Package ‘cosa’

August 24, 2019

Type Package

Title Bound Constrained Optimal Sample Allocation

Version 2.0.0

Date 2019-08-24

Description Implements bound constrained optimal sample allocation (BCOSA) framework described in Bulus & Dong (2019) <doi:10.1080/00220973.2019.1636197> for power analysis of multilevel regression discontinuity designs (MRDDs) and multilevel randomized trials (MRTs) with continuous outcomes. Separate tools for statistical power and minimum detectable effect size computations are provided.

Imports nloptr(>= 1.0.4), msm(>= 1.6.7)

Suggests knitr, rmarkdown

VignetteBuilder knitr

License GPL (>= 3)

NeedsCompilation no

Author Metin Bulus [aut, cre],
Nianbo Dong [aut]

Maintainer Metin Bulus <bulusmetin@gmail.com>

Repository CRAN

Date/Publication 2019-08-24 16:00:02 UTC

R topics documented:

cosa-package .............................................. 2
bcrd3r2 ...................................................... 3
bcrd4r2 ...................................................... 5
bcrd4r3 ...................................................... 8
bird2 ......................................................... 11
bird3 ......................................................... 13
bird4 ......................................................... 16
cosa-deprecated ........................................... 19
crd2 ......................................................... 20
Description

Bound Constrained Optimal Sample Allocation (BCOSA) functions are designed to optimize sample sizes at one or more levels subject to budget, statistical power, or effect size constraints. BCOSA can be found in the following forms: (i) under budgetary constraints given marginal costs per unit while minimizing sampling variance of the treatment effect (or, alternatively, while maximizing power rate), (ii) under statistical power or effect size (ES) constraints while minimizing the total cost, and (iii) under sample size constraints for one or more levels along with (i) or (ii). Specifying \( \text{rhots} = 0 \) or \( \text{order} = 0 \) produces results equivalent to corresponding random assignment designs, which means there is no relationship between the treatment [random] and the score variable. Therefore, BCOSA functions also allow optimization of proportion of treatment allocation \( p \) under unequal marginal costs when primary constraint is placed on the total cost. Different starting values and algorithms may produce different results when marginal cost information is not provided and sample sizes at two or more levels and \( p \) are optimized. In such cases, experimenting different starting values and/or comparing several algorithms may facilitate decisions regarding sample sizes and \( p \).

Designs available in \texttt{cosa} package:

<table>
<thead>
<tr>
<th>Design</th>
<th>Total Levels</th>
<th>Treatment Level</th>
<th>Top Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>ird1r1</td>
<td>1</td>
<td>1</td>
<td>random</td>
</tr>
<tr>
<td>bird2r1</td>
<td>2</td>
<td>1</td>
<td>random</td>
</tr>
<tr>
<td>bird2f1</td>
<td>2</td>
<td>1</td>
<td>fixed</td>
</tr>
<tr>
<td>bird3r1</td>
<td>3</td>
<td>1</td>
<td>random</td>
</tr>
<tr>
<td>bird4r1</td>
<td>4</td>
<td>1</td>
<td>random</td>
</tr>
<tr>
<td>crd2r2</td>
<td>2</td>
<td>2</td>
<td>random</td>
</tr>
<tr>
<td>bcrd3f2</td>
<td>3</td>
<td>2</td>
<td>fixed</td>
</tr>
<tr>
<td>bcrd3r2</td>
<td>3</td>
<td>2</td>
<td>random</td>
</tr>
<tr>
<td>bcrd4r2</td>
<td>4</td>
<td>2</td>
<td>random</td>
</tr>
<tr>
<td>crd3r3</td>
<td>3</td>
<td>3</td>
<td>random</td>
</tr>
<tr>
<td>bcrd4f3</td>
<td>4</td>
<td>3</td>
<td>fixed</td>
</tr>
<tr>
<td>bcrd4r3</td>
<td>4</td>
<td>3</td>
<td>random</td>
</tr>
<tr>
<td>crd4r4</td>
<td>4</td>
<td>4</td>
<td>random</td>
</tr>
</tbody>
</table>

\texttt{ird}: individual-level regression discontinuity. \texttt{bird}: blocked individual-level regression discontinuity.
nuity. crd: cluster-level regression discontinuity. bcrd: blocked cluster-level regression discontinuity.

Design parameters follow a sequential order. Numbers at the end of a sequential parameter refers to the corresponding level. For example rho2 is the proportion of variance in the outcome between level 2 units, rho3 is the proportion of variance in the outcome between level 3 units. Similarly, r21 is the proportion of the variance in the outcome explained by level 1 covariates, r22 is the proportion of the variance in the outcome explained by level 2 covariates and so on. Similar naming conventions applies to other design parameters.

**Blocked Cluster-level Regression Discontinuity (Three-level Design, Discontinuity at Level 2)**

**Description**

Use `mdes.bcrd3r2()` to calculate minimum detectable effect size, `power.bcrd3r2()` to calculate statistical power, and `cosa.bcrd3r2()` for constrained optimal sample allocation.

**Usage**

```r
mdes.bcrd3r2(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal",
             power = .80, alpha = .05, two.tailed = TRUE, df = n3 - g3 - 1,
             rho2, rho3, omega3, r21 = 0, r22 = 0, r2t3 = 0, g3 = 0,
             rate.tp = 1, rate.cc = 0, p = .50, n1, n2, n3)
```

```r
power.bcrd3r2(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal",
               es = .25, alpha = .05, two.tailed = TRUE, df = n3 - g3 - 1,
               rho2, rho3, omega3, r21 = 0, r22 = 0, r2t3 = 0, g3 = 0,
               rate.tp = 1, rate.cc = 0, p = .50, n1, n2, n3)
```

```r
cosa.bcrd3r2(score = NULL, order = 2, rhots = NULL,
              k1 = -6, k2 = 6, dists = "normal",
              cn1 = 0, cn2 = 0, cn3 = 0, cost = NULL,
              n1 = NULL, n2 = NULL, n3 = NULL, p = NULL,
              n0 = c(10, 3, 100 + g3), p0 = .499,
              constrain = "power", round = TRUE, max.power = FALSE,
              local.solver = c("LBFGS", "SLSQP"),
              power = .80, es = .25, alpha = .05, two.tailed = TRUE,
              rho2, rho3, omega3, g3 = 0, r21 = 0, r22 = 0, r2t3 = 0)
```

**Arguments**

- **score** list; an object with class ‘score’ returned from `inspect.score()` function.
- **order** integer; order of functional form for the score variable, 0 for corresponding random assignment designs, 1 for RD design with linear score variable, 2 for RD design with linear + quadratic score variable.
rhots: correlation between the treatment and the scoring variable. Specify rhots = 0 or order = 0 to obtain results equivalent to random assignment designs.

k1: numeric; left truncation point for truncated normal dist., or lower bound for uniform dist., ignored when rhots = 0 or order = 0.

k2: numeric; right truncation point for truncated normal dist., or upper bound for uniform dist., ignored when rhots = 0 or order = 0.

dists: character; distribution of the score variable, "normal" or "uniform". By default, dists = "normal" specification implies a truncated normal distribution with k1 = -6 and k2 = 6.

power: statistical power (1 - β).

es: effect size (Cohen’s d).

alpha: probability of type I error (α).

two.tailed: logical; TRUE for two-tailed hypothesis testing.

df: degrees of freedom.

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).

omega3: ratio of the treatment effect variance between level 3 units to the variance in the outcome between level 3 units.

g3: number of covariates at level 3.

r21: proportion of level 1 variance in the outcome explained by level 1 covariates.

r22: proportion of level 1 variance in the outcome explained by level 2 covariates.

r2t3: proportion of treatment effect variance between level 3 units explained by level 3 covariates.

rate.tp: treatment group participation rate.

rate.cc: control group crossover rate.

p: proportion of level 2 units in treatment condition.

n1: average number of level 1 units per level 2 unit.

n2: average number of level 2 units per level 3 unit.

n3: number of level 3 units.

cn1: marginal cost per level 1 unit in treatment and control conditions, e.g. c(10, 5).

cn2: marginal cost per level 2 unit in treatment and control conditions, e.g. c(50, 20).

cn3: marginal cost per level 3 unit.

cost: total cost or budget.

p0: starting value for p when rhots = 0 and p = NULL. Starting value is replaced with average when p is constrained by bounds.

n0: vector of starting values for n1, n2, n3 (positional). Starting values are replaced with averages when sample sizes are constrained by bounds.

constrain: character; "cost", "power", or "mdes".

round: logical; TRUE for rounded COSA solution.

max.power: logical; TRUE for maximizing power instead of minimizing variance.

local.solver: subset of c("LBFGS", "SLSQP")
Value

- **parms**: list of parameters used in the function.
- **df**: degrees of freedom.
- **sse**: standardized standard error.
- **cosa**: constrained optimal sample allocation.
- **mdes**: minimum detectable effect size and \((1 - \alpha)\%\) confidence limits.
- **power**: statistical power \((1 - \beta)\)

Examples

```r
score.obj <- inspect.score(rnorm(10000), cutoff = 0)
power.bcrd3r2(score.obj, order = 2,
   es = 0.25, rho2 = .20, rho3 = .10, omega3 = .30,
   g3 = 0, r2t3 = 0, n1 = 20, n2 = 3, n3 = 50)

# with 5 blocks df = n3 - 2*(n blocks) - g3
# n3: number of level 3 units across five blocks
# increase in power rate due to r2t3 is made up for by reduction in df
power.bcrd3r2(score.obj, order = 2, df = 50 - 2*5 - 0,
   es = 0.25, rho2 = .20, rho3 = .10, omega3 = .30,
   g3 = 0, r2t3 = .30, n1 = 20, n2 = 3, n3 = 50)

# optimal combination of sample sizes for level 1, level 2, and level 3
# that produce power = .80 (given range restrictions for level 1 and level 2)
cosa.bcrd3r2(score.obj, order = 2,
   constrain = "power", power = .80,
   es = 0.25, rho2 = .20, rho3 = .10, omega3 = .30,
   g3 = 0, r2t3 = 0,
   n1 = c(10, 30), n2 = c(2, 5), n3 = NULL)
```

Description

Use `mdes.bcrd4r2()` to calculate minimum detectable effect size, `power.bcrd4r2()` to calculate statistical power, and use `cosa.bcrd4r2()` for constrained optimal sample allocation.

Usage

```r
mdes.bcrd4r2(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal",
   power = .80, alpha = .05, two.tailed = TRUE, df = n4 - g4 - 1,
   rho2, rho3, rho4, omega3, omega4,
   r21 = 0, r22 = 0, r2t3 = 0, r2t4 = 0, g4 = 0,
```
rate.tp = 1, rate.cc = 0, p = .50, n1, n2, n3, n4)

power.bcrd4r2(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal", 
es = .25, alpha = .05, two.tailed = TRUE, df = n4 - g4 - 1, 
rho2, rho3, rho4, omega3, omega4, 
r21 = 0, r22 = 0, r2t3 = 0, r2t4 = 0, g4 = 0, 
rate.tp = 1, rate.cc = 0, p = .50, n1, n2, n3, n4)

cosa.bcrd4r2(score = NULL, order = 2, rhots = NULL, 
k1 = -6, k2 = 6, dists = "normal", 
ct1 = 0, ct2 = 0, ct3 = 0, ct4 = 0, cost = NULL, 
n1 = NULL, n2 = NULL, n3 = NULL, n4 = NULL, p = NULL, 
n0 = c(10, 3, 100, 5 + g4), p0 = .499, 
constrain = "power", round = TRUE, max.power = FALSE, 
local.solver = c("LBFGS", "SLSQP"), 
power = .80, es = .25, alpha = .05, two.tailed = TRUE, 
rho2, rho3, rho4, omega3, omega4, 
g4 = 0, r21 = 0, r22 = 0, r2t3 = 0, r2t4 = 0)

Arguments

score list; an object with class 'score' returned from inspect.score() function.
order integer; order of functional form for the score variable, 0 for corresponding 
random assignment designs, 1 for RD design with linear score variable, 2 for 
RD design with linear + quadratic score variable
rhots correlation between the treatment and the scoring variable. Specify rhots = 0 
or order = 0 to obtain results equivalent to random assignment designs.
k1 numeric; left truncation point for truncated normal dist., or lower bound for 
uniform dist., ignored when rhots = 0 or order = 0.
k2 numeric; right truncation point for truncated normal dist., or upper bound for 
uniform dist., ignored when rhots = 0 or order = 0.
dists character; distribution of the score variable, "normal" or "uniform". By de- 
default, dists = "normal" specification implies a truncated normal distribution 
with k1 = -6 and k2 = 6.
power statistical power (1 - \beta).
es effect size (Cohen's d).
alpha probability of type I error (\alpha).
two.tailed logical; TRUE for two-tailed hypothesis testing.
df degrees of freedom.
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).
omega3 ratio of the treatment effect variance between level 3 units to the variance in the 
outcome between level 3 units.
omega4 ratio of the treatment effect variance between level 4 units to the variance in the outcome between 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 1 variance in the outcome explained by level 2 covariates.
r2t3 proportion of treatment effect variance between level 3 units explained by level 3 covariates.
r2t4 proportion of treatment effect variance between level 4 units explained by level 4 covariates.
rate.tp treatment group participation rate.
rate.cc control group crossover rate.
p proportion of level 2 units in treatment condition.
n1 average number of level 1 units per level 2 unit.
n2 average number of level 2 units per level 3 unit.
n3 average number of level 3 units per level 4 unit.
n4 number of level 4 units.
cn1 marginal cost per level 1 unit in treatment and control conditions, e.g. c(10,5).
cn2 marginal cost per level 2 unit in treatment and control conditions, e.g. c(50,20).
cn3 marginal cost per level 3 unit.
cn4 marginal cost per level 4 unit.
cost total cost or budget.
p0 starting value for p when rhots = 0 and p = NULL. Starting value is replaced with average when p is constrained by bounds.
n0 vector of starting values for n1, n2, n3, n4 (positional). Starting values are replaced with averages when sample sizes are constrained by bounds.
constrain character; "cost", "power", or "mdes".
round logical; TRUE for rounded COSA solution.
max.power logical; TRUE for maximizing power instead of minimizing variance.
local.solver subset of c("LBFGS", "SLSQP").

Value

parms list of parameters used in the function.
df degrees of freedom.
sse standardized standard error.
cosa constrained optimal sample allocation.
mdes minimum detectable effect size and (1 - α)% confidence limits.
power statistical power (1 - β)
Examples

```r
score.obj <- inspect.score(rnorm(10000), cutoff = 0)
mdes.bcrd4r2(score.obj, order = 2,
  power = .80, rho2 = .20, rho3 = .10, rho4 = .05,
  omega3 = .30, omega4 = .30,
  g4 = 0, r2t4 = 0,
  n1 = 20, n2 = 3, n3 = 20, n4 = 10)

power.bcrd4r2(score.obj, order = 2,
  es = 0.242, rho2 = .20, rho3 = .10, rho4 = .05,
  omega3 = .30, omega4 = .30,
  g4 = 0, r2t4 = 0,
  n1 = 20, n2 = 3, n3 = 20, n4 = 10)

# optimal combination of sample sizes for level 1, level 2, level 3, and level 4
# that produce power = .80 (given range restrictions for level 1 and level 2)
cosa.bcrd4r2(score.obj, order = 2,
  constrain = "power", power = .80,
  es = 0.25, rho2 = .20, rho3 = .10, rho4 = .05,
  omega3 = .30, omega4 = .30,
  g4 = 0, r2t4 = 0,
  n1 = c(10, 30), n2 = c(2, 5), n3 = NULL, n4 = NULL)
```

---

**bcrd4r3**  
*Blocked Cluster-level Regression Discontinuity (Four-level Design, Discontinuity at Level 3)*

**Description**

Use `mdes.bcrd4r3()` to calculate minimum detectable effect size, `power.bcrd4r3()` to calculate statistical power, and `cosa.bcrd4r3()` for constrained optimal sample allocation.

**Usage**

```r
mdes.bcrd4r3(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal",
  power = .80, alpha = .05, two.tailed = TRUE, df = n4 - g4 - 1,
  rho2, rho3, rho4, omega4,
  r21 = 0, r22 = 0, r23 = 0, r2t4 = 0, g4 = 0,
  rate.tp = 1, rate.cc = 0, p = .50, n1, n2, n3, n4)

power.bcrd4r3(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal",
  es = .25, alpha = .05, two.tailed = TRUE, df = n4 - g4 - 1,
  rho2, rho3, rho4, omega4,
  r21 = 0, r22 = 0, r23 = 0, r2t4 = 0, g4 = 0,
  rate.tp = 1, rate.cc = 0, p = .50, n1, n2, n3, n4)

cosa.bcrd4r3(score = NULL, order = 2, rhots = NULL,
  constrain = "power", power = .80,
  es = 0.25, rho2 = .20, rho3 = .10, rho4 = .05,
  omega3 = .30, omega4 = .30,
  g4 = 0, r2t4 = 0,
  n1 = c(10, 30), n2 = c(2, 5), n3 = NULL, n4 = NULL)
```
k1 = -6, k2 = 6, dists = "normal",
cn1 = 0, cn2 = 0, cn3 = 0, cn4 = 0, cost = NULL,
n1 = NULL, n2 = NULL, n3 = NULL, n4 = NULL, p = NULL,
n0 = c(10, 3, 100, 5 + g4), p0 = .499,
constrain = "power", round = TRUE, max.power = FALSE,
local.solver = c("LBFGS", "SLSQP"),
power = .80, es = .25, alpha = .05, two.tailed = TRUE,
rho2, rho3, rho4, omega4,
g4 = 0, r21 = 0, r22 = 0, r23 = 0, r2t4 = 0)

Arguments

score list; an object with class 'score' returned from inspect.score() function.
order integer; order of functional form for the score variable, 0 for corresponding random assignment designs, 1 for RD design with linear score variable, 2 for RD design with linear + quadratic score variable
rhots correlation between the treatment and the scoring variable. Specify rhots = 0 or order = 0 to obtain results equivalent to random assignment designs.
k1 numeric; left truncation point for truncated normal dist., or lower bound for uniform dist., ignored when rhots = 0 or order = 0.
k2 numeric; right truncation point for truncated normal dist., or upper bound for uniform dist., ignored when rhots = 0 or order = 0.
dists character; distribution of the score variable, "normal" or "uniform". By default, dists = "normal" specification implies a truncated normal distribution with k1 = -6 and k2 = 6.
power statistical power (1 - β).
es effect size (Cohen’s d).
alpha probability of type I error (α).
two.tailed logical; TRUE for two-tailed hypothesis testing.
df degrees of freedom.
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).
omega4 ratio of the treatment effect variance between level 4 units to the variance in the outcome between 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 between level 4 units explained by level 4 covariates.
rate.tp treatment group participation rate.
rate.cc control group crossover rate.

p proportion of level 3 units in treatment condition.

n1 average number of level 1 units per level 2 unit.

n2 average number of level 2 units per level 3 unit.

n3 average number of level 3 units per level 4 unit.

n4 number of level 4 units.

cn1 marginal cost per level 1 unit in treatment and control conditions, e.g. c(10, 5).

cn2 marginal cost per level 2 unit in treatment and control conditions, e.g. c(50, 20).

cn3 marginal cost per level 3 unit in treatment and control conditions, e.g. c(80, 50).

cn4 marginal cost per level 4 unit.

cost total cost or budget.

p0 starting value for p when rhots = 0 and p = NULL. Starting value is replaced with average when p is constrained by bounds.

n0 vector of starting values for n1, n2, n3, n4 (positional). Starting values are replaced with averages when sample sizes are constrained by bounds.

constrain character; "cost", "power", or "mdes".

round logical; TRUE for rounded COSA solution.

max.power logical; TRUE for maximizing power instead of minimizing variance.

local.solver subset of c("LBFGS", "SLSQP").

### Value

parms list of parameters used in the function.

df degrees of freedom.

sse standardized standard error.

cosa constrained optimal sample allocation.

mdes minimum detectable effect size and (1 - α)% confidence limits.

power statistical power (1 - β)

### Examples

```r
score.obj <- inspect.score(rnorm(10000), cutoff = 0)
mdes.bcrd4r3(score.obj, order = 2,
              power = .80, rho2 = .20, rho3 = .10, rho4 = .05,
              omega4 = .30, g4 = 0, r2t4 = 0,
              n1 = 20, n2 = 3, n3 = 20, n4 = 10)

power.bcrd4r3(score.obj, order = 2,
               es = 0.334, rho2 = .20, rho3 = .10, rho4 = .05,
               omega4 = .30, g4 = 0, r2t4 = 0,
               n1 = 20, n2 = 3, n3 = 20, n4 = 10)
```
# optimal combination of sample sizes for level 1, level 2, level 3, and level 4
# that produce power = .80 (given range restrictions for level 1, level 2, and level 4)
cosa.bcrd4r3(score.obj, order = 2,
    constrain = "power", power = .80,
    es = 0.25, rho2 = .20, rho3 = .10, rho4 = .05,
    omega4 = .30, g4 = 0, r2t4 = 0,
    n1 = c(10, 30), n2 = c(2, 5),
    n3 = NULL, n4 = c(3, 10))

Description

Use mdes.bird2() to calculate minimum detectable effect size, power.bird2() to calculate statistical power, and cosa.bird2() for constrained optimal sample allocation. To consider fixed block effects, modify degrees of freedom in <output>.bird2() functions as \( n_2 - 2*nb - g_2 \) where \( n_2 \) is total number of level 2 units across blocks, and \( nb \) is number of blocks. Keep in mind that \( r_2t_2 \) now includes information about blocks, but this fact will not be reflected in \( g_2 \). See examples below.

Usage

mdes.bird2(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal",
    power = .80, alpha = .05, two.tailed = TRUE, df = n2 - g2 - 1,
    rho2, omega2, r21 = 0, r2t2 = 0, g2 = 0,
    rate.tp = 1, rate.cc = 0, p = .50, n1, n2)

power.bird2(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal",
    es = .25, alpha = .05, two.tailed = TRUE, df = n2 - g2 - 1,
    rho2, omega2, r21 = 0, r2t2 = 0, g2 = 0,
    rate.tp = 1, rate.cc = 0, p = .50, n1, n2)

cosa.bird2(score = NULL, order = 2, rhots = NULL,
    k1 = -6, k2 = 6, dists = "normal",
    cn1 = 0, cn2 = 0, cost = NULL,
    n1 = NULL, n2 = NULL, p = NULL,
    n0 = c(10, 100 + g2), p0 = .499,
    constrain = "power", round = TRUE, max.power = FALSE,
    local.solver = c("LBFGS", "SLSQP"),
    power = .80, es = .25, alpha = .05, two.tailed = TRUE,
    rho2, omega2, g2 = 0, r21 = 0, r2t2 = 0)

Arguments

score list; an object with class 'score' returned from inspect.score() function.
order
integer; order of functional form for the score variable, 0 for corresponding random assignment designs, 1 for RD design with linear score variable, 2 for RD design with linear + quadratic score variable.

rhots
correlation between the treatment and the scoring variable. Specify rhots = 0 or order = 0 to obtain results equivalent to random assignment designs.

k1
numeric; left truncation point for truncated normal dist., or lower bound for uniform dist., ignored when rhots = 0 or order = 0.

k2
numeric; right truncation point for truncated normal dist., or upper bound for uniform dist., ignored when rhots = 0 or order = 0.

dists
character; distribution of the score variable, "normal" or "uniform". By default, dists = "normal" specification implies a truncated normal distribution with k1 = -6 and k2 = 6.

power
statistical power (1 - β).

es
effect size (Cohen’s d).

alpha
probability of type I error (α).

two.tailed
logical; TRUE for two-tailed hypothesis testing.

df
degrees of freedom.

rho2
proportion of variance in the outcome between level 2 units (unconditional ICC2).

omega2
ratio of the treatment effect variance between level 2 units to the variance in the outcome between level 2 units.

g2
number of covariates at level 2.

r21
proportion of level 1 variance in the outcome explained by level 1 covariates.

r2t2
proportion of treatment effect variance between level 2 units explained by level 2 covariates.

rate.tp
treatment group participation rate.

rate.cc
control group crossover rate.

p
proportion of level 1 units in treatment condition.

n1
average number of level 1 units per level 2 units.

n2
number of level 2 units.

cn1
marginal cost per level 1 unit in treatment and control conditions, e.g. c(10,5).

cn2
marginal cost per level 2 unit.

cost
total cost or budget.

p0
starting value for p when rhots = 0 and p = NULL. Starting value is replaced with average when p is constrained by bounds.

n0
vector of starting values for n1, n2 (positional). Starting values are replaced with averages when sample sizes are constrained by bounds.

constrain
character; "cost", "power", or "mdes".

round
logical; TRUE for rounded COSA solution.

max.power
logical; TRUE for maximizing power instead of minimizing variance.

local.solver
subset of c("LBFGS","SLSQP")
Value

- **params**: list of parameters used in the function.
- **df**: degrees of freedom.
- **sse**: standardized standard error.
- **cosa**: constrained optimal sample allocation.
- **mdes**: minimum detectable effect size and \((1 - \alpha)\%\) confidence limits.
- **power**: statistical power \((1 - \beta)\)

Examples

```r
score.obj <- inspect.score(rnorm(10000), cutoff = 0)
power.bird2(score.obj, order = 2, 
es = 0.25, rho2 = .20, omega2 = .30, 
g2 = 0, r2t2 = 0, n1 = 50, n2 = 30)
```

# with 5 blocks df = n2 - 2*(n blocks) - g2
# n2: number of level 2 units across five blocks
```r
power.bird2(score.obj, order = 2, df = 100 - 2*5 - 0, 
es = 0.25, rho2 = .20, omega2 = .30, 
g2 = 0, r2t2 = .30, n1 = 50, n2 = 30)
```

# optimal combination of sample sizes for level 1 and level 2
# around 20 and 50 respectively, that produce power = .80
```r
cosa.bird2(score.obj, order = 2, 
  constrain = "power", power = .80, 
es = 0.25, rho2 = .20, omega2 = .30, 
g2 = 0, r2t2 = 0, 
n0 = c(20, 50), n1 = NULL, n2 = NULL)
```

Description

Use `mdes.bird3()` to calculate minimum detectable effect size, `power.bird3()` to calculate statistical power, and `cosa.bird3()` for constrained optimal sample allocation. To consider fixed block effects, modify degrees of freedom in `<output>.bird3()` functions as \(n3 - 2*nb - g3\) where \(n3\) is total number of level 3 units across blocks, and \(nb\) is number of blocks. Keep in mind that \(r2t3\) now includes information about blocks, but this fact will not be reflected in \(g3\). See examples below.
Usage

mdes.bird3(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal",
            power = .80, alpha = .05, two.tailed = TRUE, df = n3 - g3 - 1,
            rho2, rho3, omega2, omega3, r21 = 0, r2t2 = 0, r2t3 = 0, g3 = 0,
            rate.tp = 1, rate.cc = 0, p = .50, n1, n2, n3)

power.bird3(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal",
            es = .25, alpha = .05, two.tailed = TRUE, df = n3 - g3 - 1,
            rho2, rho3, omeg2a, omeg3a, r21 = 0, r2t2 = 0, r2t3 = 0, g3 = 0,
            rate.tp = 1, rate.cc = 0, p = .50, n1, n2, n3)

cosa.bird3(score = NULL, order = 2, rhots = NULL,
            k1 = -6, k2 = 6, dists = "normal",
            cn1 = 0, cn2 = 0, cn3 = 0, cost = NULL,
            n1 = NULL, n2 = NULL, n3 = NULL, p = NULL,
            n0 = c(10, 3, 100 + g3), p0 = .499,
            constrain = "power", round = TRUE, max.power = FALSE,
            local.solver = c("LBFGS", "SLSQP"),
            power = .80, es = .25, alpha = .05, two.tailed = TRUE,
            rho2, rho3, omega2, omega3,
            g3 = 0, r21 = 0, r2t2 = 0, r2t3 = 0)

Arguments

score list; an object with class 'score' returned from `inspect.score()` function.

order integer; order of functional form for the score variable, 0 for corresponding
random assignment designs, 1 for RD design with linear score variable, 2 for
RD design with linear + quadratic score variable

rhots correlation between the treatment and the scoring variable. Specify rhots = 0
or order = 0 to obtain results equivalent to random assignment designs.

k1 numeric; left truncation point for truncated normal dist., or lower bound for
uniform dist., ignored when rhots = 0 or order = 0.

k2 numeric; right truncation point for truncated normal dist., or upper bound for
uniform dist., ignored when rhots = 0 or order = 0.

dists character; distribution of the score variable, "normal" or "uniform". By de-
default, dists = "normal" specification implies a truncated normal distribution
with k1 = -6 and k2 = 6.

power statistical power (1 - β).

es effect size (Cohen’s d).

alpha probability of type I error (α).

two.tailed logical; TRUE for two-tailed hypothesis testing.

df degrees of freedom.

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).
omega2  ratio of the treatment effect variance between level 2 units to the variance in the outcome between level 2 units.

omega3  ratio of the treatment effect variance between level 3 units to the variance in the outcome between level 3 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 between level 2 units explained by level 2 covariates.

r2t3  proportion of treatment effect variance between level 3 units explained by level 3 covariates.

rate.tp  treatment group participation rate.

rate.cc  control group crossover rate.

p  proportion of level 1 units in treatment condition.

n1  average number of level 1 units per level 2 unit.

n2  average number of level 2 units per level 3 unit.

n3  number of level 3 units.

cn1  marginal cost per level 1 unit in treatment and control conditions, e.g. c(10,5).

cn2  marginal cost per level 2 unit.

cn3  marginal cost per level 3 unit.

cost  total cost or budget.

p0  starting value for p when rhots = 0 and p = NULL. Starting value is replaced with average when p is constrained by bounds.

n0  vector of starting values for n1, n2, n3 (positional). Starting values are replaced with averages when sample sizes are constrained by bounds.

constrain  character; "cost", "power", or "mdes".

round  logical; TRUE for rounded COSA solution.

max.power  logical; TRUE for maximizing power instead of minimizing variance.

local.solver  subset of c("LBFGS", "SLSQP")

Value

parms  list of parameters used in the function.

df  degrees of freedom.

sse  standardized standard error.

cosa  constrained optimal sample allocation.

mdes  minimum detectable effect size and (1 - α)% confidence limits.

power  statistical power (1 - β)
Examples

```r
score.obj <- inspect.score(rnorm(10000), cutoff = 0)

power.bird3(score.obj, order = 2,
            es = 0.25, rho2 = .20, rho3 = .10,
            omega2 = .30, omega3 = .30,
            g3 = 0, r2t3 = 0, n1 = 50, n2 = 3, n3 = 15)
```

# with 5 blocks df = n3 - 2*(n blocks) - g3  
# n3: number of level 3 units across five blocks  
# increase in r2t3 does not make up for reduction in df

```r
power.bird3(score.obj, order = 2, df = 15 - 2*5 - 0,
            es = 0.25, rho2 = .20, rho3 = .10,
            omega2 = .30, omega3 = .30,
            g3 = 0, r2t3 = .30, n1 = 50, n2 = 3, n3 = 15)
```

# optimal combination of sample sizes for level 1, level 2 and level 3  
# that produce power = .80 (given range restrictions)

```r
cosa.bird3(score.obj, order = 2,
            constrain = "power", power = .80,
            es = 0.25, rho2 = .20, rho3 = .10,
            omega2 = .30, omega3 = .30,
            g3 = 0, r2t3 = 0,
            n1 = c(15,30), n2 = c(3, 5), n3 = c(10,30))
```

---

**bird4**  
*Blocked Individual-level Regression Discontinuity (Four-level Design, Discontinuity at Level 1)*

---

**Description**

Use `mdes.bird4()` to calculate minimum detectable effect size, `power.bird4()` to calculate statistical power, and `cosa.bird4()` for constrained optimal sample allocation.

**Usage**

```r
mdes.bird4(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal",
           power = .80, alpha = .05, two.tailed = TRUE, df = n4 - g4 - 1,
           rho2, rho3, rho4, omega2, omega3, omega4,
           r21 = 0, r2t2 = 0, r2t3 = 0, r2t4 = 0, g4 = 0,
           rate.tp = 1, rate.cc = 0, p = .50, n1, n2, n3, n4)
```

```r
power.bird4(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal",
             es = .25, alpha = .05, two.tailed = TRUE, df = n4 - g4 - 1,
             rho2, rho3, rho4, omega2, omega3, omega4,
             r21 = 0, r2t2 = 0, r2t3 = 0, r2t4 = 0, g4 = 0,
             rate.tp = 1, rate.cc = 0, p = .50, n1, n2, n3, n4)
```
cosa.bird4(score = NULL, order = 2, rhots = NULL, 
    k1 = -6, k2 = 6, dists = "normal", 
    cn1 = 0, cn2 = 0, cn3 = 0, cn4 = 0, cost = NULL, 
    n1 = NULL, n2 = NULL, n3 = NULL, n4 = NULL, p = NULL, 
    n0 = c(10, 3, 100, 5 + g4), p0 = .499, 
    constrain = "power", round = TRUE, max.power = FALSE, 
    local.solver = c("LBFGS", "SLSQP"), 
    power = .80, es = .25, alpha = .05, two.tailed = TRUE, 
    rho2, rho3, rho4, omega2, omega3, omega4, 
    g4 = 0, r21 = 0, r2t2 = 0, r2t3 = 0, r2t4 = 0)

Arguments

score list; an object with class ‘score’ returned from inspect.score() function.

order integer; order of functional form for the score variable, 0 for corresponding random assignment designs, 1 for RD design with linear score variable, 2 for RD design with linear + quadratic score variable

rhots correlation between the treatment and the scoring variable. Specify rhots = 0 or order = 0 to obtain results equivalent to random assignment designs.

k1 numeric; left truncation point for truncated normal dist., or lower bound for uniform dist., ignored when rhots = 0 or order = 0.

k2 numeric; right truncation point for truncated normal dist., or upper bound for uniform dist., ignored when rhots = 0 or order = 0.

dists character; distribution of the score variable, "normal" or "uniform". By default, dists = "normal" specification implies a truncated normal distribution with k1 = -6 and k2 = 6.

power statistical power (1 - β).

es effect size (Cohen’s d).

alpha probability of type I error (α).

two.tailed logical; TRUE for two-tailed hypothesis testing.

df degrees of freedom.

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).

omega2 ratio of the treatment effect variance between level 2 units to the variance in the outcome between level 2 units.

omega3 ratio of the treatment effect variance between level 3 units to the variance in the outcome between level 2 units.

omega4 ratio of the treatment effect variance between level 4 units to the variance in the outcome between level 4 units.

g4 number of covariates at level 4.

r21 proportion of level 1 variance in the outcome explained by level 1 covariates.
\textbf{r2t2} proportion of treatment effect variance between level 2 units explained by level 2 covariates.

\textbf{r2t3} proportion of treatment effect variance between level 3 units explained by level 3 covariates.

\textbf{r2t4} proportion of treatment effect variance between level 4 units explained by level 4 covariates.

\textbf{rate.tp} treatment group participation rate.

\textbf{rate.cc} control group crossover rate.

\textbf{p} proportion of level 1 units in treatment condition.

\textbf{n1} average number of level 1 units per level 2 unit.

\textbf{n2} average number of level 2 units per level 3 unit.

\textbf{n3} average number of level 3 units per level 4 unit.

\textbf{n4} number of level 4 units.

\textbf{cn1} marginal cost per level 1 unit in treatment and control conditions.

\textbf{cn2} marginal cost per level 2 unit.

\textbf{cn3} marginal cost per level 3 unit.

\textbf{cn4} marginal cost per level 4 unit.

\textbf{cost} total cost or budget.

\textbf{p0} starting value for \( p \) when \( \rho_{\text{hots}} = 0 \) and \( p = \text{NULL} \). Starting value is replaced with average when \( p \) is constrained by bounds.

\textbf{n0} vector of starting values for \( n1, n2, n3, n4 \) (positional). Starting values are replaced with averages when sample sizes are constrained by bounds.

\textbf{constrain} character; "cost", "power", or "mdes".

\textbf{round} logical; TRUE for rounded COSA solution.

\textbf{max.power} logical; TRUE for maximizing power instead of minimizing variance.

\textbf{local.solver} subset of c("LBFGS", "SLSQP").

\textbf{Value}

\textbf{parms} list of parameters used in the function.

\textbf{df} degrees of freedom.

\textbf{sse} standardized standard error.

\textbf{cosa} constrained optimal sample allocation.

\textbf{mdes} minimum detectable effect size and \((1 - \alpha)\%\) confidence limits.

\textbf{power} statistical power \((1 - \beta)\).
Examples

```r
score.obj <- inspect.score(rnorm(10000), cutoff = 0)
mdes.bird4(score.obj, order = 2,
  power = .80, rho2 = .20, rho3 = .10, rho4 = .05,
  omega2 = .30, omega3 = .30, omega4 = .30,
  g4 = 0, r2t4 = 0, n1 = 20, n2 = 3, n3 = 20, n4 = 10)

power.bird4(score.obj, order = 2,
  es = .152, rho2 = .20, rho3 = .10, rho4 = .05,
  omega2 = .30, omega3 = .30, omega4 = .30,
  g4 = 0, r2t4 = 0, n1 = 20, n2 = 3, n3 = 20, n4 = 10)

# optimal combination of sample sizes for level 1, level 2, level 3 and level 4
# that produce power = .80 (given range restrictions)
cosa.bird4(score.obj, order = 2,
  constrain = "power", power = .80,
  es = .25, rho2 = .20, rho3 = .10, rho4 = .05,
  omega2 = .30, omega3 = .30, omega4 = .30,
  g4 = 0, r2t4 = 0,
  n1 = c(15, 30), n2 = c(2, 5),
  n3 = c(10, 30), n4 = c(5, 20))
```

Description

Some function are renamed and depreciated. They may be removed in the future.

Details

Depreciated function names:

- `power.crd2r2` is depreciated, use `power.crd2` instead.
- `mdes.crd2r2` is depreciated, use `mdes.crd2` instead.
- `cosa.crd2r2` is depreciated, use `cosa.crd2` instead.
- `power.crd3r3` is depreciated, use `power.crd3` instead.
- `mdes.crd3r3` is depreciated, use `mdes.crd3` instead.
- `cosa.crd3r3` is depreciated, use `cosa.crd3` instead.
- `power.crd4r4` is depreciated, use `power.crd4` instead.
- `mdes.crd4r4` is depreciated, use `mdes.crd4` instead.
- `cosa.crd4r4` is depreciated, use `cosa.crd4` instead.
- `power.ira1r1` is depreciated, use `power.ira` instead.
- `mdes.ira1r1` is depreciated, use `mdes.ira` instead.
- `power.bira2r1` is depreciated, use `power.bira2` instead.
• mdes.bira2r1 is depreciated, use mdes.bira2 instead.
• cosa.bira2r1 is depreciated, use cosa.bira2 instead.
• power.bira3r1 is depreciated, use power.bira3 instead.
• mdes.bira3r1 is depreciated, use mdes.bira3 instead.
• cosa.bira3r1 is depreciated, use cosa.bira3 instead.
• power.bira4r1 is depreciated, use power.bira4 instead.
• mdes.bira4r1 is depreciated, use mdes.bira4 instead.
• cosa.bira4r1 is depreciated, use cosa.bira4 instead.

**Description**

Use mdes.crd2() to calculate minimum detectable effect size, power.crd2() to calculate statistical power, and cosa.crd2() for constrained optimal sample allocation. If higher level strata or fixed blocks exist, use mdes.bcrd3f2() to calculate minimum detectable effect size, power.bcrd3f2() to calculate statistical power, and cosa.bcrd3f2() for constrained optimal sample allocation. Alternatively modify degrees of freedom in <output>.crd2() functions as $n_2 - 2n_b - g_2 - \text{order}$ where $n_2$ is total number of level 2 units across blocks, $n_b$ is number of blocks. Keep in mind that $r_{22}$ now includes information about blocks, but this fact will not be reflected in $g_2$. See examples below.

**Usage**

mdes.crd2(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal",
           power = .80, alpha = .05, two.tailed = TRUE, df = n2 - g2 - \text{order} - 2,
           rho2, r21 = 0, r22 = 0, g2 = 0, rate.tp = 1, rate.cc = 0, p = .50, n1, n2)

power.crd2(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal",
           es = .25, alpha = .05, two.tailed = TRUE, df = n2 - g2 - \text{order} - 2,
           rho2, r21 = 0, r22 = 0, g2 = 0, rate.tp = 1, rate.cc = 0, p = .50, n1, n2)

cosa.crd2(score = NULL, order = 2, rhots = NULL,
          k1 = -6, k2 = 6, dists = "normal",
          cn1 = 0, cn2 = 0, cost = NULL,
          n1 = NULL, n2 = NULL, p = NULL,
          n0 = c(10, 100 + g2 + \text{order}), p0 = .499,
          constrain = "power", round = TRUE,
          max.power = FALSE, local.solver = c("LBFGS", "SLSQP"),
          power = .80, es = .25, alpha = .05, two.tailed = TRUE,
          rho2, g2 = 0, r21 = 0, r22 = 0)

mdes.bcrd3f2(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal",
             n2 - 2n_b - g_2 - \text{order} - 2,
             rho2, g2 = 0, r21 = 0, r22 = 0)
power = .80, alpha = .05, two.tailed = TRUE, df = n3 * (n2 - 2) - g2 - order,
    rho2, r21 = 0, r22 = 0, g2 = 0,
    rate.tp = 1, rate.cc = 0, p = .50, n1, n2, n3)

power.bcrd3f2(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal",
    es = .25, alpha = .05, two.tailed = TRUE, df = n3 * (n2 - 2) - g2 - order,
    rho2, r21 = 0, r22 = 0, g2 = 0,
    rate.tp = 1, rate.cc = 0, p = .50, n1, n2, n3)

cosa.bcrd3f2(score = NULL, order = 2, rhots = NULL,
    k1 = -6, k2 = 6, dists = "normal",
    cn1 = 0, cn2 = 0, cn3 = 0, cost = NULL,
    n1 = NULL, n2 = NULL, n3 = NULL, p = NULL,
    n0 = c(10, 100 + g2, 5), p0 = .499,
    constrain = "power", round = TRUE, max.power = FALSE,
    local.solver = c("LBFGS", "SLSQP"),
    power = .80, es = .25, alpha = .05, two.tailed = TRUE,
    rho2, g2 = 0, r21 = 0, r22 = 0)

Arguments

score list; an object with class 'score' returned from inspect.score() function.

order integer; order of functional form for the score variable, 0 for corresponding
    random assignment designs, 1 for RD design with linear score variable, 2 for
    RD design with linear + quadratic score variable

rhots correlation between the treatment and the scoring variable. Specify rhots = 0
    or order = 0 to obtain results equivalent to random assignment designs.

k1 numeric; left truncation point for truncated normal dist., or lower bound for
    uniform dist., ignored when rhots = 0 or order = 0.

k2 numeric; right truncation point for truncated normal dist., or upper bound for
    uniform dist., ignored when rhots = 0 or order = 0.

dists character; distribution of the score variable, "normal" or "uniform". By de-
    fault, dists = "normal" specification implies a truncated normal distribution
    with k1 = -6 and k2 = 6.

power statistical power (1 - β).

es effect size (Cohen’s d).

alpha probability of type I error (α).

two.tailed logical; TRUE for two-tailed hypothesis testing.

df degrees of freedom.

rho2 proportion of variance in the outcome between level 2 units (unconditional ICC2).

g2 number of covariates at level 2.

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.

rate.tp treatment group participation rate.
rate.cc  control group crossover rate.
\( p \)  proportion of level 2 units in treatment condition.
\( n_1 \)  average number of level 1 units per level 2 unit.
\( n_2 \)  number of level 2 units (per stratum or block, if exists).
\( n_3 \)  number of stratum or fixed blocks.
\( cn_1 \)  marginal costs per level 1 unit in treatment and control conditions, e.g. \( c(10,5) \).
\( cn_2 \)  marginal costs per level 2 unit in treatment and control conditions, e.g. \( c(50,30) \).
\( cn_3 \)  marginal costs per stratum or fixed block.
\( \text{cost} \)  total cost or budget.
\( n_0 \)  vector of starting values for \( n_1, n_2 \) or \( n_1, n_2, n_3 \) (positional). Starting values are replaced with averages when sample sizes are constrained by bounds.
\( p_0 \)  starting value for \( p \) when \( \rho_{hts} = 0 \) or \( \text{order} = 0 \), and \( p = \text{NULL} \). Starting value is replaced with average when \( p \) is constrained by bounds.
\( \text{constrain} \)  character; "cost", "power", or "es".
\( \text{round} \)  logical; \( \text{TRUE} \) for rounded COSA solution.
\( \text{max.power} \)  logical; \( \text{TRUE} \) for maximizing power instead of minimizing variance, applies when \( \text{constrain} = \text{"cost"} \).
\( \text{local.solver} \)  subset of \( \text{c("LBFGS","SLSQP")} \).

**Value**

- **parms**  list of parameters used in the function.
- **df**  degrees of freedom.
- **sse**  standardized standard error.
- **cosa**  constrained optimal sample allocation.
- **mdes**  minimum detectable effect size and \( (1 - \alpha)\% \) confidence limits.
- **power**  statistical power \( (1 - \beta) \).

**Examples**

```r
score.obj <- inspect.score(rnorm(10000), cutoff = 0)
power.crd2(score.obj, order = 2,
  es = .25, rho2 = .20, g2 = 0, r22 = 0,
  n1 = 50, n2 = 100)

# with 5 blocks \( \text{df} = n_2 - 2 \times (n \text{ blocks}) - \text{order} - g_2 \)
# \( n_2 \): number of level 2 units across five blocks
power.crd2(score.obj, order = 2, df = 100 - 2*5 - 2 - 0,
  es = .25, rho2 = .20, g2 = 0, r22 = .30,
  n1 = 50, n2 = 100)

# compare
# \( n_2 \): number of level 2 units per block, \( n_3 \): number of blocks
power.bcrd3f2(score.obj, order = 2,
```

crd3

Cluster-level Regression Discontinuity (Three-level Design, Discontinuity at Level 3, w/ or w/o Strata or Fixed Blocks)

Description

Use mdes.crd3() to calculate minimum detectable effect size, power.crd3() to calculate statistical power, and cosa.crd3() for constrained optimal sample allocation. If higher level strata or fixed blocks exist, use mdes.bcrd4f3() to calculate minimum detectable effect size, power.bcrd4f3() to calculate statistical power, and cosa.bcrd4f3() for constrained optimal sample allocation. Alternatively modify degrees of freedom in <output>.crd3() functions as n3-2*nb-g3-order where n3 is total number of level 3 units across blocks, and nb is number of blocks. Keep in mind that r23 now includes information about blocks, but this fact will not be reflected in g3. See examples below.

Usage

mdes.crd3(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal", power = .80, alpha = .05, two.tailed = TRUE, df = n3 - g3 - order - 2, rho2, rho3, r21 = 0, r22 = 0, r23 = 0, g3 = 0, rate.tp = 1, rate.cc = 0, p = .50, n1, n2, n3)

power.crd3(score = NULL, order = 2, rhots = NULL, es = .25, alpha = .05, two.tailed = TRUE, df = n3 - g3 - order - 2, rho2, rho3, r21 = 0, r22 = 0, r23 = 0, g3 = 0, rate.tp = 1, rate.cc = 0, p = .50, n1, n2, n3)

cosa.crd3r3(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal", cn1 = 0, cn2 = 0, cn3 = 0, cost = NULL, n1 = NULL, n2 = NULL, n3 = NULL, p = NULL, n0 = c(10, 3, 100 + g3 + order), p0 = .499, constrain = "power", round = TRUE, max.power = FALSE, local.solver = c("LBFGS", "SLSQP"), power = .80, es = .25, alpha = .05, two.tailed = TRUE, rho2, rho3, g3 = 0, r21 = 0, r22 = 0, r23 = 0)

mdes.bcrd4f3(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal",
power = .80, alpha = .05, two.tailed = TRUE, df = n4 * (n3 - 2) - g3 - order,
 rho2, rho3, r21 = 0, r22 = 0, r23 = 0, g3 = 0,
 rate.tp = 1, rate.cc = 0, p = .50, n1, n2, n3, n4)

power.bcrd4f3(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal",
 es = .25, alpha = .05, two.tailed = TRUE, df = n4 * (n3 - 2) - g3 - order,
 rho2, rho3, r21 = 0, r22 = 0, r23 = 0, g3 = 0,
 rate.tp = 1, rate.cc = 0, p = .50, n1, n2, n3, n4)

cosa.bcrd4f3(score = NULL, order = 2, rhots = NULL,
 k1 = -6, k2 = 6, dists = "normal",
 cn1 = 0, cn2 = 0, cn3 = 0, cn4 = 0, cost = NULL,
 n1 = NULL, n2 = NULL, n3 = NULL, n4 = NULL,
 p = NULL, n0 = c(10, 3, 100 + g3, 5), p0 = .499,
 constrain = "power", round = TRUE, max.power = FALSE,
 local.solver = c("LBFGS", "SLSQP"),
 power = .80, es = .25, alpha = .05, two.tailed = TRUE,
 rho2, rho3, g3 = 0, r21 = 0, r22 = 0, r23 = 0)

Arguments

- **score**: list; an object with class 'score' returned from inspect.score() function.
- **order**: integer; order of functional form for the score variable, 0 for corresponding random assignment designs, 1 for RD design with linear score variable, 2 for RD design with linear + quadratic score variable.
- **rhots**: correlation between the treatment and the scoring variable. Specify rhots = 0 or order = 0 to obtain results equivalent to random assignment designs.
- **k1**: numeric; left truncation point for truncated normal dist., or lower bound for uniform dist., ignored when rhots = 0 or order = 0.
- **k2**: numeric; right truncation point for truncated normal dist., or upper bound for uniform dist., ignored when rhots = 0 or order = 0.
- **dists**: character; distribution of the score variable, "normal" or "uniform". By default, dists = "normal" specification implies a truncated normal distribution with k1 = -6 and k2 = 6.
- **power**: statistical power (1 - β).
- **es**: effect size (Cohen’s d).
- **alpha**: probability of type I error (α).
- **two.tailed**: logical; TRUE for two-tailed hypothesis testing.
- **df**: degrees of freedom.
- **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).
- **g3**: number of covariates at level 3.
- **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.
proportion of level 3 variance in the outcome explained by level 3 covariates.

treatment group participation rate.

control group crossover rate.

proportion of level 3 units in treatment condition.

average number of level 1 units per level 2 unit.

average number of level 2 units per level 3 unit.

number of level 3 units(per stratum or block, if exists).

number of stratum or fixed blocks.

marginal cost per level 1 unit in treatment and control conditions, e.g. c(10, 5).

marginal cost per level 2 unit in treatment and control conditions, e.g. c(50, 30).

marginal cost per level 3 unit in treatment and control conditions, e.g. c(80, 50).

marginal cost per stratum or fixed block.

total cost or budget.

starting value for \( p \) when \( \rho = 0 \) and \( p = \text{NULL} \). Starting value is replaced with average when \( p \) is constrained by bounds.

vector of starting values for \( n_1, n_2, n_3 \) or \( n_1, n_2, n_3, n_4 \) (positional). Starting values are replaced with averages when sample sizes are constrained by bounds.

character; "cost", "power", or "mdes".

logical; \( \text{TRUE} \) for rounded COSA solution.

logical; \( \text{TRUE} \) for maximizing power instead of minimizing variance.

subset of c("LBFGS", "SLSQP")

list of parameters used in the function.

degrees of freedom.

standardized standard error.

constrained optimal sample allocation.

minimum detectable effect size and \( (1 - \alpha)\% \) confidence limits.

statistical power \( (1 - \beta) \)

\[
\text{score.obj} \leftarrow \text{inspect.score(rnorm(10000), cutoff = 0)}
\]

\[
\text{power.crd3(score.obj, order = 2, es = .25, rho2 = .20, rho3 = .10, g3 = 0, r23 = 0, n1 = 20, n2 = 3, n3 = 100)}
\]

# with 5 blocks \( df = n_3 - 2 \times (n \text{ blocks}) - \text{order} - g3 \)
# \( n_3 \): number of level 3 units across five blocks

\[
\text{power.crd3(score.obj, order = 2, df = 100 - 2 \times 5 - 2 - 0, es = .25, rho2 = .20, rho3 = .10, g3 = 0, r23 = .30, n1 = 20, n2 = 3, n3 = 100)}
\]
# compare
# n3: number of level 3 units per block, n4: number of blocks
power.bcrd4f3(score.obj, order = 2,
    es = .25, rho2 = .20, rho3 = .10,
    g3 = 0, r23 = .30,
    n1 = 20, n2 = 3, n3 = 20, n4 = 5)

# optimal combination of sample sizes for level 1 and level 3
# that produce power = .80 (given range restriction for level 1 sample size)
cosa.bcrd4f3(score.obj, order = 2,
    constrain = "power", power = .80,
    es = .25, rho2 = .20, rho3 = .10, g3 = 0, r23 = .30,
    n1 = c(20, 60), n2 = 2, n3 = NULL, n4 = 5)

crd4

Cluster-level Regression Discontinuity (Four-level Design, Discontinuity at Level 4)

Description

Use mdes.crd4() to calculate minimum detectable effect size, power.crd4() to calculate statistical power, and cosa.crd4() for constrained optimal sample allocation.

Usage

mdes.crd4(power = .80, alpha = .05, two.tailed = TRUE, df = n4 - g4 - order - 2,
    score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal",
    rho2, rho3, rho4, r21 = 0, r22 = 0, r23 = 0, r24 = 0,
    g4 = 0, rate.tp = 1, rate.cc = 0, p = .50, n1, n2, n3, n4)

describe.crd4(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal",
    es = .25, alpha = .05, two.tailed = TRUE, df = n4 - g4 - order - 2,
    rho2, rho3, rho4, r21 = 0, r22 = 0, r23 = 0, r24 = 0,
    g4 = 0, rate.tp = 1, rate.cc = 0, p = .50, n1, n2, n3, n4)

cosa.crd4(score = NULL, order = 2, rhots = NULL,
    k1 = -6, k2 = 6, dists = "normal",
    cn1 = 0, cn2 = 0, cn3 = 0, cn4 = 0, cost = NULL,
    n1 = NULL, n2 = NULL, n3 = NULL, n4 = NULL, p = NULL,
    n0 = c(10, 3, 100, 5 + g4 + order), p0 = .499,
    constrain = "power", round = TRUE, max.power = FALSE,
    local.solver = c("LBFGS", "SLSQP"),
    power = .80, es = .25, alpha = .05, two.tailed = TRUE,
    rho2, rho3, rho4, g4 = 0, r21 = 0, r22 = 0, r23 = 0, r24 = 0)
Arguments

- **score**: list; an object with class 'score' returned from inspect.score() function.
- **order**: integer; order of functional form for the score variable, 0 for corresponding random assignment designs, 1 for RD design with linear score variable, 2 for RD design with linear + quadratic score variable.
- **rhots**: correlation between the treatment and the scoring variable. Specify rhots = 0 or order = 0 to obtain results equivalent to random assignment designs.
- **k1**: numeric; left truncation point for truncated normal dist., or lower bound for uniform dist., ignored when rhots = 0 or order = 0.
- **k2**: numeric; right truncation point for truncated normal dist., or upper bound for uniform dist., ignored when rhots = 0 or order = 0.
- **dists**: character; distribution of the score variable, "normal" or "uniform". By default, dists = "normal" specification implies a truncated normal distribution with k1 = -6 and k2 = 6.
- **power**: statistical power (1 - β).
- **es**: effect size (Cohen’s d).
- **alpha**: probability of type I error (α).
- **two.tailed**: logical; TRUE for two-tailed hypothesis testing.
- **df**: degrees of freedom.
- **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).
- **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.
- **rate.tp**: treatment group participation rate.
- **rate.cc**: control group crossover rate.
- **p**: proportion of level 4 units in treatment condition.
- **n1**: average number of level 1 units per level 2 unit.
- **n2**: average number of level 2 units per level 3 unit.
- **n3**: average number of level 3 units per level 4 unit.
- **n4**: number of level 4 units.
- **cn1**: marginal cost per level 1 unit in treatment and control conditions, e.g. c(10, 5).
- **cn2**: marginal cost per level 2 unit in treatment and control conditions, e.g. c(50, 30).
- **cn3**: marginal cost per level 3 unit in treatment and control conditions, e.g. c(80, 50).
- **cn4**: marginal cost per level 4 unit in treatment and control conditions, e.g. c(100, 40).
- **cost**: total cost or budget.
**p0**
starting value for p when rhots = 0 and p = NULL. Starting value is replaced with average when p is constrained by bounds.

**n0**
vector of starting values for n1, n2, n3, n4 (positional). Starting values are replaced with averages when sample sizes are constrained by bounds.

**constrain**
character; "cost", "power", or "mdes".

**round**
logical; TRUE for rounded COSA solution.

**max.power**
logical; TRUE for maximizing power instead of minimizing variance.

**local.solver**
subset of c("LBFGS", "SLSQP").

**Value**

**params**
list of parameters used in the function.

**df**
degrees of freedom.

**sse**
standardized standard error.

**cosa**
constrained optimal sample allocation.

**mdes**
minimum detectable effect size and (1 - α)% confidence limits.

**power**
statistical power (1 - β)

**Examples**

```r
score.obj <- inspect.score(rnorm(10000), cutoff = 0)
power.crd4(score.obj, order = 2,
ess = .25, rho2 = .20, rho3 = .10, rho4 = .05,
g4 = 0, r24 = 0, n1 = 20, n2 = 3, n3 = 20, n4 = 20)

# optimal combination of sample sizes for level 1, level 3 and level 4
# that produce power = .80 (given range restriction for level 1 sample size)
cosa.crd4(score.obj, order = 2,
   constrain = "power", power = .80,
ess = .25, rho2 = .20, rho3 = .10, rho4 = .05,
g4 = 0, r24 = 0,
n1 = c(20, 60), n2 = 2, n3 = NULL, n4 = NULL)
```

---

**inspect.score**

*Inspects Relations between Treatment, Score and Score^2 Triad*

**Description**

Inspects relations between Treatment, Score and Score^2 triad, outputs correlations and design effects for linear and linear + quadratic functional forms for the score variable.

**Usage**

```r
inspect.score(score = NULL, sim = FALSE, p = NULL, cutoff = NULL, treat.lower = FALSE, mu = 0, sigma = 1, k1 = -1e+10, k2 = 1e+10,
dists = "normal", ndraw = 1000, nsim = 1000)
```
inspect.score

Arguments

sim logical; if TRUE results are based on simulation.
score vector; score variable.
p proportion of units in treatment condition.
cutoff cutoff.
treat.lower logical; if TRUE subjects below cutoff are treated.
mu mean of truncated normal - applies when score = NULL and dists = "normal".
sigma standard deviation of truncated normal - applies when score = NULL and dists = "normal".
k1 left truncation point for empirical, truncated normal, or uniform distribution.
k2 right truncation point for empirical, truncated normal, or uniform distribution.
dists char; type of distribution, "normal" or "uniform".
ndraw number of draws - applies when sim = TRUE.
nsim number of simulations - applies when sim = TRUE.

Value

parms list; list of parameters used in the computation.
cutoff cutoff score (computed if p is provided).
treat.lower logical; not used very much but may be utilized in the future.
p proportion of subjects treated (computed if cutoff is provided).
rhots correlation between Treatment and Score.
rhots2 correlation between Treatment and Score^2.
rhoss2 correlation between Score and Score^2.
d1 design effect for linear functional form.
d2 design effect for linear + quadratic functional form.

Examples

inspect.score(score = rnorm(10000), p = .50)
# default based on ~ N(0,1)
inspect.score(p = .50)
inspect.score(sim = TRUE, p = .50)
**ird**

Simple Individual-level Regression Discontinuity (w/ or w/o Strata or Fixed Blocks)

**Description**

Use `mdes.ird()` to calculate minimum detectable effect size and `power.ird()` to calculate statistical power. If higher level strata or fixed blocks exist, use `mdes.bird2f1()` to calculate minimum detectable effect size, `power.bird2f1()` to calculate statistical power, and `cosa.bird2f1()` for constrained optimal sample allocation. Alternatively modify degrees of freedom in `<output>.ird()` functions as `n1 - 2*nb - g1 - order` where `n1` is total number of subjects across blocks, and `nb` is number of blocks. Keep in mind that `r21` now includes information about blocks, but this fact will not be reflected in `g1`. See examples below.

**Usage**

```r
mdes.ird(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal", power = .80, alpha = .05, two.tailed = TRUE, df = n1 - g1 - order - 2, r21 = 0, g1 = 0, rate.tp = 1, rate.cc = 0, p = .50, n1)
```

```r
power.ird(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal", es = .25, alpha = .05, two.tailed = TRUE, df = n1 - g1 - order - 2, r21 = 0, g1 = 0, rate.tp = 1, rate.cc = 0, p = .50, n1)
```

```r
mdes.bird2f1(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal", power = .80, alpha = .05, two.tailed = TRUE, df = n2 * (n1 - 2) - g1 - order, r21 = 0, g1 = 0, rate.tp = 1, rate.cc = 0, p = .50, n1, n2 = 1)
```

```r
power.bird2f1(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal", es = .25, alpha = .05, two.tailed = TRUE, df = n2 * (n1 - 2) - g1 - order, r21 = 0, g1 = 0, rate.tp = 1, rate.cc = 0, p = .50, n1, n2 = 1)
```

```r
cosa.bird2f1(score = NULL, order = 2, rhots = NULL, k1 = -6, k2 = 6, dists = "normal", cn1 = 0, cn2 = 0, cost = NULL, n1 = NULL, n2 = NULL, p = NULL, n0 = c(400 + g1, 5), p0 = .499, constrain = "power", round = TRUE, max.power = FALSE, local.solver = c("LBFGS", "SLSQP"), power = .80, es = .25, alpha = .05, two.tailed = TRUE, g1 = 0, r21 = 0)
```

**Arguments**

- `score`: list; an object with class `score` returned from `inspect.score()` function.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>order</strong></td>
<td>integer; order of functional form for the score variable, 0 for corresponding random assignment designs, 1 for RD design with linear score variable, 2 for RD design with linear + quadratic score variable</td>
</tr>
<tr>
<td><strong>rhots</strong></td>
<td>correlation between the treatment and the scoring variable. Specify rhots = 0 or order = 0 to obtain results equivalent to random assignment designs.</td>
</tr>
<tr>
<td><strong>k1</strong></td>
<td>numeric; left truncation point for truncated normal dist., or lower bound for uniform dist., ignored when rhots = 0 or order = 0.</td>
</tr>
<tr>
<td><strong>k2</strong></td>
<td>numeric; right truncation point for truncated normal dist., or upper bound for uniform dist., ignored when rhots = 0 or order = 0.</td>
</tr>
<tr>
<td><strong>dists</strong></td>
<td>character; distribution of the score variable, &quot;normal&quot; or &quot;uniform&quot;. By default, dists = &quot;normal&quot; specification implies a truncated normal distribution with k1 = -6 and k2 = 6.</td>
</tr>
<tr>
<td><strong>power</strong></td>
<td>statistical power (1 - β).</td>
</tr>
<tr>
<td><strong>es</strong></td>
<td>numeric &gt; 0; effect size (Cohen's d).</td>
</tr>
<tr>
<td><strong>alpha</strong></td>
<td>probability of type I error (α).</td>
</tr>
<tr>
<td><strong>two.tailed</strong></td>
<td>logical; TRUE for two-tailed hypothesis testing.</td>
</tr>
<tr>
<td><strong>df</strong></td>
<td>degrees of freedom.</td>
</tr>
<tr>
<td><strong>g1</strong></td>
<td>number of covariates.</td>
</tr>
<tr>
<td><strong>r21</strong></td>
<td>proportion of variance in the outcome explained by covariates.</td>
</tr>
<tr>
<td><strong>p</strong></td>
<td>proportion of units in treatment condition.</td>
</tr>
<tr>
<td><strong>rate.tp</strong></td>
<td>treatment group participation rate.</td>
</tr>
<tr>
<td><strong>rate.cc</strong></td>
<td>control group crossover rate.</td>
</tr>
<tr>
<td><strong>n1</strong></td>
<td>sample size (per stratum or block, if exists).</td>
</tr>
<tr>
<td><strong>n2</strong></td>
<td>number of stratum or fixed blocks.</td>
</tr>
<tr>
<td><strong>cn1</strong></td>
<td>marginal cost per unit in treatment and control conditions, e.g. c(10, 5).</td>
</tr>
<tr>
<td><strong>cn2</strong></td>
<td>marginal cost per stratum or fixed block.</td>
</tr>
<tr>
<td><strong>cost</strong></td>
<td>total cost or budget.</td>
</tr>
<tr>
<td><strong>constrain</strong></td>
<td>character; &quot;cost&quot;, &quot;power&quot;, or &quot;es&quot;.</td>
</tr>
<tr>
<td><strong>n0</strong></td>
<td>starting value for n1 or n1, n2. Starting value is replaced with average when sample size is constrained by bounds.</td>
</tr>
<tr>
<td><strong>p0</strong></td>
<td>starting value for p when rhots = 0 and p = NULL. Starting value is replaced with average when p is constrained by bounds.</td>
</tr>
<tr>
<td><strong>round</strong></td>
<td>logical; TRUE for rounded COSA solution.</td>
</tr>
<tr>
<td><strong>max.power</strong></td>
<td>logical; TRUE for maximizing power instead of minimizing variance, applies when constrain = &quot;cost&quot;</td>
</tr>
<tr>
<td><strong>local.solver</strong></td>
<td>subset of c(&quot;LBFGS&quot;, &quot;SLSQP&quot;)</td>
</tr>
</tbody>
</table>
Value

- **parms**: list of parameters used in the function.
- **df**: degrees of freedom.
- **sse**: standardized standard error.
- **cosa**: constrained optimal sample allocation.
- **mdes**: minimum detectable effect size and \((1 - \alpha)\%\) confidence limits.
- **power**: statistical power \((1 - \beta)\)

Examples

```r
score.obj <- inspect.score(rnorm(10000), cutoff = 0)
power.ird(score.obj, order = 2,
          es = 0.25, g1 = 0, r21 = 0, n = 400)
# with 5 blocks \(df = n1 - 2\times(n\text{ blocks}) - \text{order} - g1\)
# \(n1\): number of subjects across five blocks
power.ird(score.obj, order = 2, df = 400 - 2\times5 - 2 - 0,
          es = 0.25, g1 = 0, r21 = .30, n = 400)

# compare
# \(n1\): number of subjects per block, \(n2\): number of blocks
power.bird2f1(score.obj, order = 2,
               es = 0.25, g1 = 0, r21 = .30,
               n1 = 80, n2 = 5)

# optimal combination of sample sizes for subjects and blocks
# that produce power = .80 (given range restrictions)
cosa.bird2f1(score.obj, order = 2,
             constrain = "power", power = .80,
             es = 0.25, g1 = 0, r21 = .30,
             n1 = c(100, 200), n2 = c(5, 10))
```

moments

**Moments of Empirical, Truncated Normal, or Uniform Distributions**

Description

If data (vector) is provided use `emp.moment()` function, otherwise for truncated normal distribution use `tnorm.moment()`, and for uniform distribution use `unif.moment()`.

Usage

```r
tnorm.moment(mu = 0, sigma = 1, k1 = -10, k2 = 10, order = 1, central = FALSE)
unif.moment(k1 = 0, k2 = 1, order = 1, central = FALSE)
emp.moment(x, order = 1, central = FALSE, absolute = FALSE, na.rm = FALSE)
```
Arguments

mu  mean of truncated normal - applies to `tnorm.moment()`.
sigma  standard deviation of truncated normal - applies to `tnorm.moment()`.
k1  left truncation point for truncated normal distribution or lower bound for uniform distribution.
k2  right truncation point for truncated normal distribution or upper bound for uniform distribution.
order  + int; order of moment
x  a vector of values - applies to `emp.moment()`.
central  logical; if TRUE produces central moments.
absolute  logical; if TRUE produces absolute moments - applies to `emp.moment()`.
na.rm  logical; if TRUE removes missing values - applies to `emp.moment()`.

Examples

tnorm.moment(k1 = -20, k2 = 20, order = 4, central = FALSE)
emp.moment(rnorm(10000), order = 4, central = FALSE)
unif.moment(k1 = 0, k2 = 1, order = 4, central = FALSE)
emp.moment(runif(10000), order = 4, central = FALSE)

plot

Power and MDES Curves

Description

Plots statistical power or minimum detectable effect size curves with \((1-\alpha)\times100\ %\) confidence interval for the design of interest.

Usage

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

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

```r
plot(x, ypar = "mdes", xpar = NULL,
     xlim = NULL, ylim = NULL,
     xlab = NULL, ylab = NULL,
     main = NULL, sub = NULL,
     locate = FALSE, benchmark = NULL, ...)
```

### Arguments

- **x**: an object returned from functions in `cosa` package.
- **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.
- **ylab**: y axis label.
- **main**: title for the plot.
- **sub**: subtitle for the plot.
- **locate**: logical; TRUE locates parameter values for design x on the plot.
- **benchmark**: benchmark line.
- **...**: other graphical parameters to pass to `plot.new()`.

### Examples

```r
d1 <- mdes.bcrd3r2(rho2 = .10, rho3 = .20, omega3 = .30,
                   n1 = 20, n2 = 44, n3 = 50)
plot(d1, xpar = "n3", xlim = c(30, 100))
```

---

### Description

Vectorizes BCOSA solutions based on multiple sets of parameter values. This is particularly useful when multiple designs are to be considered.

### Usage

```r
vectorize.cosa(design, args.grid, args.names = NULL, ordered = TRUE, ncase = 10L)
```
Vectorize BCOSA

Arguments

- **design**: an object returned from one of the cosa.<design>() functions.
- **args.grid**: vector or matrix: arguments’ grid consisting of sets of parameter values. A vector of values (for a single parameter) or a matrix (for multiple parameters).
- **args.names**: character list; arguments’ names. Default option args.names = NULL uses column names from args.grid
- **ordered**: logical: whether results should be ordered (cases with worst power rate or highest total cost are on top).
- **ncase**: integer: number of cases to be subsetted, ignored if ordered = FALSE.

Examples

```r
design <- cosa.crd2r2(rhots = 0, round = FALSE, constrain = "es", power = .80, cn1 = c(20, 10), cn2 = c(200, 50), es = .25, rho2 = .10, g2 = 3, r22 = .30, n1 = NULL, n2 = NULL, p = NULL)

args.grid <- expand.grid(rho2 = seq(.15, .25, .05))

vectorize.cosa(design, args.grid = args.grid, ordered = FALSE)
```
Index

bcd3f2 (crd2), 20
bcd3r2, 3
bcd4f3 (crd3), 23
bcd4r2, 5
bcd4r3, 8
bird2, 11
bird2f1 (ird), 30
bird2r1 (bird2), 11
bird3, 13
bird3r1 (bird3), 13
bird4, 16
bird4r1 (bird4), 16

cosa-deprecated, 19
cosa-package, 2
cosa.bcd3f2 (crd2), 20
cosa.bcd3r2 (bcd3r2), 3
cosa.bcd4f3 (crd3), 23
cosa.bcd4r2 (bcd4r2), 5
cosa.bcd4r3 (bcd4r3), 8
cosa.bird2 (bird2), 11
cosa.bird2f1 (ird), 30
cosa.bird2r1 (bird2), 11
cosa.bird3 (bird3), 13
cosa.bird3r1 (bird3), 13
cosa.bird4 (bird4), 16
cosa.bird4r1 (bird4), 16
cosa.crd2 (crd2), 20
cosa.crd2r2 (crd2), 20
cosa.crd3 (crd3), 23
cosa.crd3r3 (crd3), 23
cosa.crd4 (crd4), 26
cosa.crd4r4 (crd4), 26
crd2, 20
crd2r2 (crd2), 20
crd3, 23
crd3r3 (crd3), 23
crd4, 26
crd4r4 (crd4), 26

emp.moment (moments), 32
inspect.score, 28
ird, 30
ird1r1 (ird), 30

mdes.bcrd3f2 (crd2), 20
mdes.bcrd3r2 (bcd3r2), 3
mdes.bcrd4f3 (crd3), 23
mdes.bcrd4r2 (bcd4r2), 5
mdes.bcrd4r3 (bcd4r3), 8
mdes.bird2 (bird2), 11
mdes.bird2f1 (ird), 30
mdes.bird2r1 (bird2), 11
mdes.bird3 (bird3), 13
mdes.bird3r1 (bird3), 13
mdes.bird4 (bird4), 16
mdes.bird4r1 (bird4), 16
mdes.crd2 (crd2), 20
mdes.crd2r2 (crd2), 20
mdes.crd3 (crd3), 23
mdes.crd3r3 (crd3), 23
mdes.crd4 (crd4), 26
mdes.crd4r4 (crd4), 26

moments, 32

plot, 33
power.bcrd3f2 (crd2), 20
power.bcrd3r2 (bcd3r2), 3
power.bcd4f3 (crd3), 23
power.bcrd4r2 (bcd4r2), 5
power.bcrd4r3 (bcd4r3), 8
power.bird2 (bird2), 11
power.bird2f1 (ird), 30
power.bird2r1 (bird2), 11
power.bird3 (bird3), 13
power.bird3r1 (bird3), 13
power.bird4 (bird4), 16
INDEX

power.bird4r1 (bird4), 16
power.crd2 (crd2), 20
power.crd2r2 (crd2), 20
power.crd3 (crd3), 23
power.crd3r3 (crd3), 23
power.crd4 (crd4), 26
power.crd4r4 (crd4), 26
power.ird (ird), 30
power.ird1r1 (ird), 30

tnorm.moment (moments), 32
unif.moment (moments), 32

Vectorize BCOSA, 34
vectorize.cosa (Vectorize BCOSA), 34