Package ‘PowerUpR’

October 25, 2021

Type Package

Title Power Analysis Tools for Multilevel Randomized Experiments

Version 1.1.0

Date 2021-10-24

Description Includes tools to calculate statistical power, minimum detectable effect size (MDES), MDES difference (MDESD), and minimum required sample size for various multilevel randomized experiments (MRE) with continuous outcomes.
Accomodates 14 types of MRE designs to detect main treatment effect, seven types of MRE designs to detect moderated treatment effect (2-1-1, 2-1-2, 2-2-1, 2-2-2, 3-3-1, 3-3-2, and 3-3-3 designs; <total.lev> - <trt.lev> - <mod.lev>),
five types of MRE designs to detect mediated treatment effects (2-1-1, 2-2-1, 3-1-1, 3-2-1, and 3-3-1 designs; <trt.lev> - <med.lev> - <out.lev>), four types of partially nested (PN) design to detect main treatment effect, and three types of PN designs to detect mediated treatment effects (2/1, 3/1, 3/2; <trt.arm.lev> / <ctrl.arm.lev>).
See 'PowerUp!' Excel series at <https://www.causalevaluation.org/>.

Suggests knitr, rmarkdown

VignetteBuilder knitr

License GPL (>= 3)

NeedsCompilation no

Author Metin Bulus [aut, cre],
Nianbo Dong [aut],
Benjamin Kelcey [aut],
Jessaca Spybrook [aut]

Maintainer Metin Bulus <bulusmetin@gmail.com>

Repository CRAN

Date/Publication 2021-10-25 16:10:02 UTC

R topics documented:

`PowerUpR-package` *Power Analysis Tools for Multilevel Randomized Experiments*

Description

PowerUp! series consist of three excel-based applications to design various multilevel randomized experiments to detect main treatment effects, and to design two- and three-level cluster-randomized trials (CRTs) to detect multilevel moderation and mediation. For more information please refer to `https://www.causalevaluation.org/`.

`bcra3r2` *Three-Level Blocked Cluster-level Random Assignment Design, Treatment at Level 2*

Description

For three-level cluster-randomized block designs (treatment at level 2, with random effects across level 3 blocks), use `mdes.bcra3r2()` to calculate the minimum detectable effect size, `power.bcra3r2()` to calculate the statistical power, and `mrss.bcra3r2()` to calculate the minimum required sample size.

For partially nested blocked cluster randomized trials (interventions clusters in treatment groups) use `mdes.bcra3r2_pn()` to calculate the minimum detectable effect size, `power.bcra3r2_pn()` to calculate the statistical power, and `mrss.bcra3r2_pn()` to calculate the minimum required sample size (number of blocks).

Usage

```
mdes.bcra3r2(power=.80, alpha=.05, two.tailed=TRUE,
             rho2, rho3, esv3=NULL, omega3=esv3/rho3,
             p=.50, g3=0, r21=0, r22=0, r2t3=0,
             n, J, K)

power.bcra3r2(es=.25, alpha=.05, two.tailed=TRUE,
              rho2, rho3, esv3=NULL, omega3=esv3/rho3,
              p=.50, g3=0, r21=0, r22=0, r2t3=0,
              n, J, K)

mrss.bcra3r2(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
             n, J, K0=10, tol=.10,
             rho2, rho3, esv3=NULL, omega3=esv3/rho3,
             p=.50, g3=0, r21=0, r22=0, r2t3=0)

mdes.bcra3r2_pn(power=.80, alpha=.05, two.tailed=TRUE, df=NULL,
                rho3_trt=.10, omega3=.50, rho2_trt=.20, rho_ic=0,
                p=.50, r21=0, g3=0, n, J, K, ic_size=1)

power.bcra3r2_pn(es=.25,alpha=.05, two.tailed=TRUE, df=NULL,
                 rho3_trt=.10, omega3=.50, rho2_trt=.20, rho_ic=0,
                 p=.50, r21=0, g3=0, n, J, K, ic_size=1)

mrss.bcra3r2_pn(es=.25, power=.80, alpha=.05, two.tailed=TRUE, z.test=FALSE,
                rho3_trt=.10, omega3 = .50, rho2_trt=.20, rho_ic=0,
                p=.50, r21=0, g3=0, n, J, ic_size=1, K0=10, tol=.10)
```

Arguments

`power`	statistical power $(1-\beta)$.
`es`	effect size.
`alpha`	probability of type I error.
`two.tailed`	logical; `TRUE` for two-tailed hypothesis testing, `FALSE` for one-tailed hypothesis testing.
`df`	degrees of freedom.
`rho_ic`	proportion of variance in the outcome that is between intervention clusters.
`rho2_trt`	proportion of variance in the outcome (for treatment group) that is between level 2 units.
`rho3_trt`	proportion of variance in the outcome (for treatment group) that is between level 3 units.
`rho2`	proportion of variance in the outcome between level 2 units (unconditional ICC2).
`rho3`	proportion of variance in the outcome between level 3 units (unconditional ICC3).
`esv3`	effect size variability as the ratio of the treatment effect variance between level 3 units to the total variance in the outcome (level 1 + level 2 + level 3). `esv` also works. Ignored when `omega3` is specified.

omega3	treatment effect heterogeneity as ratio of treatment effect variance among level 3 units to the residual variance at level 3.
p	average proportion of level 2 units randomly assigned to treatment within level 3 units.
g3	number of covariates at level 3.
r21	proportion of level 1 variance in the outcome explained by level 1 covariates (applies to all levels in partially nested designs).
r22	proportion of level 2 variance in the outcome explained by level 2 covariates.
r2t3	proportion of treatment effect variance among level 3 units explained by level 3 covariates.
ic_size	sample size for each intervention cluster.
n	harmonic mean of level 1 units across level 2 units (or simple average).
J	harmonic mean of level 2 units across level 3 units (or simple average).
K	number of level 3 units.
K0	starting value for K.
tol	tolerance to end iterative process for finding K.
z.test	logical; TRUE for z-test.

Value

fun	function name.
parms	list of parameters used in power calculation.
df	degrees of freedom.
ncp	noncentrality parameter.
power	statistical power $(1 - \beta)$.
mdes	minimum detectable effect size.
K	number of level 3 units.

References

Dong, N., & Maynard, R. (2013). PowerUp!: A tool for calculating minimum detectable effect sizes and minimum required sample sizes for experimental and quasi-experimental design studies. *Journal of Research on Educational Effectiveness*, *6*(1), 24-67. doi: 10.1080/19345747.2012.673143

Lohr, S., Schochet, P. Z., & Sanders, E. (2014). Partially Nested Randomized Controlled Trials in Education Research: A Guide to Design and Analysis. NCER 2014-2000. National Center for Education Research. https://ies.ed.gov/ncer/pubs/20142000/pdf/20142000.pdf

Examples

```
# cross-checks
mdes.bcra3r2(rho3=.13, rho2=.10, omega3=.4,
             n=10, J=6, K=24)
power.bcra3r2(es = .246, rho3=.13, rho2=.10, omega3=.4,
              n=10, J=6, K=24)
```

```
mrss.bcra3r2(es = .246, rho3=.13, rho2=.10, omega3=.4,
             n=10, J=6)

# cross-checks
mdes.bcra3r2_pn(rho3_trt=.10, omega3=.50,
                rho2_trt=.15, rho_ic=.20,
                n=40, J=60, K=6, ic_size=10)
power.bcra3r2_pn(es=.399, rho3_trt=.10, omega3=.50,
                rho2_trt=.15, rho_ic=.20,
                n=40, J=60, K=6, ic_size=10)
mrss.bcra3r2_pn(es=.399, rho3_trt=.10, omega3=.50,
                 rho2_trt=.15, rho_ic=.20,
                 n=40, J=60, ic_size=10)
```

`bcra4r2` *Four-Level Blocked Cluster-level Random Assignment Design, Treatment at Level 2*

Description

For four-level cluster-randomized block designs (treatment at level 2, with random effects across level 3 and 4 blocks), use `mdes.bcra4r2()` to calculate the minimum detectable effect size, `power.bcra4r2()` to calculate the statistical power, and `mrss.bcra4r2()` to calculate the minimum required sample size.

Usage

```
mdes.bcra4r2(power=.80, alpha=.05, two.tailed=TRUE,
             rho2, rho3, rho4, esv3=NULL, esv4=NULL,
             omega3=esv3/rho3, omega4=esv4/rho4,
             p=.50, r21=0, r22=0, r2t3=0, r2t4=0, g4=0,
             n, J, K, L)

power.bcra4r2(es=.25, alpha=.05, two.tailed=TRUE,
              rho2, rho3, rho4, esv3=NULL, esv4=NULL,
              omega3=esv3/rho3, omega4=esv4/rho4,
              p=.50, r21=0, r22=0, r2t3=0, r2t4=0, g4=0,
              n, J, K, L)

mrss.bcra4r2(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
             n, J, K, L0=10, tol=.10,
             rho2, rho3, rho4, esv3=NULL, esv4=NULL,
             omega3=esv3/rho3, omega4=esv4/rho4,
             p=.50, r21=0, r22=0, r2t3=0, r2t4=0, g4=0)
```

Arguments

`power` statistical power $(1-\beta)$.

`es`	effect size.
`alpha`	probability of type I error.
`two.tailed`	logical; `TRUE` for two-tailed hypothesis testing, `FALSE` for one-tailed hypothesis testing.
`rho2`	proportion of variance in the outcome between level 2 units (unconditional ICC2).
`rho3`	proportion of variance in the outcome between level 3 units (unconditional ICC3).
`rho4`	proportion of variance in the outcome between level 4 units (unconditional ICC4).
`esv3`	effect size variability as the ratio of the treatment effect variance between level 3 units to the total variance in the outcome (level 1 + level 2 + level 3 + level 4). Ignored when `omega3` is specified.
`esv4`	effect size variability as the ratio of the treatment effect variance between level 4 units to the total variance in the outcome (level 1 + level 2 + level 3 + level 4). Ignored when `omega4` is specified.
`omega3`	treatment effect heterogeneity as ratio of treatment effect variance among level 3 units to the residual variance at level 3.
`omega4`	treatment effect heterogeneity as ratio of treatment effect variance among level 4 units to the residual variance at level 4.
`p`	average proportion of level 2 units randomly assigned to treatment within level 3 units.
`g4`	number of covariates at level 4.
`r21`	proportion of level 1 variance in the outcome explained by level 1 covariates.
`r22`	proportion of level 2 variance in the outcome explained by level 2 covariates.
`r2t3`	proportion of treatment effect variance among level 3 units explained by level 3 covariates.
`r2t4`	proportion of treatment effect variance among level 4 units explained by level 4 covariates.
`n`	harmonic mean of level 1 units across level 2 units (or simple average).
`J`	harmonic mean of level 2 units across level 3 units (or simple average).
`K`	harmonic mean of level 3 units across level 4 units (or simple average).
`L`	number of level 4 units.
`L0`	starting value for `L`.
`tol`	tolerance to end iterative process for finding `L`.

Value

`fun`	function name.
`parms`	list of parameters used in power calculation.
`df`	degrees of freedom.
`ncp`	noncentrality parameter.
`power`	statistical power $(1 - \beta)$.
`mdes`	minimum detectable effect size.
`L`	number of level 4 units.

Examples

```
# cross-checks
mdes.bcra4r2(rho4=.05, rho3=.15, rho2=.15,
             omega4=.50, omega3=.50, n=10, J=4, K=4, L=20)
power.bcra4r2(es = .206, rho4=.05, rho3=.15, rho2=.15,
              omega4=.50, omega3=.50, n=10, J=4, K=4, L=20)
mrss.bcra4r2(es = .206, rho4=.05, rho3=.15, rho2=.15,
             omega4=.50, omega3=.50, n=10, J=4, K=4)
```

---

bcra4r3 *Four-Level Blocked Cluster-level Random Assignment Design, Treatment at Level 3*

Description

For four-level cluster-randomized block designs (treatment at level 3, with random effects across level 4 blocks), use `mdes.bcra4r3()` to calculate the minimum detectable effect size, `power.bcra4r3()` to calculate the statistical power, and `mrss.bcra4r3()` to calculate the minimum required sample size.

Usage

```
mdes.bcra4r3(power=.80, alpha=.05, two.tailed=TRUE,
             rho2, rho3, rho4, esv4=NULL, omega4=esv4/rho4,
             p=.50, r21=0, r22=0, r23=0, r2t4=0, g4=0,
             n, J, K, L)

power.bcra4r3(es=.25, alpha=.05, two.tailed=TRUE,
              rho2, rho3, rho4, esv4=NULL, omega4=esv4/rho4,
              p=.50, r21=0, r22=0, r23=0, r2t4=0, g4=0,
              n, J, K, L)

mrss.bcra4r3(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
             n, J, K, L0=10, tol=.10,
             rho2, rho3, rho4, esv4=NULL, omega4=esv4/rho4,
             p=.50, r21=0, r22=0, r23=0, r2t4=0, g4=0)
```

Arguments

`power`	statistical power $(1 - \beta)$.
`es`	effect size.
`alpha`	probability of type I error.
`two.tailed`	logical; `TRUE` for two-tailed hypothesis testing, `FALSE` for one-tailed hypothesis testing.
`rho2`	proportion of variance in the outcome between level 2 units (unconditional ICC2).
`rho3`	proportion of variance in the outcome between level 3 units (unconditional ICC3).

rho4	proportion of variance in the outcome between level 4 units (unconditional ICC4).
esv4	effect size variability as the ratio of the treatment effect variance between level 4 units to the total variance in the outcome (level 1 + level 2 + level 3 + level 4). esv also works. Ignored when omega4 is specified.
omega4	treatment effect heterogeneity as ratio of treatment effect variance among level 4 units to the residual variance at level 4.
p	average proportion of level 3 units randomly assigned to treatment within level 4 units.
g4	number of covariates at level 4.
r21	proportion of level 1 variance in the outcome explained by level 1 covariates.
r22	proportion of level 2 variance in the outcome explained by level 2 covariates.
r23	proportion of level 3 variance in the outcome explained by level 3 covariates.
r2t4	proportion of treatment effect variance among level 4 units explained by level 4 covariates.
n	harmonic mean of level 1 units across level 2 units (or simple average).
J	harmonic mean of level 2 units across level 3 units (or simple average).
K	harmonic mean of level 3 units across level 4 units (or simple average).
L	number of level 4 units.
L0	starting value for L.
tol	tolerance to end iterative process for finding L.

Value

fun	function name.
parms	list of parameters used in power calculation.
df	degrees of freedom.
ncp	noncentrality parameter.
power	statistical power $(1-\beta)$.
mdes	minimum detectable effect size.
L	number of level 4 units.

Examples

```
# cross-checks
mdes.bcra4r3(rho4=.05, rho3=.15, rho2=.15,
             omega4=.50, n=10, J=4, K=4, L=20)
power.bcra4r3(es = .316, rho4=.05, rho3=.15, rho2=.15,
              omega4=.50, n=10, J=4, K=4, L=20)
mrss.bcra4r3(es = .316, rho4=.05, rho3=.15, rho2=.15,
             omega4=.50, n=10, J=4, K=4)
```

bira2 *Two-Level Blocked Individual-level Random Assignment Design*

Description

For two-level randomized block designs (treatment at level 1, with random effects across level 2 blocks), use `mdes.bira2()` to calculate the minimum detectable effect size, `power.bira2()` to calculate the statistical power, and `mrss.bira2()` to calculate the minimum required sample size (number of blocks).

For treatment effect moderated by level 1 moderator use `power.mod211()`, `mdesd.mod211()`, and `mrss.mod211()` functions. For treatment effect moderated by level 2 moderator, use `power.mod212()`, `mdesd.mod212()`, and `mrss.mod212()` functions.

For partially nested blocked individual-level random assignment designs (blocked randomized controlled trial with intervention clusters) use `mdes.bira2_pn()` to calculate the minimum detectable effect size, `power.bira2_pn()` to calculate the statistical power, and `mrss.bira2_pn()` to calculate the minimum required sample size (number of blocks).

Usage

```
mdes.bira2(power=.80, alpha=.05, two.tailed=TRUE,
           rel1=1, rho2, esv2=NULL, omega2=esv2/rho2,
           g2=0, r21=0, r2t2=0, p=.50, n, J)

power.bira2(es=.25, alpha=.05, two.tailed=TRUE,
            rel1=1, rho2, esv2=NULL, omega2=esv2/rho2,
            g2=0, r21=0, r2t2=0, p=.50, n, J)

mrss.bira2(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
           rel1=1, rho2, esv2=NULL, omega2=esv2/rho2,
           r21=0, r2t2=0, J0=10, tol=.10, g2=0, p=.50, n)

power.mod211(es=.25, alpha=.05, two.tailed=TRUE,
             rho2, omega2tm, r21=0,
             p=.50, q=NULL, n, J)

mdesd.mod211(power=.80, alpha=.05, two.tailed=TRUE,
             rho2, omega2tm, g1=0, r21=0,
             p=.50, q=NULL, n, J)

mrss.mod211(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
            n, J0=10, tol=.10, rho2, omega2tm, r21=0,
            p=.50, q=NULL)

power.mod212(es=.25, alpha=.05, two.tailed=TRUE,
             rho2, omega2t, r21=0,
             p=.50, q=NULL, n, J)
```

```
mdesd.mod212(power=.80, alpha=.05, two.tailed=TRUE,
             rho2, omega2t, g1=0, r21=0,
             p=.50, q=NULL, n, J)

mrss.mod212(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
            n, J0=10, tol=.10, rho2, omega2t, r21=0,
            p=.50, q=NULL)

mdes.bira2_pn(power=.80, alpha=.05, two.tailed=TRUE, df=NULL,
              rho2_trt=.20, omega2=.50, rho_ic=0,
              p=.50, g2=0, r21=0, n, J, ic_size=1)

power.bira2_pn(es=.25,alpha=.05, two.tailed=TRUE, df=NULL,
               rho2_trt=.20, omega2=.50, rho_ic=0,
               p=.50, g2=0, r21=0, n, J, ic_size=1)

mrss.bira2_pn(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
              z.test=FALSE, rho2_trt=.20, omega2=.50, rho_ic=0,
              p=.50, g2=0, r21=0, n, ic_size=1, J0=10, tol=.10)
```

Arguments

power	statistical power $(1-\beta)$.
es	effect size.
alpha	probability of type I error.
two.tailed	logical; TRUE for two-tailed hypothesis testing, FALSE for one-tailed hypothesis testing.
df	degrees of freedom.
rho_ic	proportion of variance in the outcome (for treatment group) that is between intervention clusters.
rho2_trt	proportion of variance in the outcome (for treatment group) that is between level 2 units.
rel1	level 1 outcome reliability coefficient (see Cox \& Kelcey, 2019, p. 23).
rho2	proportion of variance in the outcome between level 2 units (unconditional ICC2). rho also works.
esv2	effect size variability as the ratio of the treatment effect variance between level 2 units to the total variance in the outcome (level 1 + level 2). esv also works. Ignored when omega2 is specified.
omega2	treatment effect heterogeneity as the ratio of the treatment effect variance between level 2 units to the unconditional level 2 residual variance. omega also works.
omega2t	standardized treatment effect variability across sites in the model that is not conditional on Level 2 moderator (ratio of the treatment effect variance between level 2 units to the total variance in the outcome.)

`omega2tm`	standardized effect variability of the moderation across sites (ratio of the moderated treatment effect variance between level 2 units to the total variance in the outcome.)
`p`	average proportion of level 1 units randomly assigned to treatment within level 2 units.
`q`	proportion of level 1 (on average) or level 2 units in the moderator subgroup.
`g1`	number of covariates at level 1.
`g2`	number of covariates at level 2.
`r21`	proportion of level 1 variance in the outcome explained by level 1 covariates (applies to all levels in partially nested designs).
`r2t2`	proportion of treatment effect variance among level 2 units explained by level 2 covariates.
`n`	level 1 sample size per block (average or harmonic mean).
`J`	number of blocks.
`ic_size`	sample size for each intervention cluster.
`J0`	starting value for `J`.
`tol`	tolerance to end iterative process for finding `J`.
`z.test`	logical; `TRUE` for z-test.

Value

`fun`	function name.
`parms`	list of parameters used in power calculation.
`df`	degrees of freedom.
`ncp`	noncentrality parameter.
`power`	statistical power $(1-\beta)$.
`mdes`	minimum detectable effect size.
`J`	number of level 2 units.

References

Cox, K., \& Kelcey, B. (2019). Optimal design of cluster-and multisite-randomized studies using fallible outcome measures. Evaluation Review, 43(3-4), 189-225. doi: 10.1177/0193841X19870878

Dong, N., Kelcey, B., \& Spybrook, J. (2020). Design considerations in multisite randomized trials probing moderated treatment effects. *Journal of Educational and Behavioral Statistics.* Advance online publication. doi: 10.3102/1076998620961492

Dong, N., \& Maynard, R. (2013). PowerUp!: A tool for calculating minimum detectable effect sizes and minimum required sample sizes for experimental and quasi-experimental design studies. *Journal of Research on Educational Effectiveness*, *6*(1), 24-67. doi: 10.1080/19345747.2012.673143

Lohr, S., Schochet, P. Z., \& Sanders, E. (2014). Partially nested randomized controlled trials in education research: A guide to design and analysis. NCER 2014-2000. National Center for Education Research. `https://ies.ed.gov/ncer/pubs/20142000/pdf/20142000.pdf`

Examples

```
# cross-checks
mdes.bira2(rho2=.17, omega2=.50, n=15, J=20)
power.bira2(es=.366, rho2=.17, omega2=.50, n=15, J=20)
mrss.bira2(es=.366, rho2=.17, omega2=.50, n=15)

# cross-checks
power.mod211(es=.248, rho2=.247, omega2tm=.148, r21=.493, n=20, J=35)
mdes.mod211(power=.853, rho2=.247, omega2tm=.148, r21=.493, n=20, J=35)
mrss.mod211(es=.248, power = .853, rho2=.247, omega2tm=.148, r21=.493, n=20)

# cross-checks
power.mod212(es=.248, rho2=.247, omega2t=.148, r21=.493, n=20, J=20)
mdes.mod212(power=.739, rho2=.247, omega2t=.148, r21=.493, n=20, J=20)
mrss.mod212(es=.248, power=.739, rho2=.247, omega2t=.148, r21=.493, n=20)

# cross-checks
mdes.bira2_pn(n=20, J=15, rho_ic=.10, ic_size=5)
power.bira2_pn(es=.357, n=20, J=15, rho_ic=.10, ic_size=5)
mrss.bira2_pn(es=.357, n=20, rho_ic=.10, ic_size=5)
```

---

bira3 *Three-Level Blocked Individual-level Random Assignment Design*

Description

For three-level randomized block designs (treatment at level 1, with random effects across level 2 and 3 blocks), use `mdes.bira3()` to calculate the minimum detectable effect size, `power.bira3()` to calculate the statistical power, and `mrss.bira3()` to calculate the minimum required sample size.

Usage

```
mdes.bira3(power=.80, alpha=.05, two.tailed=TRUE,
           rho2, rho3, esv2=NULL, esv3=NULL,
           omega2=esv2/rho2, omega3=esv3/rho3,
           p=.50, r21=0, r2t2=0, r2t3=0, g3=0,
           n, J, K)

power.bira3(es=.25, alpha=.05, two.tailed=TRUE,
            rho2, rho3, esv2=NULL, esv3=NULL,
            omega2=esv2/rho2, omega3=esv3/rho3,
            p=.50, r21=0, r2t2=0, r2t3=0, g3=0,
            n, J, K)

mrss.bira3(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
           n, J, K0=10, tol=.10,
           rho2, rho3, esv2=NULL, esv3=NULL,
           omega2=esv2/rho2, omega3=esv3/rho3,
           p=.50, r21=0, r2t2=0, r2t3=0, g3=0)
```

Arguments

power	statistical power $(1 - \beta)$.
es	effect size.
alpha	probability of type I error.
two.tailed	logical; TRUE for two-tailed hypothesis testing, FALSE for one-tailed hypothesis testing.
rho2	proportion of variance in the outcome between level 2 units (unconditional ICC2).
rho3	proportion of variance in the outcome between level 3 units (unconditional ICC3).
esv2	effect size variability as the ratio of the treatment effect variance between level 2 units to the total variance in the outcome (level 1 + level 2 + level 3). Ignored when omega2 is specified.
esv3	effect size variability as the ratio of the treatment effect variance between level 3 units to the total variance in the outcome (level 1 + level 2 + level 3). Ignored when omega3 is specified.
omega2	treatment effect heterogeneity as ratio of treatment effect variance among level 2 units to the residual variance at level 2.
omega3	treatment effect heterogeneity as ratio of treatment effect variance among level 3 units to the residual variance at level 3.
p	average proportion of level 1 units randomly assigned to treatment within level 2 units.
g3	number of covariates at level 3.
r21	proportion of level 1 variance in the outcome explained by level 1 covariates.
r2t2	proportion of treatment effect variance among level 2 units explained by level 2 covariates.
r2t3	proportion of treatment effect variance among level 3 units explained by level 3 covariates.
n	harmonic mean of level 1 units across level 2 units (or simple average).
J	harmonic mean of level 2 units across level 3 units (or simple average).
K	number of level 3 units.
K0	starting value for K.
tol	tolerance to end iterative process for finding K.

Value

fun	function name.
parms	list of parameters used in power calculation.
df	degrees of freedom.
ncp	noncentrality parameter.
power	statistical power $(1 - \beta)$.
mdes	minimum detectable effect size.
K	number of level 3 units.

Examples

```
# cross-checks
mdes.bira3(rho3=.20, rho2=.15,
           omega3=.10, omega2=.10,
           n=69, J=10, K=100)
power.bira3(es = .045, rho3=.20, rho2=.15,
            omega3=.10, omega2=.10,
            n=69, J=10, K=100)
mrss.bira3(es = .045, rho3=.20, rho2=.15,
           omega3=.10, omega2=.10,
           n=69, J=10)
```

---

bira4 *Four-Level Blocked Individual-level Random Assignment Design*

Description

For four-level randomized block designs (treatment at level 1, random effects across level 2, 3 and 4), use `mdes.bira4()` to calculate the minimum detectable effect size, `power.bira4()` to calculate the statistical power, and `mrss.bira4r1()` to calculate the minimum required sample size.

Usage

```
mdes.bira4(power=.80, alpha=.05, two.tailed=TRUE,
           rho2, rho3, rho4, esv2=NULL, esv3=NULL, esv4=NULL,
           omega2=esv2/rho2, omega3=esv3/rho3, omega4=esv4/rho4,
           p=.50, r21=0, r2t2=0, r2t3=0, r2t4=0, g4=0,
           n, J, K, L)

power.bira4(es=.25, alpha=.05, two.tailed=TRUE,
            rho2, rho3, rho4, esv2=NULL, esv3=NULL, esv4=NULL,
            omega2=esv2/rho2, omega3=esv3/rho3, omega4=esv4/rho4,
            p=.50, r21=0, r2t2=0, r2t3=0, r2t4=0, g4=0,
            n, J, K, L)

mrss.bira4(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
           n, J, K, L0=10, tol=.10,
           rho2, rho3, rho4, esv2=NULL, esv3=NULL, esv4=NULL,
           omega2=esv2/rho2, omega3=esv3/rho3, omega4=esv4/rho4,
           p=.50, r21=0, r2t2=0, r2t3=0, r2t4=0, g4=0)
```

Arguments

`power`	statistical power $(1-\beta)$.
`es`	effect size.
`alpha`	probability of type I error.

`two.tailed`	logical; `TRUE` for two-tailed hypothesis testing, `FALSE` for one-tailed hypothesis testing.
`rho2`	proportion of variance in the outcome between level 2 units (unconditional ICC2).
`rho3`	proportion of variance in the outcome between level 3 units (unconditional ICC3).
`rho4`	proportion of variance in the outcome between level 4 units (unconditional ICC4).
`esv2`	effect size variability as the ratio of the treatment effect variance between level 2 units to the total variance in the outcome (level 1 + level 2 + level 3 + level 4). Ignored when `omega2` is specified.
`esv3`	effect size variability as the ratio of the treatment effect variance between level 3 units to the total variance in the outcome (level 1 + level 2 + level 3 + level 4). Ignored when `omega3` is specified.
`esv4`	effect size variability as the ratio of the treatment effect variance between level 4 units to the total variance in the outcome (level 1 + level 2 + level 3 + level 4). Ignored when `omega4` is specified.
`omega2`	treatment effect heterogeneity as ratio of treatment effect variance among level 2 units to the residual variance at level 2.
`omega3`	treatment effect heterogeneity as ratio of treatment effect variance among level 3 units to the residual variance at level 3.
`omega4`	treatment effect heterogeneity as ratio of treatment effect variance among level 4 units to the residual variance at level 4.
`p`	average proportion of level 1 units randomly assigned to treatment within level 2 units.
`g4`	number of covariates at level 4.
`r21`	proportion of level 1 variance in the outcome explained by level 1 covariates.
`r2t2`	proportion of treatment effect variance among level 2 units explained by level 2 covariates.
`r2t3`	proportion of treatment effect variance among level 3 units explained by level 3 covariates.
`r2t4`	proportion of treatment effect variance among level 4 units explained by level 4 covariates.
`n`	harmonic mean of level 1 units across level 2 units (or simple average).
`J`	harmonic mean of level 2 units across level 3 units (or simple average).
`K`	harmonic mean of level 3 units across level 4 units (or simple average).
`L`	number of level 4 units.
`L0`	starting value for `L`.
`tol`	tolerance to end iterative process for finding `L`.

Value

`fun`	function name.
`parms`	list of parameters used in power calculation.
`df`	degrees of freedom.

ncp	noncentrality parameter.
power	statistical power $(1 - \beta)$.
mdes	minimum detectable effect size.
L	number of level 4 units.

Examples

```
# cross-checks
mdes.bira4(rho4=.05, rho3=.15, rho2=.15,
           omega4=.50, omega3=.50, omega2=.50,
           n=10, J=4, K=4, L=27)
power.bira4(es = 0.142, rho4=.05, rho3=.15, rho2=.15,
            omega4=.50, omega3=.50, omega2=.50,
            n=10, J=4, K=4, L=27)
mrss.bira4(es = 0.142, rho4=.05, rho3=.15, rho2=.15,
           omega4=.50, omega3=.50, omega2=.50,
           n=10, J=4, K=4)
```

---

conversion *Object Conversion*

Description

Use mrss.to.mdes() to convert an object returned from MRSS functions into an object returned from MDES functions, mrss.to.power() to convert an object returned from MRSS functions into an object returned from power functions, power.to.mdes() to convert an object returned from power functions into an object returned from MDES functions, mdes.to.power() to convert an object returned from MDES functions into an object returned from power functions, and mdes.to.pctl() to convert effect sizes or an object returned from MDES functions into percentiles.

Usage

```
mrss.to.mdes(object)
mrss.to.power(object)
power.to.mdes(object)
mdes.to.power(object)
mdes.to.pctl(object)
```

Arguments

object	an object returned from one of the functions in **PowerUpR** package.

Examples

```
design1 <- power.bira2(es=.15, rho2=.35, omega2=.10, n=83, J=10)
design2 <- power.to.mdes(design1)
mdes.to.pctl(design2)
```

cra2 — *Two-level Cluster-randomized Trials to Detect Main, Moderation and Mediation Effects*

Description

For main treatment effects, use `mdes.cra2()` to calculate the minimum detectable effect size, `power.cra2()` to calculate the statistical power, `mrss.cra2()` to calculate the minimum required sample size (number of clusters).

For moderator at level 1, use `mdesd.mod221()` to calculate the minimum detectable effect size, `power.mod221()` to calculate the statistical power, `mrss.mod221()` to calculate the minimum required sample size (number of clusters).

For moderator at level 2, use `mdesd.mod222()` to calculate the minimum detectable effect size, `power.mod222()` to calculate the statistical power, `mrss.mod222()` to calculate the minimum required sample size (number of clusters).

For mediator at level 1 and level 2, use `power.med211()` to calculate the statistical power for the 2-1-1 mediation, and `power.med221()` for the 2-2-1 mediation.

For cluster-randomized block designs (treatment at level 2, with fixed effects across level 3 blocks), use `mdes.bcra3f2()` to calculate the minimum detectable effect size, `power.bcra3f2()` to calculate the statistical power, and `mrss.bcra3f2()` to calculate the minimum required sample size (number of clusters per block).

For partially nested cluster randomized trials (interventions clusters in treatment groups) use `mdes.cra2_pn()` to calculate the minimum detectable effect size, `power.cra2_pn()` to calculate the statistical power, and `mrss.cra2_pn()` to calculate the minimum required sample size (number of schools).

Usage

```
mdes.cra2(power=.80, alpha=.05, two.tailed=TRUE,
          rel1 = 1, rho2, p=.50, g2=0, r21=0, r22=0,
          n, J)

mdesd.mod221(power=.80, alpha=.05, two.tailed=TRUE,
            rho2, omegam2, r21=0, r2m2=0,
            p=.50, q=NULL, n, J)

mdesd.mod222(power=.80, alpha=.05, two.tailed=TRUE,
            rho2, r22=0,
            p=.50, q=NULL, n, J)

power.cra2(es=.25, alpha=.05, two.tailed=TRUE,
           rel1 = 1, rho2, g2=0, p=.50, r21=0, r22=0,
           n, J)

power.mod221(es=.25, alpha=.05, two.tailed=TRUE,
             rho2, omegam2, r21=0, r2m2=0,
```

```
              p=.50, q=NULL, n, J)

power.mod222(es=.25, alpha=.05, two.tailed=TRUE,
             rho2, r22=0,
             p=.50, q=NULL, n, J)

power.med211(esa, esb1, esB, escp, two.tailed = TRUE, alpha = .05,
             mc = FALSE, nsims = 1000, ndraws = 1000,
             rhom2, rho2, r21, r22, r2m1, r2m2,
             p, n, J)

power.med221(esa, esb, escp, two.tailed = TRUE, alpha = .05,
             mc = FALSE, nsims = 1000, ndraws = 1000,
             rho2, r22, r21, r2m2,
             p = .50, n, J)

mrss.cra2(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
          n, J0=10, tol=.10, rel1 = 1,
          rho2, g2=0, p=.50, r21=0, r22=0)

mrss.mod221(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
            n, J0=10, tol=.10, rho2, omegam2, r21=0, r2m2=0,
            p=.50, q=NULL)

mrss.mod222(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
            n, J0=10, tol=.10, rho2, r22=0,
            p=.50, q=NULL)

mdes.bcra3f2(power=.80, alpha=.05, two.tailed=TRUE,
             rho2, p=.50, g2=0, r21=0, r22=0,
             n, J, K)

power.bcra3f2(es=.25, alpha=.05, two.tailed=TRUE,
              rho2, p=.50, g2=0, r21=0, r22=0,
              n, J, K)

mrss.bcra3f2(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
             n, K, J0=10, tol=.10,
             rho2, p=.50, g2=0, r21=0, r22=0)

mdes.cra2_pn(power=.80, alpha=.05, two.tailed=TRUE, df=NULL,
             rho2_trt=.20, rho_ic=0, p=.50,
             r21=0, n, J, ic_size=1)

power.cra2_pn(es=.25,alpha=.05, two.tailed=TRUE, df=NULL,
              rho2_trt=.20, rho_ic=0, p=.50,
              r21=0, n, J, ic_size=1)
```

```
mrss.cra2_pn(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
             z.test=FALSE, rho2_trt=.20, rho_ic=0, p=.50,
             r21=0, n, ic_size=1, J0=10, tol=.10)
```

Arguments

`power`	statistical power $(1-\beta)$
`es, esa, esb, esb1, esB, escp`	effect size for main/moderator effects, or for path coefficients a (treatment - mediator), b (level 2 mediator - outcome), b1 (level 1 mediator - outcome), B (overall mediator - outcome) or cp (direct treatment - outcome) in the mediation model.
`alpha`	probability of type I error.
`two.tailed`	logical; `FALSE` for one-tailed hypothesis testing.
`df`	degrees of freedom.
`rho_ic`	proportion of variance in the outcome that is between intervention clusters.
`rho2_trt`	proportion of variance in the outcome (for treatment group) that is between level 2 units.
`rel1`	level 1 outcome reliability coefficient (Cox \& Kelcey, 2019b).
`rho2`	proportion of variance in the outcome between level 2 units (unconditional ICC2). `rho` also works.
`rhom2`	proportion of variance in the mediator between level 2 units.
`omegam2`	ratio of the unconditional variance in the moderator effect that is between level 2 units to the residual variance between level 2 units in the null model.
`p`	proportion of level 2 units randomly assigned to treatment.
`q`	proportion of level 1 or level 2 units in the moderator subgroup.
`g2`	number of covariates at level 2.
`r21`	proportion of level 1 variance in the outcome explained by level 1 covariates (applies to all levels in partially nested designs).
`r22`	proportion of level 2 variance in the outcome explained by level 2 covariates.
`r2m1`	proportion of mediator variance at level 1 explained by level 1 covariates.
`r2m2`	proportion of variance in the moderator effect that is explained by level 2 predictors. For the mediation model, proportion of variance in the mediator explained by level 2 predictors.
`n`	harmonic mean of level 1 units across level 2 units (or simple average).
`J`	level 2 sample size.
`K`	number of level 3 units (blocks).
`ic_size`	sample size for each intervention cluster.
`J0`	starting value for `J`.
`tol`	tolerance to end iterative process for finding `J`.
`z.test`	logical; `TRUE` for z-test.

`mc`	logical; `TRUE` for monte carlo simulation based power.
`nsims`	number of replications, if `mc = TRUE`.
`ndraws`	number of draws from the distribution of the path coefficients for each replication, if `mc = TRUE`.

Value

`fun`	function name.
`parms`	list of parameters used in power calculation.
`df`	degrees of freedom.
`ncp`	noncentrality parameter.
`power`	statistical power $(1-\beta)$.
`mdes`	minimum detectable effect size.
`J`	number of level 2 units.

References

Cox, K., \& Kelcey, B. (2019a). Optimal sample allocation in group-randomized mediation studies with a group-level mediator. *The Journal of Experimental Education*, *87*(4), 616-640. doi: 10.1080/00220973.2018.1496060

Cox, K., \& Kelcey, B. (2019b). Optimal design of cluster-and multisite-randomized studies using fallible outcome measures. *Evaluation Review*, *43*(3-4), 189-225. doi: 10.1177/0193841X19870878

Dong, N., Spybrook J., Kelcey, B., \& Bulus, M. (2021). Power analyses for moderator effects with (non)random slopes in cluster randomized trials. *Methodology*, *17*(2), 92-110. doi: 10.5964/meth.4003

Kelcey, B., Dong, N., Spybrook, J., \& Cox, K. (2017). Statistical power for causally defined indirect effects in group-randomized trials with individual-level mediators. *Journal of Educational and Behavioral Statistics*, *42*(5), 499-530. doi: 10.3102/1076998617695506

Kelcey, B., Dong, N., Spybrook, J., \& Shen, Z. (2017). Experimental power for indirect effects in group-randomized studies with group-level mediators. *Multivariate behavioral research*, *52*(6), 699-719. doi: 10.1080/00273171.2017.1356212

Kelcey, B., \& Shen, Z. (2020). Strategies for efficient experimental design in studies probing 2-1-1 mediation. *The Journal of Experimental Education*, *88*(2), 311-334. doi: 10.1080/00220973.2018.1533796

Kelcey, B., Spybrook, J., \& Dong, N. (2019). Sample size planning for cluster-randomized interventions probing multilevel mediation. *Prevention Science*, *20*(3), 407-418. doi: 10.1007/s11121-01809216

Spybrook, J., Kelcey, B., \& Dong, N. (2016). Power for detecting treatment by moderator effects in two-and three-level cluster randomized trials. *Journal of Educational and Behavioral Statistics*, *41*(6), 605-627. doi: 10.3102/1076998616655442

Dong, N., \& Maynard, R. (2013). PowerUp!: A tool for calculating minimum detectable effect sizes and minimum required sample sizes for experimental and quasi-experimental design studies. *Journal of Research on Educational Effectiveness*, *6*(1), 24-67. doi: 10.1080/19345747.2012.673143

Lohr, S., Schochet, P. Z., \& Sanders, E. (2014). Partially nested randomized controlled trials in education research: A guide to design and analysis. NCER 2014-2000. National Center for Education Research. `https://ies.ed.gov/ncer/pubs/20142000/pdf/20142000.pdf`

Examples

```
# cross-checks for the main effect
mdes.cra2(rho2=.17, n=15, J=20)
power.cra2(es=.629, rho2=.17, n=15, J=20)
mrss.cra2(es=.629, rho2=.17, n=15)

# cross-checks for the randomly varying cont. L1 moderator effect
mdesd.mod221(rho2=.17, omegam2=.10, n=15, J=20)
power.mod221(es=.3563, rho2=.17, omegam2 =.10, n=15, J=20)
mrss.mod221(es=.3563, rho2=.17, omegam2 =.10, n=15)

# cross-checks for the non-randomly varying cont. L1 moderator effect
mdesd.mod221(rho2=.17, omegam2=0, n=15, J=20)
power.mod221(es=0.2957, rho2=.17, omegam2 =0, n=15, J=20)
mrss.mod221(es=0.2957, rho2=.17, omegam2 =0, n=15)

# cross-checks for the randomly varying bin. L1 moderator effect
mdesd.mod221(rho2=.17, omegam2=.10, q=.50, n=15, J=20)
power.mod221(es=.647, rho2=.17, omegam2 =.10, q=.50, n=15, J=20)
mrss.mod221(es=.647, rho2=.17, omegam2 =.10, q=.50, n=15)

# cross-checks for the non-randomly varying bin. L1 moderator effect
mdesd.mod221(rho2=.17, omegam2=0, q=.50, n=15, J=20)
power.mod221(es=0.5915, rho2=.17, omegam2 =0, q=.50, n=15, J=20)
mrss.mod221(es=0.5915, rho2=.17, omegam2 =0, q=.50, n=15)

# cross-checks for the cont. L2 moderator effect
mdesd.mod222(rho2=.17, n=15, J=100)
power.mod222(es=0.2757, rho2=.17, n=15, J=100)
mrss.mod222(es=0.2757, rho2=.17, n=15)

# cross-checks for the bin. L2 moderator effect
mdesd.mod222(rho2=.17, q=.50, n=15, J=100)
power.mod222(es=0.5514, rho2=.17, q=.50, n=15, J=100)
mrss.mod222(es=0.5514, rho2=.17, q=.50, n=15)

# 2-2-1 mediation
power.med221(esa=0.6596, esb=0.1891, escp=.1,
             rho2=.15, r22=.52, r21=.40, r2m2=.50,
             n=100, J=40, p=.5)

# 2-1-1 mediation
power.med211(esa=0.4135, esb1=0.0670, esB=0.3595, escp=.1,
            rhom2=.3, rho2=.3, r22=.6, r21=.6, r2m2=.6, r2m1=.6,
            n=30, J=80, p=.1)

# cross-checks for cluster-randomized block design
# treatment at level 2, with fixed effects across level 3 blocks
mdes.bcra3f2(rho2=.10, n=20, J=44, K=5)
power.bcra3f2(es = .145, rho2=.10, n=20, J=44, K=5)
mrss.bcra3f2(es = .145, rho2=.10, n=20, K=5)
```

```
# cross-checks for partially nested cluster-randomized trial
mdes.cra2_pn(n=40, J=70, rho2_trt=.15, rho_ic=.10, ic_size=10)
power.cra2_pn(es=.305, n=40, J=70, rho2_trt=.15, rho_ic=.10, ic_size=10)
mrss.cra2_pn(es=.305, n=40, rho2_trt=.15, rho_ic=.10, ic_size=10)
```

`cra3` *Three-level Cluster-randomized Trials to Detect Main, Moderation, and Mediation Effects*

Description

For main treatment effects, use `mdes.cra3()` to calculate the minimum detectable effect size, `power.cra3()` to calculate the statistical power, `mrss.cra3()` to calculate the minimum required sample size (number of clusters).

For moderator at level 1, use `mdesd.mod331()` to calculate the minimum detectable effect size, `power.mod331()` to calculate the statistical power, `mrss.mod331()` to calculate the minimum required sample size (number of clusters).

For moderator at level 2, use `mdesd.mod332()` to calculate the minimum detectable effect size, `power.mod332()` to calculate the statistical power, `mrss.mod332()` to calculate the minimum required sample size (number of clusters).

For moderator at level 3, use `mdesd.mod333()` to calculate the minimum detectable effect size, `power.mod333()` to calculate the statistical power, `mrss.mod333()` to calculate the minimum required sample size (number of clusters).

For mediator at level 3, use `power.med331()`, for mediator at level 2, use `power.med321()`, for mediator at level 1, use `power.med311()` to calculate the statistical power.

For cluster-randomized block designs (treatment at level 3, with fixed effects across level 4 blocks), use `mdes.bcra4f3()` to calculate the minimum detectable effect size, `power.bcra4f3()` to calculate the statistical power, and `mrss.bcra4f3()` to calculate the minimum required sample size (number of clusters per block).

Usage

```
mdes.cra3(power=.80, alpha=.05, two.tailed=TRUE,
          rho2, rho3, p=.50, g3=0, r21=0, r22=0, r23=0,
          n, J, K)

mdesd.mod331(power=.80, alpha=.05, two.tailed=TRUE,
            rho2, rho3, omegam2, omegam3,
            r21=0, r2m3=0,
            p=.50, q=NULL, n, J, K)

mdesd.mod332(power=.80, alpha=.05, two.tailed=TRUE,
            rho2, rho3, omegam3, r21=0, r22=0, r2m3=0,
            p=.50, q=NULL, n, J, K)

mdesd.mod333(power=.80, alpha=.05, two.tailed=TRUE,
```

```
             rho2, rho3, r21=0, r22=0, r23=0,
             p=.50, q=NULL, n, J, K)

power.cra3(es=.25, alpha=.05, two.tailed=TRUE,
           rho2, rho3, g3=0, r21=0, r22=0, r23=0,
           p=.50, n, J, K)

power.mod331(es=.25, alpha=.05, two.tailed=TRUE,
             rho2, rho3, omegam2, omegam3,
             r21=0, r2m3=0,
             p=.50, q=NULL, n, J, K)

power.mod332(es=.25, alpha=.05, two.tailed=TRUE,
             rho2, rho3, omegam3, r21=0, r22=0, r2m3=0,
             p=.50, q=NULL, n, J, K)

power.mod333(es=.25, alpha=.05, two.tailed=TRUE,
             rho2, rho3, r21=0, r22=0, r23=0,
             p=.50, q=NULL, n, J, K)

power.med331(esa, esB, two.tailed=TRUE, alpha=.05,
             mc=TRUE, nsims=1000, ndraws=1000,
             rho2, rho3, gm3=4, r2m3=0, r21=0, r22=0,
             g3=5, r23=0, p=.50, n, J, K)

power.med321(esa, esB, two.tailed=TRUE, alpha=.05,
             mc=TRUE, nsims=1000, ndraws=1000,
             rhom3, rho2, rho3, r2m2=0,
             gm3=4, r2m3=0, r21=0, r22=0, g3=5, r23=0,
             p=.50, n, J, K)

power.med311(esa, esB, two.tailed=TRUE, alpha=.05,
             mc=TRUE, nsims=1000, ndraws=1000,
             rhom2, rhom3, rho2, rho3,
             r2m1=0, r2m2=0, gm3=4, r2m3=0,
             r21=0, r22=0, g3=5, r23=0,
             p=.50, n, J, K)

mrss.cra3(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
          n, J, K0=10, tol=.10,
          rho2, rho3, p=.50, g3=0, r21=0, r22=0, r23=0)

mrss.mod331(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
            rho2, rho3, omegam2, omegam3,
            r21=0, r2m3=0,
            p=.50, q=NULL, n, J, K0=10, tol=.10)

mrss.mod332(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
```

```
             rho2, rho3, omegam3, r21=0, r22=0, r2m3=0,
             p=.50, q=NULL, n, J, K0=10, tol=.10)

mrss.mod333(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
            rho2, rho3, r21=0, r22=0, r23=0,
            p=.50, q=NULL, n, J, K0=10, tol=.10)

mdes.bcra4f3(power=.80, alpha=.05, two.tailed=TRUE,
             rho2, rho3, p=.50, r21=0, r22=0, r23=0, g3=0,
             n, J, K, L)

power.bcra4f3(es=.25, alpha=.05, two.tailed=TRUE,
              rho2, rho3, p=.50, r21=0, r22=0, r23=0, g3=0,
              n, J, K, L)

mrss.bcra4f3(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
             n, J, L, K0=10, tol=.10,
             rho2, rho3, p=.50, g3=0, r21=0, r22=0, r23=0)
```

Arguments

power	statistical power $(1-\beta)$.
es, esa, esB	effect size for main/moderator effects, or for path coefficients a (treatment - mediator), or B (overall mediator - outcome) in the mediation model.
alpha	probability of type I error.
two.tailed	logical; TRUE for two-tailed hypothesis testing, FALSE for one-tailed hypothesis testing.
rho2	proportion of variance in the outcome between level 2 units (unconditional ICC2).
rho3	proportion of variance in the outcome between level 3 units (unconditional ICC3).
rhom2	proportion of variance in the mediator between level 2 units.
rhom3	proportion of variance in the mediator between level 3 units.
omegam2	ratio of the unconditional variance in the moderator effect that is between level 2 units to the residual variance between level 2 units in the null model.
omegam3	ratio of the unconditional variance in the moderator effect that is between level 3 units to the residual variance between level 3 units in the null model.
p	proportion of level 3 units randomly assigned to treatment.
q	proportion of level 1, level 2, or level 3 units in the moderator subgroup.
g3	number of covariates at level 3.
gm3	number of covariates at level 3 for the mediation model.
r21	proportion of level 1 variance in the outcome explained by level 1 covariates.
r22	proportion of level 2 variance in the outcome explained by level 2 covariates.
r23	proportion of level 3 variance in the outcome explained by level 3 covariates.
r2m1	proportion of mediator variance at level 1 explained by level 1 predictors.

`r2m2`	proportion of variance in the mediator explained by level 2 predictors.
`r2m3`	proportion of variance in the moderator effect that is explained by level 3 predictors. For the mediation model, proportion of variance in the mediator explained by level 3 predictors.
`n`	harmonic mean of level 1 units across level 2 units (or simple average).
`J`	harmonic mean of level 2 units across level 3 units (or simple average).
`K`	level 3 sample size.
`L`	level 4 sample size (blocks).
`K0`	starting value for `K`.
`tol`	tolerance to end iterative process for finding `K`.
`mc`	logical; `TRUE` for monte carlo simulation based power.
`nsims`	number of replications, if `mc = TRUE`.
`ndraws`	number of draws from the distribution of the path coefficients for each replication, if `mc = TRUE`.

Value

`fun`	function name.
`parms`	list of parameters used in power calculation.
`df`	degrees of freedom.
`ncp`	noncentrality parameter.
`power`	statistical power $(1-\beta)$.
`mdes`	minimum detectable effect size.
`K`	number of level 3 units.

References

Spybrook, J., Kelcey, B., \& Dong, N. (2016). Power for detecting treatment by moderator effects in two-and three-level cluster randomized trials. *Journal of Educational and Behavioral Statistics*, *41*(6), 605-627. doi: 10.3102/1076998616655442

Dong, N., Kelcey, B., \& Spybrook, J. (2018). Power analyses for moderator effects in three-level cluster randomized trials. *The Journal of Experimental Education*, *86*(3), 489-514. doi: 10.1080/00220973.2017.1315714

Dong, N., \& Maynard, R. (2013). PowerUp!: A tool for calculating minimum detectable effect sizes and minimum required sample sizes for experimental and quasi-experimental design studies. *Journal of Research on Educational Effectiveness*, *6*(1), 24-67. doi: 10.1080/19345747.2012.673143

Kelcey, B., Xie, Y., Spybrook, J., \& Dong, N. (2020). Power and sample size determination for multilevel mediation in three-Level cluster-randomized trials. *Multivariate Behavioral Research*. Advance online publication. doi: 10.1080/00273171.2020.1738910

Kelcey, B., Spybrook, J., Dong, N., \& Bai, F. (2020). Cross-level mediation in school-randomized studies of teacher development: Experimental design and power. *Journal of Research on Educational Effectiveness*. Advance online publication. doi: 10.1080/19345747.2020.1726540

Examples

```
# cross-checks for the main effect
mdes.cra3(rho3=.06, rho2=.17, n=15, J=3, K=60)
power.cra3(es=.269, rho3=.06, rho2=.17, n=15, J=3, K=60)
mrss.cra3(es=.269, rho3=.06, rho2=.17, n=15, J=3)

# cross-checks for the randomly varying cont. L1 moderator effect
mdesd.mod331(power=.80, alpha=.05, two.tailed=TRUE,
            rho2=.17, rho3=.06, omegam2=.10, omegam3=.10,
            q=NULL, n=15, J=3, K=60)
power.mod331(es=0.1248, alpha=.05, two.tailed=TRUE,
            rho2=.17, rho3=.06, omegam2=.10, omegam3=.10,
            q=NULL, n=15, J=3, K=60)
mrss.mod331(es=0.1248, alpha=.05, two.tailed=TRUE,
            rho2=.17, rho3=.06, omegam2=.10, omegam3=.10,
            q=NULL, n=15, J=3)

# cross-checks for the non-randomly varying cont. L1 moderator effect
mdesd.mod331(power=.80, alpha=.05, two.tailed=TRUE,
            rho2=.17, rho3=.06, omegam2=0, omegam3=0,
            q=NULL, n=15, J=3, K=60)
power.mod331(es=.0946, alpha=.05, two.tailed=TRUE,
            rho2=.17, rho3=.06, omegam2=0, omegam3=0,
            q=NULL, n=15, J=3, K=60)
mrss.mod331(es=.0946, alpha=.05, two.tailed=TRUE,
            rho2=.17, rho3=.06, omegam2=0, omegam3=0,
            q=NULL, n=15, J=3)

# cross-checks for the randomly varying bin. L1 moderator effect
mdesd.mod331(power=.80, alpha=.05, two.tailed=TRUE,
            rho2=.17, rho3=.06, omegam2=.10, omegam3=.10,
            q=.50, n=15, J=3, K=60)
power.mod331(es=.2082, alpha=.05, two.tailed=TRUE,
            rho2=.17, rho3=.06, omegam2=.10, omegam3=.10,
            q=.50, n=15, J=3, K=60)
mrss.mod331(es=.2082, alpha=.05, two.tailed=TRUE,
            rho2=.17, rho3=.06, omegam2=.10, omegam3=.10,
            q=.50, n=15, J=3)

# cross-checks for the non-randomly varying bin. L1 moderator effect
mdesd.mod331(power=.80, alpha=.05, two.tailed=TRUE,
            rho2=.17, rho3=.06, omegam2=0, omegam3=0,
            q=.50, n=15, J=3, K=60)
power.mod331(es=.1893, alpha=.05, two.tailed=TRUE,
            rho2=.17, rho3=.06, omegam2=0, omegam3=0,
            q=.50, n=15, J=3, K=60)
mrss.mod331(es=.1893, alpha=.05, two.tailed=TRUE,
            rho2=.17, rho3=.06, omegam2=0, omegam3=0,
            q=.50, n=15, J=3)

# cross-checks for the randomly varying bin. L2 moderator effect
mdesd.mod332(rho3=.1, rho2=.1, omegam3=.05,
```

```
             q=.5, r21=.30, r22=.4, r2m3=0, n=20, J=4, K=60)
power.mod332(es=.2244, rho3=.1, rho2=.1, omegam3=.05,
             q=.5, r21=.30, r22=.4, r2m3=0, n=20, J=4, K=60)
mrss.mod332(es=.2244, rho3=.1, rho2=.1, omegam3=.05,
            q=.5, r21=.30, r22=.4, r2m3=0, n=20, J=4)

# cross-checks for the randomly varying cont. L2 moderator effect
mdesd.mod332(rho3=.1, rho2=.1, omegam3=.05,
            r21=.30, r22=.4, r2m3=0, n=20, J=4, K=60)
power.mod332(es=.1209, rho3=.1, rho2=.1, omegam3=.05,
            r21=.30, r22=.4, r2m3=0, n=20, J=4, K=60)
mrss.mod332(es=.1209, rho3=.1, rho2=.1, omegam3=.05,
            r21=.30, r22=.4, r2m3=0, n=20, J=4)

# cross-checks for the non-randomly varying bin. L2 moderator effect
mdesd.mod332(rho3=.1, rho2=.1, omegam3=0,
            q=.5, r21=.30, r22=.4, r2m3=0, n=20, J=4, K=60)
power.mod332(es=.2157, rho3=.1, rho2=.1, omegam3=0,
            q=.5, r21=.30, r22=.4, r2m3=0, n=20, J=4, K=60)
mrss.mod332(es=.2157, rho3=.1, rho2=.1, omegam3=0,
            q=.5, r21=.30, r22=.4, r2m3=0, n=20, J=4, K=60)

# cross-checks for the non-randomly varying cont. L2 moderator effect
mdesd.mod332(rho3=.1, rho2=.1, omegam3=0,
            r21=.30, r22=.4, r2m3=0, n=20, J=4, K=60)
power.mod332(es=.1079, rho3=.1, rho2=.1, omegam3=0,
             r21=.30, r22=.4, r2m3=0, n=20, J=4, K=60)
mrss.mod332(es=.1079, rho3=.1, rho2=.1, omegam3=0,
             r21=.30, r22=.4, r2m3=0, n=20, J=4, K=60)

# cross-checks for the randomly varying bin. L3 moderator effect
mdesd.mod333(rho3=.1, rho2=.1, q=.5,
            r21=.3, r22=.4, r23=.5, n=20, J=4, K=60)
power.mod333(es=.4128, rho3=.1, rho2=.1, q=.5,
             r21=.3, r22=.4, r23=.5, n=20, J=4, K=60)
mrss.mod333(es=.4128, rho3=.1, rho2=.1, q=.5,
             r21=.3, r22=.4, r23=.5, n=20, J=4, K=60)

# cross-checks for the randomly varying cont. L3 moderator effect
mdesd.mod333(rho3=.1, rho2=.1,
             r21=.3, r22=.4, r23=.5, n=20, J=4, K=60)
power.mod333(es=.2064, rho3=.1, rho2=.1,
             r21=.3, r22=.4, r23=.5, n=20, J=4, K=60)
mrss.mod333(es=.2064, rho3=.1, rho2=.1,
             r21=.3, r22=.4, r23=.5, n=20, J=4, K=60)

# 3-3-1 mediation
power.med331(esa= .50, esB = .30, rho2 = .15, rho3 = .15,
             r21 = .20, r22 = .20, g3 = 4,
             n = 20, J = 4, K = 80, p = .5)

# 3-2-1 mediation
power.med321(esa= .51, esB = .30, rhom3 = 0.27, rho2 = .15, rho3 = .19,
```

```
              r2m2 = .07, gm3 = 4, r2m3 = .16,
              r21 = .02, r22 = .41, g3 = 5, r23 = .38,
              p = .50, n = 20, J = 4, K = 60)

# 3-1-1 mediation
power.med311(esa= .49 , esB = .30,
              rhom2 = .05, rhom3 = .26, rho2 = .15, rho3 = .20,
              r2m1 = .10, r2m2 = .07, r2m3 = .17,
              r21 = .02, r22 = .41, r23 = .38,
              p = .50, n = 20, J = 4, K = 30)

# cross-checks for cluster-randomized block design
# treatment at level 3, with fixed effects across level 4 blocks
mdes.bcra4f3(rho3=.15, rho2=.15,
              n=10, J=4, K=23, L=15)
power.bcra4f3(es=0.137, rho3=.15, rho2=.15,
               n=10, J=4, K=33, L=15)
mrss.bcra4f3(es=0.137, rho3=.15, rho2=.15,
              n=10, J=4, L=15)
```

---

cra4 *Four-Level Cluster-randomized Trial*

Description

For main treatment effects, use `mdes.cra4()` calculate the minimum detectable effect size, `power.cra4()` to calculate the statistical power, and `mrss.cra4()` to calculate the minimum required sample size.

Usage

```
mdes.cra4(power=.80, alpha=.05, two.tailed=TRUE,
          rho2, rho3, rho4, p=.50, r21=0, r22=0, r23=0, r24=0, g4=0,
          n, J, K, L)

power.cra4(es=.25, alpha=.05, two.tailed=TRUE,
           rho2, rho3, rho4, p=.50, r21=0, r22=0, r23=0, r24=0, g4=0,
           n, J, K, L)

mrss.cra4(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
          n, J, K, L0=10, tol=.10,
          rho2, rho3, rho4, p=.50,
          r21=0, r22=0, r23=0, r24=0, g4=0)
```

Arguments

power	statistical power $(1-\beta)$.
es	effect size.
alpha	probability of type I error.

two.tailed	logical; TRUE for two-tailed hypothesis testing, FALSE for one-tailed hypothesis testing.
rho2	proportion of variance in the outcome between level 2 units (unconditional ICC2).
rho3	proportion of variance in the outcome between level 3 units (unconditional ICC3).
rho4	proportion of variance in the outcome between level 4 units (unconditional ICC4).
p	proportion of level 4 units randomly assigned to treatment.
g4	number of covariates at level 4.
r21	proportion of level 1 variance in the outcome explained by level 1 covariates.
r22	proportion of level 2 variance in the outcome explained by level 2 covariates.
r23	proportion of level 3 variance in the outcome explained by level 3 covariates.
r24	proportion of level 4 variance in the outcome explained by level 4 covariates.
n	harmonic mean of level 1 units across level 2 units (or simple average).
J	harmonic mean of level 2 units across level 3 units (or simple average).
K	harmonic mean of level 3 units across level 4 units (or simple average).
L	number of level 4 units.
L0	starting value for L.
tol	tolerance to end iterative process for finding L.

Value

fun	function name.
parms	list of parameters used in power calculation.
df	degrees of freedom.
ncp	noncentrality parameter.
power	statistical power $(1 - \beta)$.
mdes	minimum detectable effect size.
L	number of level 4 units.

Examples

```
# cross-checks
mdes.cra4(rho4=.05, rho3=.05, rho2=.10,
          n=10, J=2, K=3, L=20)
power.cra4(es = .412, rho4=.05, rho3=.05, rho2=.10,
           n=10, J=2, K=3, L=20)
mrss.cra4(es = .412, rho4=.05, rho3=.05, rho2=.10,
          n=10, J=2, K=3)
```

ira *Individual-level Random Assignment Designs*

Description

For an individual-level random assignment design (simple randomized controlled trial) use `mdes.ira()` to calculate the minimum detectable effect size, `power.ira()` to calculate the statistical power, and `mrss.ira()` to calculate the minimum required sample size (number of subjects).

If fixed block effects exist (randomized block designs), that is, treatment effect varies from one block to another but it is not random, use `mdes.bira2f1()` to calculate the minimum detectable effect size, `power.bira2f1()` to calculate the statistical power, and `mrss.bira2f1()` to calculate the minimum required sample size (number of subjects per block).

If block effect is constant, that is, treatment effect does not change from one block to another, use `mdes.bira2c1()` to calculate the minimum detectable effect size, `power.bira2c1()` to calculate the statistical power, and `mrss.bira2c1()` to calculate the minimum required sample size (number of subjects per block).

For partially nested individual-level random assignment designs (simple randomized controlled trial with intervention clusters) use `mdes.ira_pn()` to calculate the minimum detectable effect size, `power.ira_pn()` to calculate the statistical power, and `mrss.ira_pn()` to calculate the minimum required sample size (number of subjects).

Usage

```
mdes.ira(power=.80, alpha=.05, two.tailed=TRUE,
         p=.50, g1=0, r21=0, n)

power.ira(es=.25, alpha=.05, two.tailed=TRUE,
          p=.50, g1=0, r21=0, n)

mrss.ira(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
         n0=10, tol=.10,
         p=.50, g1=0, r21=0)

mdes.bira2f1(power=.80, alpha=.05, two.tailed=TRUE,
             p=.50, g1=0, r21=0, n, J)

power.bira2f1(es=.25, alpha=.05, two.tailed=TRUE,
              p=.50, g1=0, r21=0, n, J)

mrss.bira2f1(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
             J, n0=10, tol=.10,
             p=.50, g1=0, r21=0)

mdes.bira2c1(power=.80, alpha=.05, two.tailed=TRUE,
             p=.50, g1=0, r21=0,
             n, J)
```

```
power.bira2c1(es=.25, alpha=.05, two.tailed=TRUE,
              p=.50, g1=0, r21=0,
              n, J)

mrss.bira2c1(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
             J, n0=10, tol=.10,
             p=.50, g1=0, r21=0)

mdes.ira_pn(power=.80, alpha=.05, two.tailed=TRUE,
            df=NULL, ratio_tc_var=1,
            rho_ic=.20, p=.50, r21=0, n, ic_size=1)

power.ira_pn(es=.25,alpha=.05, two.tailed=TRUE,
             df=NULL, ratio_tc_var=1,
             rho_ic=.20, p=.50, r21=0, n, ic_size=1)

mrss.ira_pn(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
            ratio_tc_var=1, z.test=FALSE,
            rho_ic=.20, p=.50, r21=0, ic_size=1, n0=500, tol=.10)
```

Arguments

`power`	statistical power $(1 - \beta)$.
`es`	effect size.
`alpha`	probability of type I error.
`two.tailed`	logical; `TRUE` for two-tailed hypothesis testing, `FALSE` for one-tailed hypothesis testing.
`df`	degrees of freedom.
`ratio_tc_var`	ratio of the treatment group variance to the control group variance (needed for Satterthwaite approximation to df). Ignored when `df` argument is specified.
`rho_ic`	proportion of variance in the outcome (for treatment group) that is between intervention clusters.
`p`	proportion of units randomly assigned to the treatment condition.
`g1`	number of covariates. `g` also works.
`r21`	proportion of variance in the outcome explained by covariates (and fixed blocks if exists). `r2` also works.
`n`	sample size (number of subjects per block, if exist, calculated using simple average or harmonic mean). For partially nested design `n` is total number of subjects (in all intervention clusters and control group).
`J`	level 2 sample size (number of blocks).
`ic_size`	sample size in each intervention cluster on average.
`n0`	starting value for `n`.
`tol`	tolerance to end iterative process for finding `n` or `J`.
`z.test`	logical; `TRUE` for z-test.

Value

fun	function name.
parms	list of parameters used in power calculation.
df	degrees of freedom.
ncp	noncentrality parameter.
power	statistical power $(1 - \beta)$.
mdes	minimum detectable effect size.
n	sample size (number of subjects).

References

Dong, N., & Maynard, R. (2013). PowerUp!: A tool for calculating minimum detectable effect sizes and minimum required sample sizes for experimental and quasi-experimental design studies. *Journal of Research on Educational Effectiveness*, *6*(1), 24-67. doi: 10.1080/19345747.2012.673143

Lohr, S., Schochet, P. Z., & Sanders, E. (2014). Partially Nested Randomized Controlled Trials in Education Research: A Guide to Design and Analysis. NCER 2014-2000. National Center for Education Research. https://ies.ed.gov/ncer/pubs/20142000/pdf/20142000.pdf

Examples

```
# cross-checks
mdes.ira(n=250)
power.ira(es=.356, n=250)
mrss.ira(es=.356)

# cross-checks
mdes.bira2c1(n=15, J=20)
power.bira2c1(es=.325, n=15, J=20)
mrss.bira2c1(es=.325, J=20)

# cross-checks
mdes.bira2f1(n=15, J=20)
power.bira2f1(es=.325, n=15, J=20)
mrss.bira2f1(es=.325, J=20)

# cross-checks
mdes.ira_pn(n=250)
power.ira_pn(es=.377, n=250)
mrss.ira_pn(es=.377, n=250)
```

---

med_pn *Partially Nested Designs Probing Multilevel Mediation*

Description

Two/One partially nested design occurs when subjects are recruited into groups for the treatment condition, but the remaining participants (or waiting list) are left for the control group (without any intervention clusters). If the interest is to design a sufficiently powered study to detect mediation effect in Two/One design, use `power.med_pn21()`.

Three/One partially nested design occurs when subjects are recruited into groups, and a number of groups are guided by the same leader for the treatment condition, but the remaining participants (or waiting list) are left for the control group (without any intervention clusters). If the interest is to design a sufficiently powered study to detect mediation effect in Three/One design, use `power.med_pn31()`.

Three/Two partially nested design occurs when subjects are recruited into groups, and a number of groups are guided by the same leader for the treatment condition, however, separate from Two/One and Three/One designs, the remaining participants are nested within providers in the control group (without any intervention clusters, but with nested structure). If the interest is to design a sufficiently powered study to detect mediation effect in Three/Two design, use `power.med_pn31()`.

Usage

```
power.med_pn21(esa = .50, esB = .50, esb1 = .10,
               two.tailed = TRUE, alpha = .05,
               mc = TRUE, nsims = 1000, ndraws = 1000,
               rhom_trt = .20, rho_trt = .20,
               r2m1_ctrl = .20, r2m1_trt = .20, r2m2_trt = .20,
               r2y1z_trt = 0, r2y2z_trt = 0,
               r2y1_trt = NULL, r2y2_trt = NULL,
               g1_ctrl = 0, n_ctrl =  20,
               g2_trt = 0, n_trt = 30, J_trt = 20)

power.med_pn31(esa = .50, esB = .50, esb1 = .10,  esb2 = .10,
               two.tailed = TRUE, alpha = .05,
               mc = TRUE, nsims = 1000, ndraws = 1000,
               rhom3_trt = .20, rhom2_trt = .20,
               rho3_trt = .20, rho2_trt = .20,
               r2m1_ctrl = .20, r2m1_trt = .20, r2m2_trt = .20, r2m3_trt = .20,
               r2y1z_trt = 0, r2y2z_trt = 0, r2y3z_trt = 0,
               r2y1_trt = NULL, r2y2_trt = NULL, r2y3_trt = NULL,
               g1_ctrl = 0, n_ctrl =  20,
               g3_trt = 0, n_trt = 30, J_trt = 20, K_trt = 20)

power.med_pn32(esa = .50, esB = .50, esb1 = .10,  esb2 = .10,
               two.tailed = TRUE, alpha = .05,
               mc = TRUE, nsims = 1000, ndraws = 1000,
               rhom2_ctrl = .20, rhom3_trt = .20, rhom2_trt = .20,
               rho3_trt = .20, rho2_trt = .20,
               r2m1_ctrl = .20, r2m2_ctrl = .20,
               r2m1_trt = .20, r2m2_trt = .20, r2m3_trt = .20,
               r2y1z_trt = 0, r2y2z_trt = 0, r2y3z_trt = 0,
               r2y1_trt = NULL, r2y2_trt = NULL, r2y3_trt = NULL,
```

```
g2_ctrl = 0, n_ctrl = 30, J_ctrl = 20,
g3_trt = 0, n_trt =  20, J_trt = 20,  K_trt = 20)
```

Arguments

`esa`	Standardized mean difference in mediator values between treatment and control groups (esa = esa_trt - esa_ctrl).
`esB`	Standardized regression coefficient for mediator - outcome path, capturing level 1 and level 2 (between intervention clusters) effects (esB = esb1 + esb2 for Two/One design, or esB = esb1 + esb2 + esb3 for Three/One or Three/Two designs).
`esb1`	Standardized regression coefficient for mediator - outcome path at level 1 in the treatment group (within first intervention clusters).
`esb2`	Standardized regression coefficient for aggregate mediator - outcome path at level 2 in the treatment group (between first intervention clusters).
`alpha`	probability of type I error.
`two.tailed`	logical; `TRUE` for two-tailed hypothesis testing, `FALSE` for one-tailed hypothesis testing.
`rhom2_ctrl`	Proportion of unconditional mediator variance between level 2 units (intervention clusters) in the control group.
`rhom3_trt`	Proportion of unconditional mediator variance between level 3 units (second intervention clusters) in the treatment group.
`rhom2_trt`	Proportion of unconditional mediator variance between level 2 units (first intervention clusters) in the treatment group.
`rhom_trt`	Proportion of unconditional mediator variance between level 2 units (intervention clusters) in the treatment group.
`rho3_trt`	Proportion of unconditional outcome variance between level 3 units (second intervention clusters) in the treatment group.
`rho2_trt`	Proportion of unconditional outcome variance between level 2 units (first intervention clusters) in the treatment group.
`rho_trt`	Proportion of unconditional outcome variance between level 2 units (intervention clusters) in the control group.
`r2m1_ctrl`	Proportion of variance in the mediator explained by level 1 covariates in the control group.
`r2m2_ctrl`	Proportion of variance in the aggregate mediator explained by level 2 covariates in the control group.
`r2m1_trt`	Proportion of variance in the level 1 mediator explained by level 1 covariates in the treatment group
`r2m2_trt`	Proportion of variance in the aggregate mediator (at level 2) explained by level 2 covariates in the treatment group.
`r2m3_trt`	Proportion of variance in the aggregate mediator (at level 3) explained by level 3 covariates in the treatment group.
`r2y1z_trt`	Proportion of variance in the level 1 outcome explained by level 1 covariates in the treatment group.

r2y2z_trt	Proportion of variance in the aggregate outcome (at level 2) explained by level 2 covariates in the treatment group.
r2y3z_trt	Proportion of variance in the aggregate outcome (at level 3) explained by level 3 covariates in the treatment group.
r2y1_trt	Proportion of variance in the level 1 outcome explained by level 1 predictors (including mediator) in the treatment group.
r2y2_trt	Proportion of variance in the aggregate outcome (level 2) explained by level 2 predictors (including aggregate mediator) in the treatment group.
r2y3_trt	Proportion of variance in the aggregate outcome (level 3) explained by level 3 predictors (including aggregate mediator) in the treatment group.
g1_ctrl	Number of covariates introduced at level 1 in the control group.
g2_ctrl	Number of covariates introduced at level 2 (intervention clusters) in the control group.
n_ctrl	Total number of subjects for two/one mediation. Number of subjects per level 2 unit (intervention cluster) in the control group for Three/One and Three/Two mediation.
J_ctrl	Number of level 2 units (intervention clusters) in the control group.
g2_trt	Number of covariates introduced at level 2 (intervention clusters) in the treatment group.
g3_trt	Number of covariates introduced at level 3 (second intervention clusters) in the treatment group.
n_trt	Number of subjects per level 2 unit (first intervention cluster).
J_trt	Number of level 2 units per level 3 unit (second intervention cluster.)
K_trt	Number of level 3 units (second intervention clusters).
mc	logical; TRUE for monte carlo simulation based power.
nsims	number of replications, if mc = TRUE.
ndraws	number of draws from the distribution of the path coefficients for each replication, if mc = TRUE.

Value

fun	function name.
parms	list of parameters used in power calculation.
df	degrees of freedom.
ncp	noncentrality parameter.
power	statistical power $(1-\beta)$.

References

Kelcey, B., Bai, F., \& Xie, Y. (2020). Statistical power in partially nested designs probing multilevel mediation. *Psychotherapy Research*. Advance online publication. doi: 10.1080/10503307.2020.1717012

Examples

```
# Two/One partially nested design
power.med_pn21(esa = .40, esB = .40, esb1 = .40,
               two.tailed = TRUE, alpha = .05,
               rhom_trt = .20, rho_trt = .20,
               r2m1_ctrl = .60, r2m1_trt = .60, r2m2_trt = .60,
               r2y1_trt = .50, r2y2_trt = .50,
               n_ctrl = 50, n_trt =  6, J_trt = 50)

# Three/One partially nested design
power.med_pn31(esa = .50, esB = .50, esb1 = .10, esb2 = .10,
               rhom3_trt = .10, rhom2_trt = .20, rho3_trt = .10, rho2_trt = .20,
               r2m1_ctrl = .20, r2m1_trt = .20, r2m2_trt = .20, r2m3_trt = .20,
               r2y1_trt = .20, r2y2_trt = .20, r2y3_trt = .20,
               n_ctrl = 60, n_trt =  20, J_trt = 10, K_trt = 60)

# Three/Two partially nested design
power.med_pn32(esa = .50, esB = .50, esb1 = .10, esb2 = .10,
               rhom2_ctrl = .20, rhom3_trt = .10, rhom2_trt = .20,
               rho3_trt = .10, rho2_trt = .20,
               r2m1_ctrl = .20, r2m2_ctrl = .20,
               r2m1_trt = .20, r2m2_trt = .20, r2m3_trt = .20,
               r2y1_trt = .20, r2y2_trt = .20, r2y3_trt = .20,
               n_ctrl = 24, n_trt =  24, J_trt = 12, J_ctrl = 60, K_trt = 60)
```

---

plots *Plots*

Description

Plots statistical power, minimum detectable effect size (MDES), or MDES difference (MDESD) curves with (1-α)x100 % confidence interval.

Usage

```
  ## S3 method for class 'power'
plot(x, ypar = "power",  xpar = NULL,
          xlim = NULL, ylim = NULL,
          xlab = NULL, ylab = NULL,
          main = NULL, sub = NULL,
          locate = FALSE, ...)

  ## S3 method for class 'mdes'
plot(x, ypar = "power",  xpar = NULL,
          xlim = NULL, ylim = NULL,
          xlab = NULL, ylab = NULL,
          main = NULL, sub = NULL,
          locate = FALSE, ...)
```

```
  ## S3 method for class 'mrss'
plot(x, ypar = "power",  xpar = NULL,
          xlim = NULL, ylim = NULL,
          xlab = NULL, ylab = NULL,
          main = NULL, sub = NULL,
          locate = FALSE, ...)
```

Arguments

x	an object returned from one of the PowerUpR functions.
ypar	character; "mdes" or "power" on y axis .
xpar	character; one of the sample sizes on x axis.
xlim	limits for xpar.
ylim	limits for ypar.
xlab	x axis label (ignored for objects returned from power.med211(), power.med221(), and power.med321() functions).
ylab	y axis label (ignored for objects returned from power.med211(), power.med221(), and power.med321() functions).
main	title for the plot (ignored for objects returned from power.med221() and power.med211() functions).
sub	subtitle for the plot (ignored for objects returned from power.med221() and power.med211() functions).
locate	logical; TRUE locates parameter values for design x on the plot.
...	other graphical parameters to pass to plot.new().

Examples

```
design1 <- mdes.cra3(rho3=.06, rho2=.17, n=15, J=3, K=60)
plot(design1, ypar = "mdes", xpar = "K", xlim = c(30, 100))
plot(design1, ypar = "power", xpar = "K", xlim = c(30, 100))

design2 <- power.cra3(es=.269, rho3=.06, rho2=.17, n=15, J=3, K=60)
plot(design2, ypar = "mdes", xpar = "K", xlim = c(30, 100))
plot(design2, ypar = "power", xpar = "K", xlim = c(30, 100))
```

---

replication *Unambiguous Test of Replication for Ensemble of Studies*

Description

Use `power.rep()` for the statistical power, `mdh.rep()` for the minimum detectable heterogeneity, and `mrns.rep()` for the minimum required number of studies. Functions implement methods designed to conduct unambiguous test of replication for ensemble of studies (Hedges & Schauer, 2019). `mdh` argument is the effect heterogeneity above and beyond sampling variability. An `mdh = 0` specification means effects are same across subgroups or moderator levels in the population. Effects will vary from each other solely due to sampling error. In this case, with large samples, heterogeneity detected after ensample of studies are conducted will be equal to unity.

Usage

```
power.rep(k = 2L, mdh = 1/4, mdh.null = 0, alpha = .05)

mdh.rep(k = 2L, mdh.max = 15, alpha = .05, power = 0.80,
        mdh.null = 0, step = .001, plot = FALSE)

mrns.rep(power = .80, mdh = 1/4, mdh.null = 0, alpha = .05,
         tol = .001)
```

Arguments

`k`	number of replications.
`power`	statistical power $(1-\beta)$.
`alpha`	probability of type I error.
`mdh`	minimum detectable heterogeneity (MDH).
`mdh.null`	MDH for null hypothesis.
`mdh.max`	maximum of possible MDH values for grid search.
`step`	step size to generate possible MDH values.
`plot`	logical; if TRUE plots MDH - power curve.
`tol`	tolerance to end iterative process for finding `k`

Value

`fun`	function name.
`parms`	list of parameters used in the calculation.
`df`	degrees of freedom.
`power`	statistical power $(1-\beta)$.
`mdh`	minimum detectable heterogeneity (MDH).
`k`	minimum required number of studies.
`df`	degrees of freedom.

References

Hedges, L. V., & Schauer, J. (2019). Statistical analyses for studying replication: Meta-analytic perspectives. Psychological Methods, 24(5), 557-570. http://dx.doi.org/10.1037/met0000189

Examples

```
# cross-checks
power.rep(k = 20L, mdh = 0.50)
mdh.rep(k = 20L, power = .39)
mrns.rep(power = .39, mdh = .50)
```

---

t1t2.error *Plots Type I and Type II Error Rates*

Description

t1t2.error plots Type I (α) and Type II (β) error rates using central and noncentral t distributions for any objects returned from one of the PowerUpR functions.

Usage

```
t1t2.error(object)
```

Arguments

object an object returned from one of the PowerUpR functions.

Examples

```
  ## Not run:

    design1 <- mdes.bira2(rho2=.35, omega2=.10,
                            n=83, J=480)
    t1t2.error(design1)


## End(Not run)
```

Index