TRUE, plot.by = NULL, nlim = NULL, Jlim = NULL,
```

```
Klim = NULL, plim = NULL, varlim = NULL, nlab = NULL,
Jlab = NULL, Klab = NULL, plab = NULL, varlab = NULL,
vartitle = NULL, verbose = TRUE, iter = 100, tol = 1e-10)
```

Arguments

n	the level-1 sample size per level-2 unit.
J	the level-2 sample size per level-3 unit.
K	the level-3 sample size per level-4 unit.
p	the proportion of level-4 clusters/units to be assigned to treatment.
icc2	the unconditional intraclass correlation coefficient (ICC) at level 2.
icc3	the unconditional intraclass correlation coefficient (ICC) at level 3.
icc4	the unconditional intraclass correlation coefficient (ICC) at level 4.
r12	the proportion of level-1 variance explained by covariates.
r22	the proportion of level-2 variance explained by covariates.
r32	the proportion of level-3 variance explained by covariates.
r42	the proportion of level-4 variance explained by covariates.
c1	the cost of sampling one level-1 unit in control condition.
c2	the cost of sampling one level-2 unit in control condition.
c3	the cost of sampling one level-3 unit in control condition.
c4	the cost of sampling one level-4 unit in control condition.
c1t	the cost of sampling one level-1 unit in treatment condition.
c2t	the cost of sampling one level-2 unit in treatment condition.
c3t	the cost of sampling one level-3 unit in treatment condition.
c4t	the cost of sampling one level-4 unit in treatment condition.
m	total budget, default value is the total costs of sampling 60 level-4 units across treatment conditions.
plots	logical, provide variance plots if TRUE, otherwise not; default value is TRUE.
plot.by	specify variance plot by `n`, `J`, `K` and/or `p`; default value is plot.by = list(n = "n", J = "J", K = 'K', p = "p").
nlim	the plot range for n, default value is c(2, 50).
Jlim	the plot range for J, default value is c(2, 50).
Klim	the plot range for K, default value is c(2, 50).
plim	the plot range for p, default value is c(0, 1).
varlim	the plot range for variance, default value is c(0, 0.05).
nlab	the plot label for `n`, default value is "Level-1 Sample Size: n".
Jlab	the plot label for `J`, default value is "Level-2 Sample Size: J".
Klab	the plot label for `K`, default value is "Level-3 Sample Size: K".
plab	the plot label for `p`, default value is "Proportion Level-4 Units in Treatment: p".

`varlab`	the plot label for variance, default value is "Variance".
`vartitle`	the title of variance plot, default value is NULL.
`verbose`	logical; print the values of `n`, `J`, `K`, and `p` if TRUE, otherwise not; default value is TRUE.
`iter`	number of iterations; default value is 100.
`tol`	tolerance for convergence; default value is 1e-10.

Value

unconstrained or constrained optimal sample allocation (`n`, `J`, `K`, and `p`). The function also returns the variance of treatment effect, function name, design type, and parameters used in the calculation.

References

Shen, Z. (in progress). Using optimal sample allocation to improve statistical precision and design efficiency for multilevel randomized trials (Unpublished doctoral dissertation). University of Cincinnati, Cincinnati, OH.

Examples

```
# unconstrained optimal design #---------
  myod1 <- od.4(icc2 = 0.2, icc3 = 0.1, icc4 = 0.05,
               r12 = 0.5, r22 = 0.5, r32 = 0.5, r42 = 0.5,
               c1 = 1, c2 = 5, c3 = 25, c4 = 125,
               c1t = 1, c2t = 50, c3t = 250, c4t = 2500,
               varlim = c(0, 0.01))
  myod1$out # output
# plots by p and K
  myod1 <- od.4(icc2 = 0.2, icc3 = 0.1, icc4 = 0.05,
               r12 = 0.5, r22 = 0.5, r32 = 0.5, r42 = 0.5,
               c1 = 1, c2 = 5, c3 = 25, c4 = 125,
               c1t = 1, c2t = 50, c3t = 250, c4t = 2500,
               varlim = c(0, 0.01), plot.by = list(p = 'p', K = 'K'))

# constrained optimal design with p = 0.5 #---------
  myod2 <- od.4(icc2 = 0.2, icc3 = 0.1, icc4 = 0.05, p = 0.5,
               r12 = 0.5, r22 = 0.5, r32 = 0.5, r42 = 0.5,
               c1 = 1, c2 = 5, c3 = 25, c4 = 125,
               c1t = 1, c2t = 50, c3t = 250, c4t = 2500,
               varlim = c(0, 0.01))
  myod2$out
# relative efficiency (RE)
  myre <- re(od = myod1, subod= myod2)
  myre$re # RE = 0.78

# constrained optimal design with K = 20 #---------
  myod3 <- od.4(icc2 = 0.2, icc3 = 0.1, icc4 = 0.05,  K = 20,
               r12 = 0.5, r22 = 0.5, r32 = 0.5, r42 = 0.5,
               c1 = 1, c2 = 5, c3 = 25, c4 = 125,
               c1t = 1, c2t = 50, c3t = 250, c4t = 2500,
               varlim = c(0, 0.01))
```

```
  myod3$out
# relative efficiency (RE)
  myre <- re(od = myod1, subod= myod3)
  myre$re # RE = 0.67

# constrained n, J, K and p, no calculation performed #---------
  myod4 <- od.4(icc2 = 0.2, icc3 = 0.1, icc4 = 0.05,
              r12 = 0.5, n = 10, J = 10, K = 20, p = 0.5,
              r22 = 0.5, r32 = 0.5, r42 = 0.5,
              c1 = 1, c2 = 5, c3 = 25, c4 = 125,
              c1t = 1, c2t = 50, c3t = 250, c4t = 2500,
              varlim = c(0, 0.01))
  myod4$out
# relative efficiency (RE)
  myre <- re(od = myod1, subod= myod4)
  myre$re # RE = 0.27
```

`od.4m` *Optimal sample allocation calculation for four-level multisite randomized trials*

Description

The optimal design of four-level multisite randomized trials (MRTs) is to choose the sample allocation that minimizes the variance of treatment effect under fixed budget and cost structure. The optimal design parameters include the level-1 sample size per level-2 unit (`n`), the level-2 sample size per level-3 unit (`J`), the level-3 sample size per level-4 unit (`K`), and the proportion of level-3 units to be assigned to treatment (`p`). This function solves the optimal `n`, `J`, `K` and/or `p` with and without constraints.

Usage

```
od.4m(n = NULL, J = NULL, K = NULL, p = NULL, icc2 = NULL,
  icc3 = NULL, icc4 = NULL, r12 = NULL, r22 = NULL, r32 = NULL,
  r42m = NULL, c1 = NULL, c2 = NULL, c3 = NULL, c4 = NULL,
  c1t = NULL, c2t = NULL, c3t = NULL, omega = NULL, m = NULL,
  plots = TRUE, plot.by = NULL, nlim = NULL, Jlim = NULL,
  Klim = NULL, plim = NULL, varlim = NULL, nlab = NULL,
  Jlab = NULL, Klab = NULL, plab = NULL, varlab = NULL,
  vartitle = NULL, verbose = TRUE, iter = 100, tol = 1e-10)
```

Arguments

`n`	the level-1 sample size per level-2 unit.
`J`	the level-2 sample size per level-3 unit.
`K`	the level-3 sample size per level-4 unit.
`p`	the proportion of level-3 units to be assigned to treatment.

`icc2`	the unconditional intraclass correlation coefficient (ICC) at level 2.
`icc3`	the unconditional intraclass correlation coefficient (ICC) at level 3.
`icc4`	the unconditional intraclass correlation coefficient (ICC) at level 4.
`r12`	the proportion of level-1 variance explained by covariates.
`r22`	the proportion of level-2 variance explained by covariates.
`r32`	the proportion of level-3 variance explained by covariates.
`r42m`	the proportion of variance of site-specific treatment effect explained by covariates.
`c1`	the cost of sampling one level-1 unit in control condition.
`c2`	the cost of sampling one level-2 unit in control condition.
`c3`	the cost of sampling one level-3 unit in control condition.
`c4`	the cost of sampling one level-4 unit.
`c1t`	the cost of sampling one level-1 unit in treatment condition.
`c2t`	the cost of sampling one level-2 unit in treatment condition.
`c3t`	the cost of sampling one level-3 unit in treatment condition.
`omega`	the standardized variance of site-specific treatment effect
`m`	total budget, default is the total costs of sampling 60 level-4 units.
`plots`	logical, provide variance plots if TRUE, otherwise not; default value is TRUE.
`plot.by`	specify variance plot by `n`, `J`, `K` and/or `p`; default value is plot.by = list(n = "n", J = "J", K = 'K', p = "p").
`nlim`	the plot range for n, default value is c(2, 50).
`Jlim`	the plot range for J, default value is c(2, 50).
`Klim`	the plot range for K, default value is c(2, 50).
`plim`	the plot range for p, default value is c(0, 1).
`varlim`	the plot range for variance, default value is c(0, 0.05).
`nlab`	the plot label for `n`, default value is "Level-1 Sample Size: n".
`Jlab`	the plot label for `J`, default value is "Level-2 Sample Size: J".
`Klab`	the plot label for `K`, default value is "Level-3 Sample Size: K".
`plab`	the plot label for `p`, default value is "Proportion Level-3 Units in Treatment: p".
`varlab`	the plot label for variance, default value is "Variance".
`vartitle`	the title of variance plot, default value is NULL.
`verbose`	logical; print the values of `n`, `J`, `K`, and `p` if TRUE, otherwise not; default value is TRUE.
`iter`	number of iterations; default value is 100.
`tol`	tolerance for convergence; default value is 1e-10.

Value

unconstrained or constrained optimal sample allocation (`n`, `J`, `K`, and `p`). The function also returns the variance of treatment effect, function name, design type, and parameters used in the calculation.

References

Shen, Z. (in progress). Using optimal sample allocation to improve statistical precision and design efficiency for multilevel randomized trials (Unpublished doctoral dissertation). University of Cincinnati, Cincinnati, OH.

Examples

```
# unconstrained optimal design #---------
  myod1 <- od.4m(icc2 = 0.2, icc3 = 0.1, icc4 = 0.05, omega = 0.02,
              r12 = 0.5, r22 = 0.5, r32 = 0.5, r42m = 0.5,
              c1 = 1, c2 = 5, c3 = 25,
              c1t = 1, c2t = 50, c3t = 250, c4 = 500,
              varlim = c(0, 0.005))
  myod1$out # output
# plots by p and K
  myod1 <- od.4m(icc2 = 0.2, icc3 = 0.1, icc4 = 0.05, omega = 0.02,
              r12 = 0.5, r22 = 0.5, r32 = 0.5, r42m = 0.5,
              c1 = 1, c2 = 5, c3 = 25,
              c1t = 1, c2t = 50, c3t = 250, c4 = 500,
              varlim = c(0, 0.005), plot.by = list(p = 'p', K = 'K'))

# constrained optimal design with p = 0.5 #---------
  myod2 <- od.4m(icc2 = 0.2, icc3 = 0.1, icc4 = 0.05, omega = 0.02,
              r12 = 0.5, r22 = 0.5, r32 = 0.5, r42m = 0.5,
              c1 = 1, c2 = 5, c3 = 25,
              c1t = 1, c2t = 50, c3t = 250, c4 = 500,
              varlim = c(0, 0.005), p = 0.5)
  myod2$out
# relative efficiency (RE)
  myre <- re(od = myod1, subod= myod2)
  myre$re # RE = 0.88

# constrained optimal design with J = 20 #---------
  myod3 <- od.4m(icc2 = 0.2, icc3 = 0.1, icc4 = 0.05, omega = 0.02,
              r12 = 0.5, r22 = 0.5, r32 = 0.5, r42m = 0.5,
              c1 = 1, c2 = 5, c3 = 25,
              c1t = 1, c2t = 50, c3t = 250, c4 = 500,
              varlim = c(0, 0.005), J = 20)
  myod3$out
# relative efficiency (RE)
  myre <- re(od = myod1, subod= myod3)
  myre$re # RE = 0.58

# constrained n, J, K and p, no calculation performed #---------
  myod4 <- od.4m(icc2 = 0.2, icc3 = 0.1, icc4 = 0.05, omega = 0.02,
              r12 = 0.5, r22 = 0.5, r32 = 0.5, r42m = 0.5,
              c1 = 1, c2 = 5, c3 = 25,
              c1t = 1, c2t = 50, c3t = 250, c4 = 500,
              varlim = c(0, 0.005), p = 0.5, n = 15, J = 20, K = 5)
  myod4$out
# relative efficiency (RE)
  myre <- re(od = myod1, subod= myod4)
```

```
myre$re # RE = 0.46
```

---

power.1 *Budget and/or sample size, power, MDES calculation for individual randomized controlled trials*

Description

This function can calculate required budget for desired power, power or minimum detectable effect size (MDES) under fixed budget for individual randomized controlled trials (RCTs). It also can perform conventional power analyses (e.g., required sample size, power, and MDES calculation).

Usage

```
power.1(cost.model = TRUE, expr = NULL, constraint = NULL,
  sig.level = 0.05, two.tailed = TRUE, d = NULL, power = NULL,
  m = NULL, n = NULL, p = NULL, r12 = NULL, q = NULL,
  c1 = NULL, c1t = NULL, dlim = NULL, powerlim = NULL,
  nlim = NULL, mlim = NULL, rounded = TRUE)
```

Arguments

`cost.model`	logical; power analyses accommodating costs and budget (e.g., required budget for desired power, power/MDES under fixed budget) if TRUE, otherwise conventional power analyses (e.g., required sample size, power, or MDES calculation); default value is TRUE.
`expr`	returned object from function `od.1`; default value is NULL; if `expr` is specified, parameter values of `r12`, `c1`, `c1t`, and `p` used or solved in function `od.1` will be passed to the current function; only the value of `p` that specified or solved in function `od.1` can be overwritten if `constraint` is specified.
`constraint`	specify the constrained value of `p` in list format to overwrite that from `expr`; default value is NULL.
`sig.level`	significance level or type I error rate, default value is 0.05.
`two.tailed`	logical; two-tailed tests if TRUE, otherwise one-tailed tests; default value is TRUE.
`d`	effect size.
`power`	statistical power.
`m`	total budget.
`n`	the total sample size.
`p`	the proportion of individuals to be assigned to treatment.
`r12`	the proportion of outcome variance explained by covariates.
`q`	the number of covariates.
`c1`	the cost of sampling one unit in control condition.

`c1t` the cost of sampling one unit in treatment condition.

`dlim` the range for searching the root of effect size (`d`) numerically, default value is c(0, 5).

`powerlim` the range for searching the root of power (`power`) numerically, default value is c(1e-10, 1 - 1e-10).

`nlim` the range for searching the root of sample size (`n`) numerically, default value is c(4, 10e10)

`mlim` the range for searching the root of budget (`m`) numerically, default value is the costs sampling `nlim` units across treatment conditions or c(4 * ncost, 10e10 * ncost) with ncost = ((1 - p) * c1 + p * c1t)

`rounded` logical; round `p` that is from functions `od.1` to two decimal places if TRUE, otherwise no rounding; default value is TRUE.

Value

Required budget (or required sample size), statistical power, or MDES depending on the specification of parameters. The function also returns the function name, design type, and parameters used in the calculation.

References

Shen, Z. (in progress). Using optimal sample allocation to improve statistical precision and design efficiency for multilevel randomized trials (Unpublished doctoral dissertation). University of Cincinnati, Cincinnati, OH.

Examples

```
# unconstrained optimal design
  myod1 <- od.1(r12 = 0.5, c1 = 1, c1t = 5, varlim = c(0, 0.2))
  myod1$out   # p = 0.31

# ------- power analyses by default considering costs and budget -------
# required budget and sample size
  mym.1 <- power.1(expr = myod1, d = 0.2, q = 1, power = 0.8)
  mym.1$out  # m = 1032 n = 461
  # mym.1$par  # parameters and their values used for the function
# or equivalently, specify every argument in the function
  mym.1 <- power.1(d = 0.2, power = 0.8, c1 = 1, c1t = 5,
                   r12 = 0.5, p = 0.31, q = 1)
# required budget and sample size with constrained p
  mym.2 <- power.1(expr = myod1, d = 0.2, q = 1, power = 0.8,
                constraint = list(p = 0.5))
  mym.2$out  # m = 1183, n = 394

# Power calculation
  mypower <- power.1(expr = myod1, q = 1, d = 0.2, m = 1032)
  mypower$out  # power = 0.80
# Power calculation under constrained p (p = 0.5)
  mypower.1 <- power.1(expr = myod1, q = 1, d = 0.2, m = 1032,
                constraint = list(p = 0.5))
```

```
  mypower.1$out  # power = 0.74

# MDES calculation
  mymdes <- power.1(expr = myod1, q = 1, power = 0.80, m = 1032)
  mymdes$out  # d = 0.20


# ------- conventional power analyses with cost.model = FALSE-------
# Required sample size n
  myn <- power.1(cost.model = FALSE, expr = myod1, d = 0.2, q = 1, power = 0.8)
  myn$out  # n = 461
  # myn$par  # parameters and their values used for the function
# or equivalently, specify every argument in the function
  myn <- power.1(cost.model = FALSE, d = 0.2, power = 0.8,
                 r12 = 0.5, p = 0.31, q = 1)

# Power calculation
  mypower1 <- power.1(cost.model = FALSE, expr = myod1, n = 461, d = 0.2, q = 1)
  mypower1$out  # power = 0.80

# MDES calculation
  mymdes1 <- power.1(cost.model = FALSE, expr = myod1, n = 461, power = 0.8, q = 1)
  mymdes1$out  # d = 0.20
```

---

power.2 *Budget and/or sample size, power, MDES calculation for two-level CRTs*

Description

This function can calculate required budget for desired power, power or minimum detectable effect size (MDES) under fixed budget for two-level cluster randomized trials (CRTs). It also can perform conventional power analyses (e.g., required sample size, power, and MDES calculation).

Usage

```
power.2(cost.model = TRUE, expr = NULL, constraint = NULL,
  sig.level = 0.05, two.tailed = TRUE, d = NULL, power = NULL,
  m = NULL, n = NULL, J = NULL, p = NULL, icc = NULL,
  r12 = NULL, r22 = NULL, q = NULL, c1 = NULL, c2 = NULL,
  c1t = NULL, c2t = NULL, dlim = NULL, powerlim = NULL,
  Jlim = NULL, mlim = NULL, rounded = TRUE)
```

Arguments

`cost.model` logical; power analyses accommodating costs and budget (e.g., required budget for desired power, power/MDES under fixed budget) if TRUE, otherwise conventional power analyses (e.g., required sample size, power, or MDES calculation); default value is TRUE.

expr	returned object from function od.2; default is NULL; if expr is specified, parameter values of icc, r12, r22, c1, c2, c1t, c2t, n, and p used or solved in function od.2 will be passed to the current function; only the values of n and p that specified or solved in function od.2 can be overwritten if constraint is specified.
constraint	specify the constrained values of n and/or p in list format to overwrite those from expr; default is NULL.
sig.level	significance level or type I error rate, default value is 0.05.
two.tailed	logical; two-tailed tests if TRUE, otherwise one-tailed tests; default value is TRUE.
d	effect size.
power	statistical power.
m	total budget.
n	the level-1 sample size per level-2 unit.
J	the total level-2 sample size.
p	the proportion of level-2 clusters/units to be assigned to treatment.
icc	the unconditional intraclass correlation coefficient (ICC) in population or in each treatment condition.
r12	the proportion of level-1 variance explained by covariates.
r22	the proportion of level-2 variance explained by covariates.
q	the number of level-2 covariates.
c1	the cost of sampling one level-1 unit in control condition.
c2	the cost of sampling one level-2 unit in control condition.
c1t	the cost of sampling one level-1 unit in treatment condition.
c2t	the cost of sampling one level-2 unit in treatment condition.
dlim	the range for searching the root of effect size (d) numerically, default value is c(0, 5).
powerlim	the range for searching the root of power (power) numerically, default value is c(1e-10, 1 - 1e-10).
Jlim	the range for searching the root of level-2 sample size (J) numerically, default is c(4, 10e10)
mlim	the range for searching the root of budget (m) numerically, default is the costs sampling Jlim level-2 units across treatment conditions or c(4 * Jcost , 10e10 * Jcost), with Jcost = ((1 - p) * (c1 * n + c2) + p * (c1t * n + c2t))
rounded	logical; round n and p that are from functions od.2 to integer and two decimal places, respectively if TRUE, otherwise no rounding; default value is TRUE.

Value

Required budget (and/or required level-2 sample size), statistical power, or MDES depending on the specification of parameters. The function also returns the function name, design type, and parameters used in the calculation.

References

Shen, Z., & Kelcey, B. (2018, April). Optimal design of cluster randomized trials under condition- and unit-specific cost structures. Roundtable discussion presented at American Educational Research Association (AERA) annual conference, New York City, NY;

Shen, Z., & Kelcey, B. (revise & resubmit). Optimal sample allocation under unequal costs in cluster-randomized trials. Journal of Educational and Behavioral Statistics.

Shen, Z. (in progress). Using optimal sample allocation to improve statistical precision and design efficiency for multilevel randomized trials (Unpublished doctoral dissertation). University of Cincinnati, Cincinnati, OH.

Examples

```
# unconstrained optimal design
  myod1 <- od.2(icc = 0.2, r12 = 0.5, r22 = 0.5, c1 = 1, c2 = 5, c1t = 1, c2t = 50)
  myod1$out   # n = 8.9, p = 0.33

# ------- power analyses by default considering costs and budget -------
# required budget and sample size
  mym.1 <- power.2(expr = myod1, d = 0.2, q = 1, power = 0.8)
  mym.1$out  # m = 3755, J = 130.2
  #mym.1$par  # parameters and their values used for the function
# or equivalently, specify every argument in the function
  mym.1 <- power.2(d = 0.2, power = 0.8, icc = 0.2,
                c1 = 1, c2 = 5, c1t = 1, c2t = 50,
                 r12 = 0.5, r22 = 0.5, n = 9, p = 0.33, q = 1)
# required budget and sample size with constrained p
  mym.2 <- power.2(expr = myod1, d = 0.2, q = 1, power = 0.8,
              constraint = list(p = 0.5))
  mym.2$out  # m = 4210, J = 115.3
# required budget and sample size with constrained p and n
  mym.3 <- power.2(expr = myod1, d = 0.2, q = 1, power = 0.8,
              constraint = list(p = 0.5, n = 20))
  mym.3$out  # m = 4568, J = 96.2

# Power calculation
  mypower <- power.2(expr = myod1, q = 1, d = 0.2, m = 3755)
  mypower$out  # power = 0.80
# Power calculation under constrained p (p = 0.5)
  mypower.1 <- power.2(expr = myod1, q = 1, d = 0.2, m = 3755,
              constraint = list(p = 0.5))
  mypower.1$out  # power = 0.75

# MDES calculation
  mymdes <- power.2(expr = myod1, q = 1, power = 0.80, m = 3755)
  mymdes$out  # d = 0.20


# ------- conventional power analyses with cost.model = FALSE-------
# Required J
  myJ <- power.2(cost.model = FALSE, expr = myod1, d = 0.2, q = 1, power = 0.8)
  myJ$out  # J = 130.2
```

```
  #myJ$par  # parameters and their values used for the function
# or equivalently, specify every argument in the function
  myJ <- power.2(cost.model = FALSE, d = 0.2, power = 0.8, icc = 0.2,
                 r12 = 0.5, r22 = 0.5, n = 9, p = 0.33, q = 1)

# Power calculation
  mypower1 <- power.2(cost.model = FALSE, expr = myod1, J = 130, d = 0.2, q = 1)
  mypower1$out  # power = 0.80

# MDES calculation
  mymdes1 <- power.2(cost.model = FALSE, expr = myod1, J = 130, power = 0.8, q = 1)
  mymdes1$out  # d = 0.20
```

`power.2m` *Budget and/or sample size, power, MDES calculation for two-level multisite randomized trials*

Description

This function can calculate required budget for desired power, power or minimum detectable effect size (MDES) under fixed budget for two-level multisite randomized trials (MRTs). It also can perform conventional power analyses (e.g., required sample size, power, and MDES calculation).

Usage

```
power.2m(cost.model = TRUE, expr = NULL, constraint = NULL,
  sig.level = 0.05, two.tailed = TRUE, d = NULL, power = NULL,
  m = NULL, n = NULL, J = NULL, p = NULL, icc = NULL,
  r12 = NULL, r22m = NULL, q1 = NULL, q2 = NULL, c1 = NULL,
  c2 = NULL, c1t = NULL, omega = NULL, dlim = NULL,
  powerlim = NULL, Jlim = NULL, mlim = NULL, rounded = TRUE)
```

Arguments

`cost.model`	logical; power analyses accommodating costs and budget (e.g., required budget for desired power, power/MDES under fixed budget) if TRUE, otherwise conventional power analyses (e.g., required sample size, power, or MDES calculation); default value is TRUE.
`expr`	returned objects from function `od.2m`; default is NULL; if `expr` is specified, parameter values of `icc`, `r12`, `r22m`, `c1`, `c2`, `c1t`, `p`, and `n` used or solved in function `od.2m` will be passed to current function; only the values of `p` and `n` that specified or solved in function `od.2m` can be overwritten if `constraint` is specified.
`constraint`	specify the constrained values of `p` and/or `n` in list format to overwrite those from `expr`; default value is NULL.
`sig.level`	significance level or type I error rate, default value is 0.05.

two.tailed	logical; two-tailed tests if TRUE, otherwise one-tailed tests; default value is TRUE.
d	effect size.
power	statistical power.
m	total budget.
n	the level-1 sample size per level-2 unit.
J	the level-2 sample size per level-3 unit.
p	the proportion of level-1 units to be assigned to treatment.
icc	the unconditional intraclass correlation coefficient (ICC) in population or in each treatment condition.
r12	the proportion of level-1 variance explained by covariates.
r22m	the proportion of variance of site-specific treatment effect explained by covariates.
q1	the number of covariates at level 1.
q2	the number of covariates at level 2.
c1	the cost of sampling one level-1 unit in control condition.
c2	the cost of sampling one level-2 unit.
c1t	the cost of sampling one level-1 unit in treatment condition.
omega	the standardized variance of site-specific treatment effect
dlim	the range for searching the root of effect size (d) numerically, default value is c(0, 5).
powerlim	the range for searching the root of power (power) numerically, default value is c(1e-10, 1 - 1e-10).
Jlim	the range for searching the root of level-2 sample size (J) numerically, default is c(4, 10e10)
mlim	the range for searching the root of budget (m) numerically, default is the costs sampling Jlim level-2 units or c(4 * Jcost, 1e+10 * Jcost) with Jcost = (1 - p) * c1 * n + p * c1t * n + c2.
rounded	logical; round the values of p, n/J/K that are from functions od.4 to two decimal places and integer, respectively if TRUE, otherwise no rounding; default value is TRUE.

Value

Required budget (and/or required level-2 sample size), statistical power, or MDES depending on the specification of parameters. The function also returns the function name, design type, and parameters used in the calculation.

References

Shen, Z. (in progress). Using optimal sample allocation to improve statistical precision and design efficiency for multilevel randomized trials (Unpublished doctoral dissertation). University of Cincinnati, Cincinnati, OH.

Examples

```
# unconstrained optimal design #---------
  myod1 <- od.2m(icc = 0.2, omega = 0.02, r12 = 0.5, r22m = 0.5,
              c1 = 1, c2 = 10, c1t = 10,
              varlim = c(0, 0.005))
  myod1$out # n = 20, p = 0.37

# ------- power analyses by default considering costs and budget -------
# required budget and sample size
  mym.1 <- power.2m(expr = myod1, d = 0.2, q1 = 1, q2 = 1, power = 0.8)
  mym.1$out  # m = 612, J = 6.3
  # mym.1$par  # parameters and their values used for the function
# or equivalently, specify every argument in the function
  mym.1 <- power.2m(d = 0.2, power = 0.8, q1 = 1, q2 = 1,
                 icc = 0.2, omega = 0.02, r12 = 0.5, r22m = 0.5,
                 c1 = 1, c2 = 10, c1t = 10,
                 n = 20, p = 0.37)
# required budget and sample size with constrained p
  mym.2 <- power.2m(expr = myod1, d = 0.2, q1 = 1, q2 = 1, power = 0.8,
                 constraint = list(p = 0.5))
  mym.2$out  # m = 726, J = 6.1
# required budget and sample size with constrained p and n
  mym.3 <- power.2m(expr = myod1, d = 0.2, q1 = 1, q2 = 1, power = 0.8,
                 constraint = list(p = 0.5, n = 5))
  mym.3$out  # m = 702, J = 18.7

# Power calculation
  mypower <- power.2m(expr = myod1, q1 = 1, q2 = 1, d = 0.2, m = 612)
  mypower$out  # power = 0.80
# Power calculation under constrained p (p = 0.5)
  mypower.1 <- power.2m(expr = myod1, q1 = 1, q2 = 1, d = 0.2, m = 612,
                 constraint = list(p = 0.5))
  mypower.1$out  # power = 0.73

# MDES calculation
  mymdes <- power.2m(expr = myod1, q1 = 1, q2 = 1, power = 0.80, m = 612)
  mymdes$out  # d = 0.20


# ------- conventional power analyses with cost.model = FALSE-------
# Required sample size
  myJ <- power.2m(cost.model = FALSE, expr = myod1, d = 0.2, q1 = 1, q2 = 1, power = 0.8)
  myJ$out  # J = 6.3
  # myL$par  # parameters and their values used for the function
# or equivalently, specify every argument in the function
  myJ <- power.2m(cost.model = FALSE, d = 0.2, power = 0.8, q1 = 1, q2 = 1,
                 icc = 0.2, omega = 0.02, r12 = 0.5, r22m = 0.5,
                 c1 = 1, c2 = 10, c1t = 10,
                 n = 20, p = 0.37)

# Power calculation
  mypower1 <- power.2m(cost.model = FALSE, expr = myod1, J = 6.3, d = 0.2, q1 = 1, q2 = 1)
```

```
    mypower1$out  # power = 0.80

  # MDES calculation
   mymdes1 <- power.2m(cost.model = FALSE, expr = myod1, J = 6.3, power = 0.8, q1 = 1, q2 = 1)
    mymdes1$out  # d = 0.20
```

---

power.3 *Budget and/or sample size, power, MDES calculation for three-level CRTs*

Description

This function can calculate required budget for desired power, power or minimum detectable effect size (MDES) under fixed budget for three-level cluster randomized trials (CRTs). It also can perform conventional power analyses (e.g., required sample size, power, and MDES calculation).

Usage

```
power.3(cost.model = TRUE, expr = NULL, constraint = NULL,
  sig.level = 0.05, two.tailed = TRUE, d = NULL, power = NULL,
  m = NULL, n = NULL, J = NULL, K = NULL, p = NULL,
  icc2 = NULL, icc3 = NULL, r12 = NULL, r22 = NULL, r32 = NULL,
  q = NULL, c1 = NULL, c2 = NULL, c3 = NULL, c1t = NULL,
  c2t = NULL, c3t = NULL, dlim = NULL, powerlim = NULL,
  Klim = NULL, mlim = NULL, rounded = TRUE)
```

Arguments

cost.model	logical; power analyses accommodating costs and budget (e.g., required budget for desired power, power/MDES under fixed budget) if TRUE, otherwise conventional power analyses (e.g., required sample size, power, or MDES calculation); default value is TRUE.
expr	returned objects from function od.3; default is NULL; if expr is specified, parameter values of icc2, icc3, r12, r22, r32, c1, c2, c3, c1t, c2t, c3t, p, n, and J used or solved in function od.3 will be passed to the current function; only the values of p, n, and/or J that specified or solved in function od.3 can be overwritten if constraint is specified.
constraint	specify the constrained values of p, n, and/or J in list format to overwrite those from expr; default is NULL.
sig.level	significance level or type I error rate, default value is 0.05.
two.tailed	logical; two-tailed tests if TRUE, otherwise one-tailed tests; default value is TRUE.
d	effect size.
power	statistical power.
m	total budget.

n	the level-1 sample size per level-2 unit.
J	the level-2 sample size per level-3 unit.
K	the total level-3 sample size.
p	the proportion of level-3 clusters/units assigned to treatment.
icc2	the unconditional intraclass correlation coefficient (ICC) at level 2.
icc3	the unconditional intraclass correlation coefficient (ICC) at level 3.
r12	the proportion of level-1 variance explained by covariates.
r22	the proportion of level-2 variance explained by covariates.
r32	the proportion of level-3 variance explained by covariates.
q	the number of covariates at level 3.
c1	the cost of sampling one level-1 unit in control condition.
c2	the cost of sampling one level-2 unit in control condition.
c3	the cost of sampling one level-3 unit in control condition.
c1t	the cost of sampling one level-1 unit in treatment condition.
c2t	the cost of sampling one level-2 unit in treatment condition.
c3t	the cost of sampling one level-3 unit in treatment condition.
dlim	the range for searching the root of effect size (d) numerically, default value is c(0, 5).
powerlim	the range for searching the root of power (power) numerically, default value is c(1e-10, 1 - 1e-10).
Klim	the range for searching the root of level-3 sample size (K) numerically, default value is c(4, 1e+10)
mlim	the range for searching the root of budget (m) numerically, default value is the costs sampling Klim level-3 units across treatment conditions or c(4 * Kcost, 1e+10 * Kcost) with Kcost = ((1 - p) * (c1 * n * J + c2 * J + c3) + p * (c1t * n * J + c2t * J + c3t))
rounded	logical; round the values of p, n/J that are from functions od.3 to two decimal places and integer, respectively if TRUE, otherwise no rounding; default value is TRUE.

Value

Required budget (and/or required level-3 sample size), statistical power, or MDES depending on the specification of parameters. The function also returns the function name, design type, and parameters used in the calculation.

References

Shen, Z., & Kelcey, B. (revise & resubmit). Optimal sample allocation under unequal costs in cluster-randomized trials. Journal of Educational and Behavioral Statistics.

Shen, Z. (in progress). Using optimal sample allocation to improve statistical precision and design efficiency for multilevel randomized trials (Unpublished doctoral dissertation). University of Cincinnati, Cincinnati, OH.

Examples

```
# unconstrained optimal design
  myod1 <- od.3(icc2 = 0.2, icc3 = 0.1, r12 = 0.5, r22 = 0.5, r32 = 0.5,
              c1 = 1, c2 = 5, c3 = 25, c1t = 1, c2t = 50, c3t = 250)
  myod1$out # output  # n = 7.9, J = 3.2, p = 0.28

# ------- power analyses by default considering costs and budget -------
# required budget and sample size
  mym.1 <- power.3(expr = myod1, d = 0.2, q = 1, power = 0.8)
  mym.1$out  # m = 16032, K = 97.3
  #mym.1$par  # parameters and their values used for the function
# or equivalently, specify every argument in the function
  mym.1 <- power.3(d = 0.2, power = 0.8, q = 1,
                 icc2 = 0.2, icc3 = 0.1, r12 = 0.5, r22 = 0.5, r32 = 0.5,
                 c1 = 1, c2 = 5, c3 = 25, c1t = 1, c2t = 50, c3t = 250,
                 n = 8, J = 3, p = 0.28)
# required budget and sample size with constrained p
  mym.2 <- power.3(expr = myod1, d = 0.2, q = 1, power = 0.8,
                 constraint = list(p = 0.5))
  mym.2$out  # m = 19239, K = 78.8
# required budget and sample size with constrained p and J
  mym.3 <- power.3(expr = myod1, d = 0.2, q = 1, power = 0.8,
                 constraint = list(p = 0.5, J = 20))
  mym.3$out  # m = 39774, K = 46.9

# Power calculation
  mypower <- power.3(expr = myod1, q = 1, d = 0.2, m = 16032)
  mypower$out  # power = 0.80
# Power calculation under constrained p (p = 0.5)
  mypower.1 <- power.3(expr = myod1, q = 1, d = 0.2, m = 16032,
                 constraint = list(p = 0.5))
  mypower.1$out  # power = 0.72

# MDES calculation
  mymdes <- power.3(expr = myod1, q = 1, power = 0.80, m = 16032)
  mymdes$out  # d = 0.20


# ------- conventional power analyses with cost.model = FALSE-------
# Required sample size
  myK <- power.3(cost.model = FALSE, expr = myod1, d = 0.2, q = 1, power = 0.8)
  myK$out  # K = 97.3
  #myK$par  # parameters and their values used for the function
# or equivalently, specify every argument in the function
  myK <- power.3(cost.model = FALSE, d = 0.2, power = 0.8, q = 1,
                  icc2 = 0.2, icc3 = 0.1, r12 = 0.5, r22 = 0.5, r32 = 0.5,
                  n = 8, J = 3, p = 0.28)

# Power calculation
  mypower1 <- power.3(cost.model = FALSE, expr = myod1, K = 97, d = 0.2, q = 1)
  mypower1$out  # power = 0.80
```

```
# MDES calculation
  mymdes1 <- power.3(cost.model = FALSE, expr = myod1, K = 97, power = 0.8, q = 1)
  mymdes1$out  # d = 0.20
```

---

power.3m *Budget and/or sample size, power, MDES calculation for three-level multisite randomized trials*

Description

This function can calculate required budget for desired power, power or minimum detectable effect size (MDES) under fixed budget for three-level multisite randomized trials (MRTs). It also can perform conventional power analyses (e.g., required sample size, power, and MDES calculation).

Usage

```
power.3m(cost.model = TRUE, expr = NULL, constraint = NULL,
  sig.level = 0.05, two.tailed = TRUE, d = NULL, power = NULL,
  m = NULL, n = NULL, J = NULL, K = NULL, p = NULL,
  icc2 = NULL, icc3 = NULL, r12 = NULL, r22 = NULL, r32m = NULL,
  q2 = NULL, q3 = NULL, c1 = NULL, c2 = NULL, c3 = NULL,
  c1t = NULL, c2t = NULL, omega = NULL, dlim = NULL,
  powerlim = NULL, Klim = NULL, mlim = NULL, rounded = TRUE)
```

Arguments

`cost.model`	logical; power analyses accommodating costs and budget (e.g., required budget for desired power, power/MDES under fixed budget) if TRUE, otherwise conventional power analyses (e.g., required sample size, power, or MDES calculation); default value is TRUE.
`expr`	returned objects from function `od.3m`; default is NULL; if `expr` is specified, parameter values of `icc2`, `icc3`, `r12`, `r22`, `r32m`, `c1`, `c2`, `c3`, `c1t`, `c2t`, `p`, `n`, and `J` used or solved in function `od.3m` will be passed to current function; only the values of `p`, `n`, and/or `J` that specified or solved in function `od.3m` can be overwritten if `constraint` is specified.
`constraint`	specify the constrained values of `p`, `n`, and/or `J`, in list format to overwrite those from `expr`; default value is NULL.
`sig.level`	significance level or type I error rate, default value is 0.05.
`two.tailed`	logical; two-tailed tests if TRUE, otherwise one-tailed tests; default value is TRUE.
`d`	effect size.
`power`	statistical power.
`m`	total budget.
`n`	the level-1 sample size per level-2 unit.

J	the level-2 sample size per level-3 unit.
K	the level-3 sample size per level-4 unit.
p	the proportion of level-2 units to be assigned to treatment.
icc2	the unconditional intraclass correlation coefficient (ICC) at level 2.
icc3	the unconditional intraclass correlation coefficient (ICC) at level 3.
r12	the proportion of level-1 variance explained by covariates.
r22	the proportion of level-2 variance explained by covariates.
r32m	the proportion of variance of site-specific treatment effect explained by covariates.
q2	the number of covariates at level 2.
q3	the number of covariates at level 3.
c1	the cost of sampling one level-1 unit in control condition.
c2	the cost of sampling one level-2 unit in control condition.
c3	the cost of sampling one level-3 unit.
c1t	the cost of sampling one level-1 unit in treatment condition.
c2t	the cost of sampling one level-2 unit in treatment condition.
omega	the standardized variance of site-specific treatment effect
dlim	the range for searching the root of effect size (d) numerically, default value is c(0, 5).
powerlim	the range for searching the root of power (power) numerically, default value is c(1e-10, 1 - 1e-10).
Klim	the range for searching the root of level-3 sample size (K) numerically, default value is c(4, 1e+10)
mlim	the range for searching the root of budget (m) numerically, default is the costs sampling Klim level-3 units or c(4 * Kcost, 1e+10 * Kcost) with Kcost = ((1 - p) * (c1 * n * J + c2 * J) + p * (c1t * n * J + c2t * J) + c3.
rounded	logical; round the values of p, n/J/K that are from functions od.4 to two decimal places and integer, respectively if TRUE, otherwise no rounding; default value is TRUE.

Value

Required budget (and/or required level-3 sample size), statistical power, or MDES depending on the specification of parameters. The function also returns the function name, design type, and parameters used in the calculation.

References

Shen, Z. (in progress). Using optimal sample allocation to improve statistical precision and design efficiency for multilevel randomized trials (Unpublished doctoral dissertation). University of Cincinnati, Cincinnati, OH.

Examples

```
# unconstrained optimal design #---------
  myod1 <- od.3m(icc2 = 0.2, icc3 = 0.1, omega = 0.02,
              r12 = 0.5, r22 = 0.5, r32m = 0.5,
              c1 = 1, c2 = 5,
              c1t = 1, c2t = 200, c3 = 200,
              varlim = c(0, 0.005))
  myod1$out # n = 13, J = 15, p = 0.23

# ------- power analyses by default considering costs and budget -------
# required budget and sample size
  mym.1 <- power.3m(expr = myod1, d = 0.2, q2 = 1, q3 = 1, power = 0.8)
  mym.1$out  # m = 13177, K = 11.5
  # mym.1$par  # parameters and their values used for the function
# or equivalently, specify every argument in the function
  mym.1 <- power.3m(d = 0.2, power = 0.8, q2 = 1, q3 = 1,
                 icc2 = 0.2, icc3 = 0.1, omega = 0.02,
                  r12 = 0.5, r22 = 0.5, r32m = 0.5,
                  c1 = 1, c2 = 5,
                  c1t = 1, c2t = 200, c3 = 200,
                  n = 13, J = 15, p = 0.23)
# required budget and sample size with constrained p
  mym.2 <- power.3m(expr = myod1, d = 0.2, q2 = 1, q3 = 1, power = 0.8,
                 constraint = list(p = 0.5))
  mym.2$out  # m = 17026, K = 8.8
# required budget and sample size with constrained p and n
  mym.3 <- power.3m(expr = myod1, d = 0.2, q2 = 1, q3 = 1, power = 0.8,
                 constraint = list(p = 0.5, n = 20))
  mym.3$out  # m = 16954, K = 8.3

# Power calculation
  mypower <- power.3m(expr = myod1, q2 = 1, q3 = 1, d = 0.2, m = 13177)
  mypower$out  # power = 0.80
# Power calculation under constrained p (p = 0.5)
  mypower.1 <- power.3m(expr = myod1, q2 = 1, q3 = 1, d = 0.2, m = 13177,
                 constraint = list(p = 0.5))
  mypower.1$out  # power = 0.69

# MDES calculation
  mymdes <- power.3m(expr = myod1, q2 = 1, q3 = 1, power = 0.80, m = 13176)
  mymdes$out  # d = 0.20


# ------- conventional power analyses with cost.model = FALSE-------
# Required sample size
  myK <- power.3m(cost.model = FALSE, expr = myod1, d = 0.2, q2 = 1, q3 = 1, power = 0.8)
  myK$out  # K = 11.5
  # myK$par  # parameters and their values used for the function
# or equivalently, specify every argument in the function
  myK <- power.3m(cost.model = FALSE, d = 0.2, power = 0.8, q2 = 1, q3 = 1,
                  icc2 = 0.2, icc3 = 0.1, omega = 0.02,
                  r12 = 0.5, r22 = 0.5, r32m = 0.5,
```

```
                    c1 = 1, c2 = 5,
                    c1t = 1, c2t = 200, c3 = 200,
                    n = 13, J = 15, p = 0.23)

# Power calculation
  mypower1 <- power.3m(cost.model = FALSE, expr = myod1, K = 11.5, d = 0.2, q2 = 1, q3 = 1)
  mypower1$out  # power = 0.80

# MDES calculation
  mymdes1 <- power.3m(cost.model = FALSE, expr = myod1, K = 11.5, power = 0.8, q2 = 1, q3 = 1)
  mymdes1$out  # d = 0.20
```

---

power.4 *Budget and/or sample size, power, MDES calculation for four-level CRTs*

Description

This function can calculate required budget for desired power, power or minimum detectable effect size (MDES) under fixed budget for four-level cluster randomized trials (CRTs). It also can perform conventional power analyses (e.g., required sample size, power, and MDES calculation).

Usage

```
power.4(cost.model = TRUE, expr = NULL, constraint = NULL,
  sig.level = 0.05, two.tailed = TRUE, d = NULL, power = NULL,
  m = NULL, n = NULL, J = NULL, K = NULL, L = NULL, p = NULL,
  icc2 = NULL, icc3 = NULL, icc4 = NULL, r12 = NULL, r22 = NULL,
  r32 = NULL, r42 = NULL, q = NULL, c1 = NULL, c2 = NULL,
  c3 = NULL, c4 = NULL, c1t = NULL, c2t = NULL, c3t = NULL,
  c4t = NULL, dlim = NULL, powerlim = NULL, Llim = NULL,
  mlim = NULL, rounded = TRUE)
```

Arguments

`cost.model`	logical; power analyses accommodating costs and budget (e.g., required budget for desired power, power/MDES under fixed budget) if TRUE, otherwise conventional power analyses (e.g., required sample size, power, or MDES calculation); default value is TRUE.
`expr`	returned objects from function `od.4`; default value is NULL; if `expr` is specified, parameter values of `icc2`, `icc3`, `icc4`, `r12`, `r22`, `r32`, `r42`, `c1`, `c2`, `c3`, `c4`, `c1t`, `c2t`, `c3t`, `c4t`, `p`, `n`, `J`, and `K` used or solved in function `od.4` will be passed to current function; only the values of `p`, `n`, `J`, and/or `K` that specified or solved in function `od.4` can be overwritten if `constraint` is specified.
`constraint`	specify the constrained values of `p`, `n`, `J`, and/or `K` in list format to overwrite those from `expr`; default value is NULL.

sig.level	significance level or type I error rate, default value is 0.05.
two.tailed	logical; two-tailed tests if TRUE, otherwise one-tailed tests; default value is TRUE.
d	effect size.
power	statistical power.
m	total budget.
n	the level-1 sample size per level-2 unit.
J	the level-2 sample size per level-3 unit.
K	the level-3 sample size per level-4 unit.
L	the total level-4 sample size.
p	the proportion of level-4 clusters/units to be assigned to treatment.
icc2	the unconditional intraclass correlation coefficient (ICC) at level 2.
icc3	the unconditional intraclass correlation coefficient (ICC) at level 3.
icc4	the unconditional intraclass correlation coefficient (ICC) at level 4.
r12	the proportion of level-1 variance explained by covariates.
r22	the proportion of level-2 variance explained by covariates.
r32	the proportion of level-3 variance explained by covariates.
r42	the proportion of level-4 variance explained by covariates.
q	the number of covariates at level 4.
c1	the cost of sampling one level-1 unit in control condition.
c2	the cost of sampling one level-2 unit in control condition.
c3	the cost of sampling one level-3 unit in control condition.
c4	the cost of sampling one level-4 unit in control condition.
c1t	the cost of sampling one level-1 unit in treatment condition.
c2t	the cost of sampling one level-2 unit in treatment condition.
c3t	the cost of sampling one level-3 unit in treatment condition.
c4t	the cost of sampling one level-4 unit in treatment condition.
dlim	the range for searching the root of effect size (d) numerically, default value is c(0, 5).
powerlim	the range for searching the root of power (power) numerically, default value is c(1e-10, 1 - 1e-10).
Llim	the range for searching the root of level-4 sample size (L) numerically, default value is c(4, 1e+10).
mlim	the range for searching the root of budget (m) numerically, default value is the costs sampling Llim level-4 units across treatment conditions or c(4 * Lcost, 1e+10 * Lcost) with Lcost = ((1 - p) * (c1 * n * J * K + c2 * J * K + c3 * K + c4) + p * (c1t * n * J * K + c2t * J * K + c3t * K + c4t)).
rounded	logical; round the values of p, n/J/K that are from functions od.4 to two decimal places and integer, respectively if TRUE, otherwise no rounding; default value is TRUE.

Value

Required budget (and/or required level-4 sample size), statistical power, or MDES depending on the specification of parameters. The function also returns the function name, design type, and parameters used in the calculation.

References

Shen, Z. (in progress). Using optimal sample allocation to improve statistical precision and design efficiency for multilevel randomized trials (Unpublished doctoral dissertation). University of Cincinnati, Cincinnati, OH.

Examples

```
# unconstrained optimal design
  myod1 <- od.4(icc2 = 0.2, icc3 = 0.1, icc4 = 0.05,
              r12 = 0.5, r22 = 0.5, r32 = 0.5, r42 = 0.5,
              c1 = 1, c2 = 5, c3 = 25, c4 = 125,
              c1t = 1, c2t = 50, c3t = 250, c4t = 2500)
  myod1$out # output # n = 7.1, J = 3.2, K = 4.2, p = 0.23

# ------- power analyses by default considering costs and budget -------
# required budget and sample size
  mym.1 <- power.4(expr = myod1, d = 0.2, q = 1, power = 0.8)
  mym.1$out  # m = 71161, L = 57.1
  #mym.1$par  # parameters and their values used for the function
# or equivalently, specify every argument in the function
  mym.1 <- power.4(d = 0.2, power = 0.8, q = 1,
                 icc2 = 0.2, icc3 = 0.1, icc4 = 0.05,
                 r12 = 0.5, r22 = 0.5, r32 = 0.5, r42 = 0.5,
                 c1 = 1, c2 = 5, c3 = 25, c4 = 125,
                 c1t = 1, c2t = 50, c3t = 250, c4t = 2500,
                 n = 7, J = 3, K = 4, p = 0.23)
# required budget and sample size with constrained p (p = 0.5)
  mym.2 <- power.4(expr = myod1, d = 0.2, q = 1, power = 0.8,
                 constraint = list(p = 0.5))
  mym.2$out  # m = 93508, L = 41.1
# required budget and sample size with constrained p and K
  mym.3 <- power.4(expr = myod1, d = 0.2, q = 1, power = 0.8,
                 constraint = list(p = 0.5, K = 20))
  mym.3$out  # m = 157365, L = 25.7

# Power calculation
  mypower <- power.4(expr = myod1, q = 1, d = 0.2, m = 71161)
  mypower$out  # power = 0.80
# Power calculation under constrained p (p = 0.5)
  mypower.1 <- power.4(expr = myod1, q = 1, d = 0.2, m = 71161,
                 constraint = list(p = 0.5))
  mypower.1$out  # power = 0.68

# MDES calculation
  mymdes <- power.4(expr = myod1, q = 1, power = 0.80, m = 71161)
  mymdes$out  # d = 0.20
```

```
# ------- conventional power analyses with cost.model = FALSE-------
# Required sample size
  myL <- power.4(cost.model = FALSE, expr = myod1, d = 0.2, q = 1, power = 0.8)
  myL$out  # L = 57.1
#myL$par  # parameters and their values used for the function
# or equivalently, specify every argument in the function
  myL <- power.4(cost.model = FALSE, d = 0.2, power = 0.8, q = 1,
                 icc2 = 0.2, icc3 = 0.1, icc4 = 0.05,
                 r12 = 0.5, r22 = 0.5, r32 = 0.5, r42 = 0.5,
                 n = 7, J = 3, K = 4, p = 0.23)

# Power calculation
  mypower1 <- power.4(cost.model = FALSE, expr = myod1, L = 57, d = 0.2, q = 1)
  mypower1$out  # power = 0.80

# MDES calculation
  mymdes1 <- power.4(cost.model = FALSE, expr = myod1, L = 57, power = 0.8, q = 1)
  mymdes1$out  # d = 0.20
```

`power.4m` *Budget and/or sample size, power, MDES calculation for four-level multisite randomized trials*

Description

This function can calculate required budget for desired power, power or minimum detectable effect size (MDES) under fixed budget for four-level multisite randomized trials (MRTs). It also can perform conventional power analyses (e.g., required sample size, power, and MDES calculation).

Usage

```
power.4m(cost.model = TRUE, expr = NULL, constraint = NULL,
  sig.level = 0.05, two.tailed = TRUE, d = NULL, power = NULL,
  m = NULL, n = NULL, J = NULL, K = NULL, L = NULL, p = NULL,
  icc2 = NULL, icc3 = NULL, icc4 = NULL, r12 = NULL, r22 = NULL,
  r32 = NULL, r42m = NULL, q3 = NULL, q4 = NULL, c1 = NULL,
  c2 = NULL, c3 = NULL, c4 = NULL, c1t = NULL, c2t = NULL,
  c3t = NULL, omega = NULL, dlim = NULL, powerlim = NULL,
  Llim = NULL, mlim = NULL, rounded = TRUE)
```

Arguments

`cost.model` logical; power analyses accommodating costs and budget (e.g., required budget for desired power, power/MDES under fixed budget) if TRUE, otherwise conventional power analyses (e.g., required sample size, power, or MDES calculation); default value is TRUE.

expr	returned objects from function od.4m; default is NULL; if expr is specified, parameter values of icc2, icc3, icc4, r12, r22, r32, r42m, c1, c2, c3, c4, c1t, c2t, c3t, p, n, J, and K used or solved in function od.4m will be passed to current function; only the values of p, n, J, and/or K that specified or solved in function od.4m can be overwritten if constraint is specified.
constraint	specify the constrained values of p, n, J, and/or K in list format to overwrite those from expr; default value is NULL.
sig.level	significance level or type I error rate, default value is 0.05.
two.tailed	logical; two-tailed tests if TRUE, otherwise one-tailed tests; default value is TRUE.
d	effect size.
power	statistical power.
m	total budget.
n	the level-1 sample size per level-2 unit.
J	the level-2 sample size per level-3 unit.
K	the level-3 sample size per level-4 unit.
L	the total level-4 sample size.
p	the proportion of level-3 units to be assigned to treatment.
icc2	the unconditional intraclass correlation coefficient (ICC) at level 2.
icc3	the unconditional intraclass correlation coefficient (ICC) at level 3.
icc4	the unconditional intraclass correlation coefficient (ICC) at level 4.
r12	the proportion of level-1 variance explained by covariates.
r22	the proportion of level-2 variance explained by covariates.
r32	the proportion of level-3 variance explained by covariates.
r42m	the proportion of variance of site-specific treatment effect explained by covariates.
q3	the number of covariates at level 3.
q4	the number of covariates at level 4.
c1	the cost of sampling one level-1 unit in control condition.
c2	the cost of sampling one level-2 unit in control condition.
c3	the cost of sampling one level-3 unit in control condition.
c4	the cost of sampling one level-4 unit.
c1t	the cost of sampling one level-1 unit in treatment condition.
c2t	the cost of sampling one level-2 unit in treatment condition.
c3t	the cost of sampling one level-3 unit in treatment condition.
omega	the standardized variance of site-specific treatment effect
dlim	the range for searching the root of effect size (d) numerically, default value is c(0, 5).
powerlim	the range for searching the root of power (power) numerically, default value is c(1e-10, 1 - 1e-10).

`Llim` the range for searching the root of level-4 sample size (`L`) numerically, default value is c(4, 1e+10).

`mlim` the range for searching the root of budget (`m`) numerically, default is the costs sampling `Llim` level-4 units or c(4 * Lcost, 1e+10 * Lcost) with Lcost = ((1 - p) * (c1 * n * J * K + c2 * J * K + c3 * K) + p * (c1t * n * J * K + c2t * J * K + c3t * K) + c4.

`rounded` logical; round the values of `p`, `n`/`J`/`K` that are from functions `od.4` to two decimal places and integer, respectively if TRUE, otherwise no rounding; default value is TRUE.

Value

Required budget (and/or required level-4 sample size), statistical power, or MDES depending on the specification of parameters. The function also returns the function name, design type, and parameters used in the calculation.

References

Shen, Z. (in progress). Using optimal sample allocation to improve statistical precision and design efficiency for multilevel randomized trials (Unpublished doctoral dissertation). University of Cincinnati, Cincinnati, OH.

Examples

```
# unconstrained optimal design #---------
  myod1 <- od.4m(icc2 = 0.2, icc3 = 0.1, icc4 = 0.05, omega = 0.02,
               r12 = 0.5, r22 = 0.5, r32 = 0.5, r42m = 0.5,
               c1 = 1, c2 = 5, c3 = 25,
               c1t = 1, c2t = 50, c3t = 250, c4 = 500,
               varlim = c(0, 0.005))
  myod1$out # n = 8.3, J = 3.2, K = 4.9, p = 0.36

# ------- power analyses by default considering costs and budget -------
# required budget and sample size
  mym.1 <- power.4m(expr = myod1, d = 0.2, q3 = 1, q4 = 1, power = 0.8)
  mym.1$out  # m = 28668, L = 19.5
  # mym.1$par  # parameters and their values used for the function
# or equivalently, specify every argument in the function
  mym.1 <- power.4m(d = 0.2, power = 0.8, q3 = 1, q4 = 1,
                  icc2 = 0.2, icc3 = 0.1, icc4 = 0.05, omega = 0.02,
                  r12 = 0.5, r22 = 0.5, r32 = 0.5, r42m = 0.5,
                  c1 = 1, c2 = 5, c3 = 25,
                  c1t = 1, c2t = 50, c3t = 250, c4 = 500,
                  n = 8, J = 3, K = 5, p = 0.36)
# required budget and sample size with constrained p
  mym.2 <- power.4m(expr = myod1, d = 0.2, q3 = 1, q4 = 1, power = 0.8,
                  constraint = list(p = 0.5))
  mym.2$out  # m = 31379, L = 18.2
# required budget and sample size with constrained p and n
  mym.3 <- power.4m(expr = myod1, d = 0.2, q3 = 1, q4 = 1, power = 0.8,
                  constraint = list(p = 0.5, n = 20))
```

```
  mym.3$out  # m = 32257, L = 17.0

# Power calculation
  mypower <- power.4m(expr = myod1, q3 = 1, q4 = 1, d = 0.2, m = 28668)
  mypower$out  # power = 0.80
# Power calculation under constrained p (p = 0.5)
  mypower.1 <- power.4m(expr = myod1, q3 = 1, q4 = 1, d = 0.2, m = 28668,
               constraint = list(p = 0.5))
  mypower.1$out  # power = 0.76

# MDES calculation
  mymdes <- power.4m(expr = myod1, q3 = 1, q4 = 1, power = 0.80, m = 28668)
  mymdes$out  # d = 0.20


# ------- conventional power analyses with cost.model = FALSE-------
# Required sample size
  myL <- power.4m(cost.model = FALSE, expr = myod1, d = 0.2, q3 = 1, q4 = 1, power = 0.8)
  myL$out  # L = 19.5
  # myL$par  # parameters and their values used for the function
# or equivalently, specify every argument in the function
  myL <- power.4m(cost.model = FALSE, d = 0.2, power = 0.8, q3 = 1, q4 = 1,
                icc2 = 0.2, icc3 = 0.1, icc4 = 0.05, omega = 0.02,
               r12 = 0.5, r22 = 0.5, r32 = 0.5, r42m = 0.5,
               c1 = 1, c2 = 5, c3 = 25,
               c1t = 1, c2t = 50, c3t = 250, c4 = 500,
               n = 8, J = 3, K = 5, p = 0.36)

# Power calculation
  mypower1 <- power.4m(cost.model = FALSE, expr = myod1, L = 19.5, d = 0.2, q3 = 1, q4 = 1)
  mypower1$out  # power = 0.80

# MDES calculation
  mymdes1 <- power.4m(cost.model = FALSE, expr = myod1, L = 19.5, power = 0.8, q3 = 1, q4 = 1)
  mymdes1$out  # d = 0.20
```

---

re *Relative efficiency (RE) calculation*

Description

Calculate the relative efficiency (RE) between two designs, it returns same results as those from function rpe

Usage

```
re(od, subod, rounded = TRUE, verbose = TRUE)
```

Arguments

- `od` — returned object of first design (e.g., unconstrained optimal design) from function `od.1`, `od.2`, `od.3`, `od.4`, `od.2m`, `od.3m`, or `od.4m`
- `subod` — returned object of second design (e.g., constrained optimal design) from function `od.1`, `od.2`, `od.3`, `od.4`, `od.2m`, `od.3m`, or `od.4m`
- `rounded` — logical; round the values of `p`, `n`/`J`/`K` that are from functions to two decimal places and integer, respectively if TRUE, no rounding if FALSE; default is TRUE.
- `verbose` — logical; print the value of relative efficiency if TRUE, otherwise not; default is TRUE.

Value

Relative efficiency value

References

Shen, Z., & Kelcey, B. (2018, April). Optimal design of cluster randomized trials under condition- and unit-specific cost structures. Roundtable discussion presented at American Educational Research Association (AERA) annual conference, New York City, NY;

Shen, Z., & Kelcey, B. (revise & resubmit). Optimal sample allocation under unequal costs in cluster-randomized trials. Journal of Educational and Behavioral Statistics.

Shen, Z. (in progress). Using optimal sample allocation to improve statistical precision and design efficiency for multilevel randomized trials (Unpublished doctoral dissertation). University of Cincinnati, Cincinnati, OH.

Examples

```
# unconstrained optimal design of 2-level CRT #----------
  myod1 <- od.2(icc = 0.2, r12 = 0.5, r22 = 0.5, c1 = 1, c2 = 5, c1t = 1, c2t = 50,
              varlim = c(0.01, 0.02))
# constrained optimal design with n = 20
  myod2 <- od.2(icc = 0.2, r12 = 0.5, r22 = 0.5, c1 = 1, c2 = 5, c1t = 1, c2t = 50,
              n = 20, varlim = c(0.005, 0.025))
# relative efficiency (RE)
  myre <- re(od = myod1, subod= myod2)
  myre$out # RE = 0.88
# constrained optimal design with p = 0.5
  myod2 <- od.2(icc = 0.2, r12 = 0.5, r22 = 0.5, c1 = 1, c2 = 5, c1t = 1, c2t = 50,
             p = 0.5, varlim = c(0.005, 0.025))
# relative efficiency (RE)
  myre <- re(od = myod1, subod= myod2)
  myre$out # RE = 0.90

# unconstrained optimal design of 3-level CRT #----------
  myod1 <- od.3(icc2 = 0.2, icc3 = 0.1, r12 = 0.5, r22 = 0.5, r32 = 0.5,
             c1 = 1, c2 = 5, c3 = 25, c1t = 1, c2t = 50, c3t = 250,
             varlim = c(0.005, 0.025))
# constrained optimal design with J = 20
  myod2 <- od.3(icc2 = 0.2, icc3 = 0.1, r12 = 0.5, r22 = 0.5, r32 = 0.5, J = 20,
```

```
              c1 = 1, c2 = 5, c3 = 25, c1t = 1, c2t = 50, c3t = 250,
              varlim = c(0, 0.025))
# relative efficiency (RE)
  myre <- re(od = myod1, subod= myod2)
  myre$out # RE = 0.53

# unconstrained optimal design of 4-level CRT #---------
  myod1 <- od.4(icc2 = 0.2, icc3 = 0.1, icc4 = 0.05, r12 = 0.5,
               r22 = 0.5, r32 = 0.5, r42 = 0.5,
               c1 = 1, c2 = 5, c3 = 25, c4 = 125,
               c1t = 1, c2t = 50, c3t = 250, c4t = 2500,
               varlim = c(0, 0.01))
# constrained optimal design with p = 0.5
  myod2 <- od.4(icc2 = 0.2, icc3 = 0.1, icc4 = 0.05, r12 = 0.5, p = 0.5,
               r22 = 0.5, r32 = 0.5, r42 = 0.5,
               c1 = 1, c2 = 5, c3 = 25, c4 = 125,
               c1t = 1, c2t = 50, c3t = 250, c4t = 2500,
               varlim = c(0, 0.01))
# relative efficiency (RE)
  myre <- re(od = myod1, subod= myod2)
  myre$out # RE = 0.78
```

`rpe` *Relative precision and efficiency (RPE) calculation*

Description

Calculate the relative precision and efficiency (RPE) between two designs, it returns same results as those from function `re`

Usage

```
rpe(od, subod, rounded = TRUE, verbose = TRUE)
```

Arguments

`od`	returned object of first design (e.g., unconstrained optimal design) from function `od.1`, `od.2`, `od.3`, `od.4`, `od.2m`, `od.3m`, or `od.4m`
`subod`	returned object of second design (e.g., constrained optimal design) from function `od.1`, `od.2`, `od.3`, `od.4`, `od.2m`, `od.3m`, or `od.4m`
`rounded`	logical; round the values of `p`, `n`/`J`/`K` that are from functions to two decimal places and integer, respectively if TRUE, no rounding if FALSE; default is TRUE.
`verbose`	logical; print the value of relative precision and efficiency if TRUE, otherwise not; default is TRUE.

Value

Relative precision and efficiency value.

References

Shen, Z., & Kelcey, B. (2018, April). Optimal design of cluster randomized trials under condition- and unit-specific cost structures. Roundtable discussion presented at American Educational Research Association (AERA) annual conference, New York City, NY;

Shen, Z., & Kelcey, B. (revise & resubmit). Optimal sample allocation under unequal costs in cluster-randomized trials. Journal of Educational and Behavioral Statistics.

Shen, Z. (in progress). Using optimal sample allocation to improve statistical precision and design efficiency for multilevel randomized trials (Unpublished doctoral dissertation). University of Cincinnati, Cincinnati, OH.

Examples

```
# unconstrained optimal design of 2-level CRT #----------
  myod1 <- od.2(icc = 0.2, r12 = 0.5, r22 = 0.5, c1 = 1, c2 = 5, c1t = 1, c2t = 50,
              varlim = c(0.01, 0.02))
# constrained optimal design with n = 20
  myod2 <- od.2(icc = 0.2, r12 = 0.5, r22 = 0.5, c1 = 1, c2 = 5, c1t = 1, c2t = 50,
              n = 20, varlim = c(0.005, 0.025))
# relative precision and efficiency (RPE)
  myrpe <- rpe(od = myod1, subod= myod2)
  myrpe$out # RPE = 0.88
# constrained optimal design with p = 0.5
  myod2 <- od.2(icc = 0.2, r12 = 0.5, r22 = 0.5, c1 = 1, c2 = 5, c1t = 1, c2t = 50,
             p = 0.5, varlim = c(0.005, 0.025))
# relative precision and efficiency (RPE)
  mypre <- rpe(od = myod1, subod= myod2)
  mypre$out # RPE = 0.90

# unconstrained optimal design of 3-level CRT #----------
  myod1 <- od.3(icc2 = 0.2, icc3 = 0.1, r12 = 0.5, r22 = 0.5, r32 = 0.5,
             c1 = 1, c2 = 5, c3 = 25, c1t = 1, c2t = 50, c3t = 250,
             varlim = c(0.005, 0.025))
# constrained optimal design with J = 20
  myod2 <- od.3(icc2 = 0.2, icc3 = 0.1, r12 = 0.5, r22 = 0.5, r32 = 0.5, J = 20,
             c1 = 1, c2 = 5, c3 = 25, c1t = 1, c2t = 50, c3t = 250,
             varlim = c(0, 0.025))
# relative precision and efficiency (RPE)
  myrpe <- rpe(od = myod1, subod= myod2)
  myrpe$out # RPE = 0.53

# unconstrained optimal design of 4-level CRT #---------
  myod1 <- od.4(icc2 = 0.2, icc3 = 0.1, icc4 = 0.05, r12 = 0.5,
              r22 = 0.5, r32 = 0.5, r42 = 0.5,
              c1 = 1, c2 = 5, c3 = 25, c4 = 125,
              c1t = 1, c2t = 50, c3t = 250, c4t = 2500,
              varlim = c(0, 0.01))
# constrained optimal design with p = 0.5
  myod2 <- od.4(icc2 = 0.2, icc3 = 0.1, icc4 = 0.05, r12 = 0.5, p = 0.5,
              r22 = 0.5, r32 = 0.5, r42 = 0.5,
              c1 = 1, c2 = 5, c3 = 25, c4 = 125,
              c1t = 1, c2t = 50, c3t = 250, c4t = 2500,
```

```
                varlim = c(0, 0.01))
# relative precision and efficiency (RPE)
  myrpe <- rpe(od = myod1, subod= myod2)
  myrpe$out # RPE = 0.78
```

Index