Package ‘PowerUpR’

October 18, 2018

Type Package

Title Power Analysis Tools for Multilevel Randomized Experiments

Version 1.0.1

Date 2018-10-18

Description Includes tools to calculate statistical power, minimum detectable effect size, and minimum required sample size for various multilevel randomized experiments with continuous outcomes. Some of the functions can assist with planning two- and three-level cluster-randomized trials (CRTs) sensitive to moderation effects, and with planning two-level CRTs sensitive to 2-2-1 and 2-1-1 mediation effects. See 'PowerUp!' series at <https://www.causalevaluation.org/>.

Suggests knitr, rmarkdown

VignetteBuilder knitr

License GPL (>= 3)

NeedsCompilation no

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

Maintainer Metin Bulus <bulusmetin@gmail.com>

Repository CRAN

Date/Publication 2018-10-18 13:30:03 UTC

R topics documented:

PowerUpR-package ............................................. 2
bcr3f2 ............................................................. 2
bcr3r2 ............................................................. 4
bcr4f3 ............................................................. 5
bcr4r2 ............................................................. 7
bcr4r3 ............................................................. 9
bira2e1 ........................................................... 10
bira2f1 ........................................................... 12
bira2r1 ........................................................... 13
Description

*PowerUp! series consist of three excel-based applications which allow researchers to conduct statistical power analysis for various multilevel randomized experiments to detect main treatment effects, for two- and three-level cluster-randomized trials (CRTs) to detect moderation effects and for two-level CRTs to detect 2-2-1 and 2-1-1 mediation effects. For more information please refer to [http://www.causalevaluation.org/](http://www.causalevaluation.org/).*

**Usage**

```
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, J, K=10, tol=.10,
             rho2, p=.50, g2=0, r21=0, r22=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).
- **p**: average proportion of level 2 units randomly assigned to treatment within level 3 units.
- **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.
- **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

```r
# cross-checks
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, J=44)
```
Three-Level Blocked Cluster-level Random Assignment Design, Treatment at Level 2

Description

Use `mdes.bcra3r2()` to calculate minimum detectable effect size, `power.bcra3r2()` to calculate statistical power, and `mrss.bcra3r2()` to calculate minimum required sample size.

Usage

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

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

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

See Also
cosa.bcrd3r2

Examples

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

Description

Use mdes.bcra4f3() to calculate minimum detectable effect size, power.bcra4f3() to calculate statistical power, and mrss.bcra4f3() to calculate minimum required sample size.

Usage

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, K, L0=10, tol=.10,
   rho2, rho3, p=.50, g3=0, r21=0, r22=0, r23=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).
p     average proportion of level 3 units randomly assigned to treatment within level 4 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 2 variance in the outcome explained by level 2 covariates.
r23   proportion of level 3 variance in the outcome 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     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
mrss.bcra4f3(rho3=.15, rho2=.15,
   n=10, J=4, K=4, L=15)
power.bcra4f3(es=0.339, rho3=.15, rho2=.15,
   n=10, J=4, K=4, L=15)
mrss.bcra4f3(es=0.339, rho3=.15, rho2=.15,
   n=10, J=4, K=4)
Description

Use mdes.bcra4r2() to calculate minimum detectable effect size, power.bcra4r2() to calculate statistical power, and mrss.bcra4r2() to calculate minimum required sample size.

Usage

mdes.bcra4r2(power=.80, alpha=.05, two.tailed=TRUE,
   rho2, rho3, rho4, omega3, omega4,
   p=.50, r2l=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, omega3, omega4,
   p=.50, r2l=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, omega3, omega4,
   p=.50, r2l=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).
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.
$r_{22}$ proportion of level 2 variance in the outcome explained by level 2 covariates.

$r_{2t3}$ proportion of treatment effect variance among level 3 units explained by level 3 covariates.

$r_{2t4}$ 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.  

$L_0$ 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.

**See Also**

cosa.bcrd4r2

**Examples**

```r
# 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)
```
Four-Level Blocked Cluster-level Random Assignment Design, Treatment at Level 3

Description

Use `mdes.bcra4r3()` to calculate minimum detectable effect size, `power.bcra4r3()` to calculate statistical power, and `mrss.bcra4r3()` to calculate minimum required sample size.

Usage

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

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

```r
mrss.bcra4r3(es=.25, power=.80, alpha=.05, two.tailed=TRUE, 
n, J, K, L=0, tol=.10, 
rho2, rho3, rho4, omega4, 
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).
- `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.
\[ r_{23} \] proportion of level 3 variance in the outcome explained by level 3 covariates.

\[ r_{2t4} \] 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.

\( L_0 \) 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.

**See Also**

cosa.bcr4r3

**Examples**

```r
# cross-checks
mdes.bcr4r3(rho4=.05, rho3=.15, rho2=.15,
omega4=.50, n=10, J=4, K=4, L=20)
power.bcr4r3(es = .316, rho4=.05, rho3=.15, rho2=.15,
omega4=.50, n=10, J=4, K=4, L=20)
mrss.bcr4r3(es = .316, rho4=.05, rho3=.15, rho2=.15,
omega4=.50, n=10, J=4, K=4)
```

---

**Description**

Use `mdes.bira2c1()` to calculate minimum detectable effect size, `power.bira2c1()` to calculate statistical power, and `mrss.bira2c1()` to calculate minimum required sample size.

*Two-Level Blocked (Constant Treatment Effect) Individual-level Random Assignment Design, Treatment at Level 1*
Usage

mdes.bira2c1(power=0.80, alpha=0.05, two.tailed=TRUE, 
   p=0.50, g1=0, r21=0, 
   n, J)

power.bira2c1(es=0.25, alpha=0.05, two.tailed=TRUE, 
   p=0.50, g1=0, r21=0, 
   n, J)

mrss.bira2c1(es=0.25, power=0.80, alpha=0.05, two.tailed=TRUE, 
   n, J0=10, tol=0.10, 
   p=0.50, g1=0, r21=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.
p       average proportion of level 1 units randomly assigned to treatment within level 2 units.
g1       number of covariates at level 1.
r21      proportion of level 1 variance in the outcome explained by level 1 covariates.
n       harmonic mean of level 1 units across level 2 units (or simple average).
J       level 2 sample size.
J0      starting value for J.
tol      tolerance to end iterative process for finding J.

Value

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

Examples

# cross-checks
mdes.bira2c1(n=15, J=20)
power.bira2c1(es=.325, n=15, J=20)
mrss.bira2c1(es=.325, n=15)
bira2f1

Two-Level Blocked (Fixed Treatment Effect) Individual-level Random Assignment Design, Treatment at Level 1

Description

Use `mdes.bira2f1()` to calculate minimum detectable effect size, `power.bira2f1()` to calculate statistical power, and `mrss.bira2f1()` to calculate minimum required sample size.

Usage

```r
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, 
n, J0=10, tol=.10, 
p=.50, g1=0, r21=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.
- `p`: average proportion of level 1 units randomly assigned to treatment within level 2 units.
- `g1`: number of covariates at level 1.
- `r21`: proportion of level 1 variance in the outcome explained by level 1 covariates.
- `n`: harmonic mean of level 1 units across level 2 units (or simple average).
- `J`: level 2 sample size.
- `J0`: starting value for J.
- `tol`: tolerance to end iterative process for finding J.

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.

Examples

```r
# cross-checks
mdes.bira2r1(n=15, J=20)
power.bira2r1(es=.325, n=15, J=20)
mrss.bira2r1(es=.325, n=15)
```

bira2r1  Two-Level Blocked Individual-level Random Assignment Design, Treatment at Level 1

Description

Use `mdes.bira2r1()` to calculate minimum detectable effect size, `power.bira2r1()` to calculate statistical power, and `mrss.bira2r1()` to calculate minimum required sample size.

Usage

```r
mdes.bira2r1(power=.80, alpha=.05, two.tailed=TRUE,
              rho2, omega2, p=.50, g2=0, r2t2=0,
              n, J)
power.bira2r1(es=.25, alpha=.05, two.tailed=TRUE,
              rho2, omega2, g2=0, p=.50, r2t2=0,
              n, J)
mrss.bira2r1(es=.25, power=.80, alpha=.05, two.tailed=TRUE,
             n, J0=10, tol=.10,
             rho2, omega2, g2=0, p=.50, r2t2=0)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>power</td>
<td>statistical power ((1 - \beta)).</td>
</tr>
<tr>
<td>es</td>
<td>effect size.</td>
</tr>
<tr>
<td>alpha</td>
<td>probability of type I error.</td>
</tr>
<tr>
<td>two.tailed</td>
<td>logical; TRUE for two-tailed hypothesis testing, FALSE for one-tailed hypothesis testing.</td>
</tr>
<tr>
<td>rho2</td>
<td>proportion of variance in the outcome between level 2 units (unconditional ICC2).</td>
</tr>
<tr>
<td>omega2</td>
<td>treatment effect heterogeneity as ratio of treatment effect variance among level 2 units to the residual variance at level 2.</td>
</tr>
<tr>
<td>p</td>
<td>average proportion of level 1 units randomly assigned to treatment within level 2 units.</td>
</tr>
</tbody>
</table>
Three-Level Blocked Individual-level Random Assignment Design, Treatment at Level 1

Use `mdes.bira3r1()` to calculate minimum detectable effect size, `power.bira3r1()` to calculate statistical power, and `mrss.bira3r1()` to calculate minimum required sample size.
Usage

mdes.bira3r1(power=.80, alpha=.05, two.tailed=TRUE,  
rho2, rho3, omega2, omega3,  
p=.50, r21=0, r2t2=0, r2t3=0, g3=0,  
n, J, K)

power.bira3r1(es=.25, alpha=.05, two.tailed=TRUE,  
rho2, rho3, omega2, omega3,  
p=.50, r21=0, r2t2=0, r2t3=0, g3=0,  
n, J, K)

mrss.bira3r1(es=.25, power=.80, alpha=.05, two.tailed=TRUE,  
n, J, K0=10, tol=.10,  
rho2, rho3, omega2, omega3,  
p=.50, r21=0, r2t2=0, r2t3=0, g3=0)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>power</td>
<td>statistical power ((1 - \beta)).</td>
</tr>
<tr>
<td>es</td>
<td>effect size.</td>
</tr>
<tr>
<td>alpha</td>
<td>probability of type I error.</td>
</tr>
<tr>
<td>two.tailed</td>
<td>logical; TRUE for two-tailed hypothesis testing, FALSE for one-tailed testing.</td>
</tr>
<tr>
<td>rho2</td>
<td>proportion of variance in the outcome between level 2 units (unconditional ICC2).</td>
</tr>
<tr>
<td>rho3</td>
<td>proportion of variance in the outcome between level 3 units (unconditional ICC3).</td>
</tr>
<tr>
<td>omega2</td>
<td>treatment effect heterogeneity as ratio of treatment effect variance among level 2 units to the residual variance at level 2.</td>
</tr>
<tr>
<td>omega3</td>
<td>treatment effect heterogeneity as ratio of treatment effect variance among level 3 units to the residual variance at level 3.</td>
</tr>
<tr>
<td>p</td>
<td>average proportion of level 1 units randomly assigned to treatment within level 2 units.</td>
</tr>
<tr>
<td>g3</td>
<td>number of covariates at level 3.</td>
</tr>
<tr>
<td>r21</td>
<td>proportion of level 1 variance in the outcome explained by level 1 covariates.</td>
</tr>
<tr>
<td>r2t2</td>
<td>proportion of treatment effect variance among level 2 units explained by level 2 covariates.</td>
</tr>
<tr>
<td>r2t3</td>
<td>proportion of treatment effect variance among level 3 units explained by level 3 covariates.</td>
</tr>
<tr>
<td>n</td>
<td>harmonic mean of level 1 units across level 2 units (or simple average).</td>
</tr>
<tr>
<td>J</td>
<td>harmonic mean of level 2 units across level 3 units (or simple average).</td>
</tr>
<tr>
<td>K</td>
<td>number of level 3 units.</td>
</tr>
<tr>
<td>K0</td>
<td>starting value for K.</td>
</tr>
<tr>
<td>tol</td>
<td>tolerance to end iterative process for finding K.</td>
</tr>
</tbody>
</table>
Value

fun  function name.
params 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.

See Also

cosa.bird3r1

Examples

# cross-checks
mdes.bira3r1(rho3=.20, rho2=.15,
omega3=.10, omega2=.10,
n=69, J=10, K=100)

power.bira3r1(es = .045, rho3=.20, rho2=.15,
omega3=.10, omega2=.10,
n=69, J=10, K=100)

mrss.bira3r1(es = .045, rho3=.20, rho2=.15,
omega3=.10, omega2=.10,
n=69, J=10)

---

bira4r1  \textit{Four-Level Blocked Individual-level Random Assignment Design, Treatment at Level 1}

Description

Use \texttt{mdes.bira4r1()} to calculate minimum detectable effect size, \texttt{power.bira4r1()} to calculate statistical power, and \texttt{mrss.bira4r1()} to calculate minimum required sample size.

Usage

mdes.bira4r1(power=.80, alpha=.05, two.tailed=TRUE,
rho2, rho3, rho4, omega2, omega3, omega4,
p=.50, r2l=0, r2t2=0, r2t3=0, r2t4=0, g4=0,
n, J, K, L)

power.bira4r1(es=.25, alpha=.05, two.tailed=TRUE,
rho2, rho3, rho4, omega2, omega3, omega4,
p=.50, r2l=0, r2t2=0, r2t3=0, r2t4=0, g4=0,
n, J, K, L)
mrss.bira4r1(es=.25, power=.80, alpha=.05, two.tailed=TRUE, 
  n, J, K, L0=10, to1=.10, 
  rho2, rho3, rho4, omega2, omega3, omega4, 
  p=.50, r2t1=0, r2t2=0, r2t3=0, r2t4=0, g4=0)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>power</td>
<td>statistical power $(1 - \beta)$.</td>
</tr>
<tr>
<td>es</td>
<td>effect size.</td>
</tr>
<tr>
<td>alpha</td>
<td>probability of type I error.</td>
</tr>
<tr>
<td>two.tailed</td>
<td>logical; TRUE for two-tailed hypothesis testing, FALSE for one-tailed hypothesis testing.</td>
</tr>
<tr>
<td>rho2</td>
<td>proportion of variance in the outcome between level 2 units (unconditional ICC2).</td>
</tr>
<tr>
<td>rho3</td>
<td>proportion of variance in the outcome between level 3 units (unconditional ICC3).</td>
</tr>
<tr>
<td>rho4</td>
<td>proportion of variance in the outcome between level 4 units (unconditional ICC4).</td>
</tr>
<tr>
<td>omega2</td>
<td>treatment effect heterogeneity as ratio of treatment effect variance among level 2 units to the residual variance at level 2.</td>
</tr>
<tr>
<td>omega3</td>
<td>treatment effect heterogeneity as ratio of treatment effect variance among level 3 units to the residual variance at level 3.</td>
</tr>
<tr>
<td>omega4</td>
<td>treatment effect heterogeneity as ratio of treatment effect variance among level 4 units to the residual variance at level 4.</td>
</tr>
<tr>
<td>p</td>
<td>average proportion of level 1 units randomly assigned to treatment within level 2 units.</td>
</tr>
<tr>
<td>g4</td>
<td>number of covariates at level 4.</td>
</tr>
<tr>
<td>r21</td>
<td>proportion of level 1 variance in the outcome explained by level 1 covariates.</td>
</tr>
<tr>
<td>r2t2</td>
<td>proportion of treatment effect variance among level 2 units explained by level 2 covariates.</td>
</tr>
<tr>
<td>r2t3</td>
<td>proportion of treatment effect variance among level 3 units explained by level 3 covariates.</td>
</tr>
<tr>
<td>r2t4</td>
<td>proportion of treatment effect variance among level 4 units explained by level 4 covariates.</td>
</tr>
<tr>
<td>n</td>
<td>harmonic mean of level 1 units across level 2 units (or simple average).</td>
</tr>
<tr>
<td>J</td>
<td>harmonic mean of level 2 units across level 3 units (or simple average).</td>
</tr>
<tr>
<td>K</td>
<td>harmonic mean of level 3 units across level 4 units (or simple average).</td>
</tr>
<tr>
<td>L</td>
<td>number of level 4 units.</td>
</tr>
<tr>
<td>L0</td>
<td>starting value for L.</td>
</tr>
<tr>
<td>to1</td>
<td>tolerance to end iterative process for finding L.</td>
</tr>
</tbody>
</table>
Value

fun function name.

params 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.

See Also

cosa.bird4r1

Examples

# cross-checks
mdes.bira4r1(rho4=.05, rho3=.15, rho2=.15,
  omega4=.50, omega3=.50, omega2=.50,
  n=10, J=4, K=4, L=27)

power.bira4r1(es = 0.142, rho4=.05, rho3=.15, rho2=.15,
  omega4=.50, omega3=.50, omega2=.50,
  n=10, J=4, K=4, L=27)

mrss.bira4r1(es = 0.142, rho4=.05, rho3=.15, rho2=.15,
  omega4=.50, omega3=.50, omega2=.50,
  n=10, J=4, K=4)

---

development

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.bira2r1(es=.15, rho2=.35, omega2=.10, n=83, J=10)
design2 <- power.to.mdes(design1)
mdes.to.pctl(design2)

cra2r2

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

Description

Use mdes.<design>() to calculate minimum detectable effect size for main effect, mdesd.<design>() to calculate minimum detectable effect size difference for moderation effect, power.<design>() to calculate statistical power, and mrss.<design>() to calculate minimum required sample size. Use <output>.cra2r2() for main effect, <output>.mod221() for moderator effect at level 1, <output>.mod222() for moderator effect at level 2, power.med211() for 2-1-1 mediation effect, and power.med221() for 2-2-1 mediation effect.

Usage

mdes.cra2r2(power=.80, alpha=.05, two.tailed=TRUE, 
rho2, p=.50, g2=0, r21=0, r22=0,
 n, J)

mdesd.mod221(power=.80, alpha=.05, two.tailed=TRUE, 
rho2, omegam2, g1=0, r21=0, r2m=0,
 p=.50, q=NULL, n, J)

mdesd.mod222(power=.80, alpha=.05, two.tailed=TRUE, 
rho2, g2=0, r21=0, r22=0,
 p=.50, q=NULL, n, J)

power.cra2r2(es=.25, alpha=.05, two.tailed=TRUE, 
rho2, g2=0, p=.50, r21=0, r22=0,
 n, J)

power.mod221(es=.25, alpha=.05, two.tailed=TRUE, 
rho2, omegam2, g1=0, r21=0, r2m=0,
 p=.50, q=NULL, n, J)

power.mod222(es=.25, alpha=.05, two.tailed=TRUE, 
rho2, g2=0, r21=0, 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.cra2r2(es=.25, power=.80, alpha=.05, two.tailed=TRUE, n, J0=10, tol=.10, 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, g1=0, 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, g2=0, r21=0, r22=0, p=.50, q=NULL)

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.

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

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.

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.
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 (or mediator) effect that is explained by level 2 covariates. For the mediation model, proportion of mediator variance at level 2 explained by level 2 covariates.

*n* harmonic mean of level 1 units across level 2 units (or simple average).

*J* level 2 sample size.

*J0* starting value for *J*.

*tol* tolerance to end iterative process for finding *J*.

*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.

**See Also**

For a more flexible sample size determination see *cosa.crd2r2*.

**Examples**

```r
# cross-checks for the main effect
mdes.cra2r2(rho2=.17, n=15, J=20)
power.cra2r2(es=.629, rho2=.17, n=15, J=20)
mrss.cra2r2(es=.629, rho2=.17, n=15)

# cross-checks for the randomly varying cont. L1 moderator effect
mdes.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
mdes.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
```


Three-level Cluster-randomized Trials to Detect Main and Moderation Effects

Description

Use `mdes. <design>()` to calculate minimum detectable effect size for main effect, `mdesd. <design>()` to calculate minimum detectable effect size difference for moderation effect, `power. <design>()` to calculate statistical power, and `mrss. <design>()` to calculate minimum required sample size. Use `<output>.cra3r3()` for main effect, `<output>.mod331()` for moderator effect at level 1, `<output>.mod332()` for moderator effect at level 2, and `<output>.mod333()` for moderator effect at level 3.

Usage

```r
mdes.cra3r3(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,
```
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).
omega_m2  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.
omega_m3  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.
g1  number of covariates at level 1.
g2  number of covariates at level 2.
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.
r23  proportion of level 3 variance in the outcome explained by level 3 covariates.
r2m2  proportion of variance in the moderator effect that is explained by level 2 covariates.
r2m3  proportion of variance in the moderator effect that is 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  level 3 sample size.
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.

See Also

For a more flexible sample size determination see `cosa.crd3r3`.
Examples

# cross-checks for the main effect
mdes.cra3r3(rho3=.06, rho2=.17, n=15, J=3, K=60)
power.cra3r3(es=.269, rho3=.06, rho2=.17, n=15, J=3, K=60)
mrss.cra3r3(es=.269, rho3=.06, rho2=.17, n=15, J=3)

# cross-checks for the randomly varying cont. L1 moderator effect
mdes.modSS1(power=.80, alpha=.05, two.tailed=TRUE,  
rho2=.17, rho3=.06, omegam2=.10, omegam3=.10,  
q=0, n=15, J=3, K=60)
power.modSS1(es=.01248, alpha=.05, two.tailed=TRUE,  
rho2=.17, rho3=.06, omegam2=.10, omegam3=.10,  
q=0, n=15, J=3, K=60)
mrss.modSS1(es=.01248, alpha=.05, two.tailed=TRUE,  
rho2=.17, rho3=.06, omegam2=.10, omegam3=.10,  
q=0, n=15, J=3)

# cross-checks for the non-randomly varying cont. L1 moderator effect
mdesd.modSS1(power=.80, alpha=.05, two.tailed=TRUE,  
rho2=.17, rho3=.06, omegam2=0, omegam3=0,  
q=0, n=15, J=3, K=60)
power.modSS1(es=.0946, alpha=.05, two.tailed=TRUE,  
rho2=.17, rho3=.06, omegam2=0, omegam3=0,  
q=0, n=15, J=3, K=60)
mrss.modSS1(es=.0946, alpha=.05, two.tailed=TRUE,  
rho2=.17, rho3=.06, omegam2=0, omegam3=0,  
q=0, n=15, J=3)

# cross-checks for the randomly varying bin. L1 moderator effect
mdesd.modSS1(power=.80, alpha=.05, two.tailed=TRUE,  
rho2=.17, rho3=.06, omegam2=.10, omegam3=.10,  
q=.5, n=15, J=3, K=60)
power.modSS1(es=.2082, alpha=.05, two.tailed=TRUE,  
rho2=.17, rho3=.06, omegam2=.10, omegam3=.10,  
q=.5, n=15, J=3, K=60)
mrss.modSS1(es=.2082, alpha=.05, two.tailed=TRUE,  
rho2=.17, rho3=.06, omegam2=.10, omegam3=.10,  
q=.5, n=15, J=3)

# cross-checks for the non-randomly varying bin. L1 moderator effect
mdesd.modSS1(power=.80, alpha=.05, two.tailed=TRUE,  
rho2=.17, rho3=.06, omegam2=0, omegam3=0,  
q=.5, n=15, J=3, K=60)
power.modSS1(es=.1893, alpha=.05, two.tailed=TRUE,  
rho2=.17, rho3=.06, omegam2=0, omegam3=0,  
q=.5, n=15, J=3, K=60)
mrss.modSS1(es=.1893, alpha=.05, two.tailed=TRUE,  
rho2=.17, rho3=.06, omegam2=0, omegam3=0,  
q=.5, n=15, J=3)
Four-Level Cluster-level Random Assignment Design, Treatment at Level 4

Description

use `mdes.cra4r4()` to calculate minimum detectable effect size, `power.cra4r4()` to calculate statistical power, and `mrss.cra4r4()` to calculate minimum required sample size.

Usage

```r
mdes.cra4r4(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)
```

```r
power.cra4r4(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)
```

```r
mrss.cra4r4(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).
harmonic mean of level 3 units across level 4 units (or simple average).

number of level 4 units.

starting value for L.

tolerance to end iterative process for finding L.

function name.

list of parameters used in power calculation.

degrees of freedom.

noncentrality parameter.

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

minimum detectable effect size.

number of level 4 units.

See Also

cosa.crd4r4

Examples

```r
# cross-checks
mdes.ira1r1(power=.80, alpha=.05, two.tailed=TRUE, 
p=.50, g1=0, r21=0, n)

power.ira1r1(es=.25, alpha=.05, two.tailed=TRUE, 
p=.50, g1=0, r21=0, n)

mrss.ira1r1(es=.25, power=.80, alpha=.05, two.tailed=TRUE, 
n0=10, tol=.10, 
p=.50, g1=0, r21=0)
```

Description

Use `mdes.ira1r1()` to calculate minimum detectable effect size, `power.ira1r1()` to calculate statistical power, and `mrss.ira1r1()` to calculate minimum required sample size.

Usage

```r
mdes.ira1r1(power=.80, alpha=.05, two.tailed=TRUE, 
p=.50, g1=0, r21=0, n)

power.ira1r1(es=.25, alpha=.05, two.tailed=TRUE, 
p=.50, g1=0, r21=0, n)

mrss.ira1r1(es=.25, power=.80, alpha=.05, two.tailed=TRUE, 
n0=10, tol=.10, 
p=.50, g1=0, r21=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.
- **p**: proportion of units randomly assigned to treatment.
- **g1**: number of covariates.
- **r2**: proportion of variance in the outcome explained by covariates.
- **n**: sample size.
- **n0**: starting value for \(n\).
- **tol**: tolerance to end iterative process for finding \(n\).

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.

See Also

- `power.ird1r1`

Examples

```r
# cross-checks
mdes.ira1r1(n=250)
power.ira1r1(es=.356, n=250)
mrss.ira1r1(es=.356)
```
## Plots

Plots statistical power or minimum detectable effect size curves with \( (1-\alpha) \times 100 \% \) confidence interval.

### 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, NNN)

## 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, NNN)

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

### 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.
- **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.
- **...**: other graphical parameters to pass to `plot.new()`.
Examples

design1 <- mdesNcraSrS(rhoS=NP6L, rhoR=N17L, n=1UL, j=SL, k=6P)
plot(design1, ypar = "mdes", xpar = "K", xlim = c(30, 100))
plot(design1, ypar = "power", xpar = "K", xlim = c(30, 100))

design2 <- powerNcraSrS(es=NR69L, rhoS=NP6L, rhoR=N17L, n=1UL, J=3, K=60)
plot(design2, ypar = "mdes", xpar = "K", xlim = c(30, 100))
plot(design2, ypar = "power", xpar = "K", xlim = c(30, 100))

Description

Constrained optimal sample allocation functions are removed from **PowerUpR** package, see **cosa** package.

Format

Deprecated or defunct functions are no longer documented.

Details

Deprecated and defunct functions and their replacement are as follows:

- `optimal.ira1r1` is defunct, see **cosa** package
- `optimal.cra2r2` is defunct, see **cosa** package
- `optimal.cra3r3` is defunct, see **cosa** package
- `optimal.cra4r4` is defunct, see **cosa** package
- `optimal.bcra3f2` is defunct, see **cosa** package
- `optimal.bcra3r2` is defunct, see **cosa** package
- `optimal.bcra4r2` is defunct, see **cosa** package
- `optimal.bira4r1` is defunct, see **cosa** package
- `optimal.bira3r1` is defunct, see **cosa** package
- `optimal.bira2c1` is defunct, see **cosa** package
- `optimal.bira2f1` is defunct, see **cosa** package
- `optimal.bira2r1` is defunct, see **cosa** package
- `optimal.to.mdes` is defunct, there is no replacement function
- `optimal.to.power` is defunct, there is no replacement function
- `mdes.med211` is defunct, there is no replacement function
- `mdes.med221` is defunct, there is no replacement function
- `opt.mdes.med` is defunct, there is no replacement function
Plots Type I and Type II Error Rates

Description

t1t2.error plots Type I ($\alpha$) and Type II ($\beta$) 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.bira2r1(rho2=.35, omega2=.10,  
n=83, J=480)
t1t2.error(design1)

## End(Not run)
Index

bcra3f2, 2
bcra3r2, 4
bcra4f3, 5
bcra4r2, 7
bcra4r3, 9
bira2c1, 10
bira2f1, 12
bira2r1, 13
bira3r1, 14
bira4r1, 16
conversion, 18
cosa.bcrd3r2, 5
cosa.bcrd4r2, 8
cosa.bcrd4r3, 10
cosa.bird2r1, 14
cosa.bird3r1, 16
cosa.bird4r1, 18
cosa.crd2r2, 21
cosa.crd3r2, 24
cosa.crd4r4, 27
cra2r2, 19
bira3r3, 22
cra4r4, 26
ira1r1, 27
mdes.bcra3f2(bcra3f2), 2
mdes.bcra3r2(bcra3r2), 4
mdes.bcra4f3(bcra4f3), 5
mdes.bcra4r2(bcra4r2), 7
mdes.bcra4r3(bcra4r3), 9
mdes.bira2c1(bira2c1), 10
mdes.bira2f1(bira2f1), 12
mdes.bira2r1(bira2r1), 13
mdes.bira3r1(bira3r1), 14
mdes.bira4r1(bira4r1), 16
mdes.cra2r2(cra2r2), 19
mdes.cra3r3(cra3r3), 22
mdes.cra4r4(cra4r4), 26
mdes.ira1r1(ira1r1), 27
mdes.med211(PowerUpR-deprecated), 30
mdes.med221(PowerUpR-deprecated), 30
mdes.mod221(cra2r2), 19
mdes.mod222(cra2r2), 19
mdes.mod331(cra3r3), 22
mdes.mod332(cra3r3), 22
mdes.mod333(cra3r3), 22
mdes.to.pct1(conversion), 18
mdes.to.power(conversion), 18
mdesd.mod221(cra2r2), 19
mdesd.mod222(cra2r2), 19
mdesd.mod331(cra3r3), 22
mdesd.mod332(cra3r3), 22
mdesd.mod333(cra3r3), 22
mrss.bcra3f2(bcra3f2), 2
mrss.bcra3r2(bcra3r2), 4
mrss.bcra4f3(bcra4f3), 5
mrss.bcra4r2(bcra4r2), 7
mrss.bcra4r3(bcra4r3), 9
mrss.bira2c1(bira2c1), 10
mrss.bira2f1(bira2f1), 12
mrss.bira2r1(bira2r1), 13
mrss.bira3r1(bira3r1), 14
mrss.bira4r1(bira4r1), 16
mrss.cra2r2(cra2r2), 19
mrss.cra3r3(cra3r3), 22
mrss.ira1r1(ira1r1), 27
mrss.med211(PowerUpR-deprecated), 30
mrss.med221(PowerUpR-deprecated), 30
mrss.mod221(cra2r2), 19
mrss.mod222(cra2r2), 19
mrss.mod331(cra3r3), 22
mrss.mod332(cra3r3), 22
mrss.mod333(cra3r3), 22
mrss.to.mdes(conversion), 18
mrss.to.power(conversion), 18
opt.mdes.med(PowerUpR-deprecated), 30
optimal.bcra3f2(PowerUpR-deprecated), 30

INDEX

optimal.bcra3r2 (PowerUpR-deprecated), 30
optimal.bcra4f3 (PowerUpR-deprecated), 30
optimal.bcra4r2 (PowerUpR-deprecated), 30
optimal.bcra4r3 (PowerUpR-deprecated), 30
optimal.bira2c1 (PowerUpR-deprecated), 30
optimal.bira2f1 (PowerUpR-deprecated), 30
optimal.bira3r1 (PowerUpR-deprecated), 30
optimal.bira4r1 (PowerUpR-deprecated), 30
optimal.cra2r2 (PowerUpR-deprecated), 30
optimal.cra3r3 (PowerUpR-deprecated), 30
optimal.cra4r4 (PowerUpR-deprecated), 30
optimal.ira1r1 (PowerUpR-deprecated), 30
optimal.to.mdes (PowerUpR-deprecated), 30
optimal.to.power (PowerUpR-deprecated), 30

plot.mdes (plots), 29
plot.mrss (plots), 29
plot.power (plots), 29
plots, 29
power.bcra3f2 (bcra3f2), 2
power.bcra3r2 (bcra3r2), 4
power.bcra4f3 (bcra4f3), 5
power.bcra4r2 (bcra4r2), 7
power.bcra4r3 (bcra4r3), 9
power.bira2c1 (bira2c1), 10
power.bira2f1 (bira2f1), 12
power.bira2r1 (bira2r1), 13
power.bira3r1 (bira3r1), 14
power.bira4r1 (bira4r1), 16
power.cra2r2 (cra2r2), 19
power.cra3r3 (cra3r3), 22
power.cra4r4 (cra4r4), 26
power.ira1r1 (ira1r1), 27
power.ird1r1, 28
power.med211 (cra2r2), 19
power.med221 (cra2r2), 19
power.mod221 (cra2r2), 19

t1t2.error, 31

power.mod222 (cra2r2), 19
power.mod331 (cra3r3), 22
power.mod332 (cra3r3), 22
power.mod333 (cra3r3), 22
power.to.mdes (conversion), 18
PowerUpR (PowerUpR-package), 2
PowerUpR-deprecated, 30
PowerUpR-package, 2