Package ‘OPDOE’
March 17, 2018

Version 1.0-10
Title Optimal Design of Experiments
Author Petr Simecek <simecek@gmail.com>,
        Juergen Pilz <juergen.pilz@aau.at>,
        Mingui Wang <mail.mwang@gmail.com>,
        Albrecht Gebhardt <albrecht.gebhardt@aau.at>.
Maintainer Albrecht Gebhardt <albrecht.gebhardt@aau.at>
Description Several function related to Experimental Design are implemented here, see
Imports mvtnorm, orthopolynom, nlme, crossdes, polynom
Depends gmp
License GPL (>= 2)
NeedsCompilation no
Repository CRAN
Date/Publication 2018-03-17 22:49:18 UTC

R topics documented:
cattle ......................................................... 2
design.reg.polynom ........................................... 3
design.regression ............................................. 4
had ............................................................ 4
heights ......................................................... 5
hemp .......................................................... 6
OPDOE-undocumented ........................................ 6
print.design.regression ...................................... 7
print.triangular.test ........................................ 7
size.anova ................................................... 8
size_a.three_way .......................................... 10
size_b.three_way .......................................... 11
size_b.two_way ............................................. 13
size_bc.three_way .......................................... 14
Description

milk fat performance (in kg per lactation) of heifers of three sires from Holstein Frisian cattle to select the sire with the highest breeding value for milk fat performance.

Usage

data(cattle)

Format

The format is: num [1:5, 1:3] 132 128 135 121 138 166 172 176 169 ...

Author(s)

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt

References


Examples

data(cattle)
size.seq_select.mean(data=cattle, delta=10, P=0.95)
Description

Determines locations and number of replications for a polynomial regression design.

Needs specification of order of polynom, borders of intervall and total number of measurements as input.

Usage

design.regression.polynom(a, b, k, n)

design.reg.polynom(...)  

Arguments

a  lower bound of interval
b  upper bound of interval
k  order of polynom
n  total number of planned measurements
... only used for call wrapper design.reg.polynom

Details

Uses Legendre Polynomials to determine the support points for the design:

If $a = -1$, $b = 1$: places $k + 1$ support points in $[-1,1]$, located at the roots of $(1 - x^2) \frac{dP_k(x)}{dx}$
where $P_k(x)$ is the Legendre polynomial of degree $k$.

Distributes the $n$ measurements almost equally over the support points.

Value

Object of class design.regression

Note

design.reg.polynom is a call wrapper for backward compatibility for design.regression.polynom

Author(s)

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt

References

Examples

```r
x <- design.regression.polynom(10, 100, 3, 45)
x
```

---

**design.regression**  
*Regression Design Object*

---

**Description**

An `design.regression` object is created with `design.regression.polynom`.

**Arguments**

- `triangular.test` object is a list of character, currently only "polynomial" is implemented
- `locations` chosen locations
- `replications` chosen replications per location
- `interval` vector of size 2 storing the given interval

**Author(s)**

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt

**References**


**See Also**

`design.regression.polynom`

---

**had**  
*Stored Hadmard matrices*

---

**Description**

Some stored Hadmard matrices, used in `hadamard.matrix`.

**Details**

Stored matrices from http://www2.research.att.com/~njas/hadamard/ filling the gaps up to 256 in `hadamard.matrix`, 260 is the next gap.
heights

male / female heights data

Description

Body heights of male and female students collected in a classroom experiment.

Usage

data(heights)

Format

A data frame with 7 observations on the following 2 variables.

female  a numeric vector
male    a numeric vector

Author(s)

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt, Minghui Wang

References


Examples

data(heights)
attach(heights)
tt <- triangular.test.norm(x=female[1:3],
y=male[1:3], mu1=170, mu2=176, mu0=164,
alpha=0.05, beta=0.2, sigma=7)
# Test is yet unfinished, add the remaining values:
tt <- update(tt,x=female[4:7], y=male[4:7])
# Test is finished now
### Hemp data

**Description**

age and height of hemp plants.

**Usage**

```r
data(hemp)
```

**Format**

A data frame with 14 observations on the following 2 variables.

- `x` a numeric vector
- `y` a numeric vector

**References**


### (still) undocumented functions

**Description**

Undocumented / internal functions

**Details**

Some of these functions are not intended to be called by the user, others still lack their own documentation page. In the mean time see the referenced book.

**References**

Description
Prints a regression design object.

Usage
## S3 method for class 'design.regression'
print(x, epl = 6, ...)

Arguments
- `x` design.regression object
- `epl` integer, entries per line
- `...` additional print arguments

Author(s)
Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt

References

See Also
design.regression

---

Description
Prints a triangular.test object.

Usage
## S3 method for class 'triangular.test'
print(x, ...)

---
Arguments

x 
triangular.test object

... 
additional parameters for print

Author(s)

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt

References


See Also

triangular.test.norm, triangular.test.prop

size.anova Design of Experiments for ANOVA

Description

This function provides access to several functions returning the optimal number of levels and / or observations in different types of One-Way, Two-Way and Three-Way ANOVA.

Usage

size.anova(model, hypothesis = "", assumption = "",
          a = NULL, b = NULL, c = NULL, n = NULL, alpha, beta, delta, cases)

Arguments

model A character string describing the model, allowed characters are (>x) and the letters abeABC, capital letters stand for random factors, lower case letters for fixed factors, x means cross classification, > nested classification, brackets () are used to specify mixed model, the term in brackets has to come first. Spaces are allowed.

Examples: One-Way fixed: a, Two-Way: axB, a>b, Three-Way: axbxc, axbxC, a>b>c, (axb)>C, ...

hypothesis Character string describing Null hypothesis, can be omitted in most cases if it is clear that a test for no effects of factor A is performed, "a".

Other possibilities: "axb", "a>b", "c" and some more.

assumption Character string. A few functions need an assumption on sigma, like "sigma_AB=0,b=c", see the referenced book until this page is updated.

a Number of levels of fixed factor A
b  Number of levels of fixed factor B

Number of levels of fixed factor C

n  Number of Observations

alpha  Risk of 1st kind

beta  Risk of 2nd kind

delta  The minimum difference to be detected

cases  Specifies whether the "maximin" or "maximin" sizes are to be determined.

Details

see chapter 3 in the referenced book

Value

named integer giving the desired size(s)

Note

Depending on the selected model and hypothesis omit one or two of the sizes a, b, c, n. The function then tries to get its optimal value.

Author(s)

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt, Minghui Wang

References


Examples

size.anova(model="a", a=4,
           alpha=0.05, beta=0.1, delta=2, case="maximin")
size.anova(model="a", a=4,
           alpha=0.05, beta=0.1, delta=2, case="minimin")

size.anova(model="axb", hypothesis="a", a=6, b=4,
           alpha=0.05, beta=0.1, delta=1, cases="maximin")
size.anova(model="axb", hypothesis="a", a=6, b=4,
           alpha=0.05, beta=0.1, delta=1, cases="minimin")

size.anova(model="axb", hypothesis="axb", a=6, b=4,
           alpha=0.05, beta=0.1, delta=1, cases="maximin")
size.anova(model="axb", hypothesis="axb", a=6, b=4,
           alpha=0.05, beta=0.1, delta=1, cases="minimin")

size.anova(model="axBxC", hypothesis="a",
           assumption="sigma_AC=0, b=c", a=6, n=2,
Three-way analysis of variance – mixed classification (A × B) × C

Description

Returns the optimal number of levels for factor A (and B).

Usage

```r
size_a.three_way(size_aNthree_way_mixed_cxbinaNmodel_S_cHalphaL betaL deltaL bL cL nL casesI)
size_a.three_way(size_aNthree_way_mixed_cxbinaNmodel_7_cHalphaL betaL deltaL bL cL nL casesI)
size_ab.three_way(size_abNthree_way_mixed_cxbinaNmodel_7_cHalphaL betaL deltaL cL nL casesI)
```

Arguments

- `alpha`: Risk of 1st kind
- `beta`: Risk of 2nd kind
- `delta`: The minimum difference to be detected
- `b`: Number of levels of fixed factor B
- `c`: Number of levels of fixed factor C
- `n`: Number of replications
- `cases`: Specifies whether the "maximin" or "maximin" sizes are to be determined
size_b.three_way

Details

see chapter 3 in the referenced book

Value

Integer(s) giving the size(s).

Note

Better use size.anova which allows a cleaner notation.

Author(s)

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt, Minghui Wang

References


See Also

size.anova

Examples

size_a.three_way_mixed_cxbina.model_3_c(0.05, 0.1, 0.5, 5, 4, 1, "maximin")
size_a.three_way_mixed_cxbina.model_3_c(0.05, 0.1, 0.5, 5, 4, 1, "minimin")
size_a.three_way_mixed_cxbina.model_7_c(0.05, 0.1, 0.5, 5, 4, 1, "maximin")
size_a.three_way_mixed_cxbina.model_7_c(0.05, 0.1, 0.5, 4, 1, "minimin")
size_ab.three_way_mixed_cxbina.model_7_c(0.05, 0.1, 0.5, 5, 2, "maximin")
size_ab.three_way_mixed_cxbina.model_7_c(0.05, 0.1, 0.5, 5, 2, "minimin")

size_b.three_way

Three-way analysis of variance – nested and mixed classification \( A \succ B \succ C \) and \( (A \times B) \succ C \) model III, IV and VII

Description

Returns the optimal number of levels for factor B.

Usage

size_b.three_way_mixed_ab_in_c.model_3_a(alpha, beta, delta, a, c, n, cases)
Arguments

- `alpha`: Risk of 1st kind
- `beta`: Risk of 2nd kind
- `delta`: The minimum difference to be detected
- `a`: Number of levels of fixed factor A
- `c`: Number of levels of fixed factor C
- `n`: Number of replications
- `cases`: Specifies whether the "maximin" or "minimin" sizes are to be determined

Details

see chapter 3 in the referenced book

Value

Integer giving the size.

Note

Better use `size.anova` which allows a cleaner notation.

Author(s)

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt, Minghui Wang

References


See Also

`size.anova`

Examples

```r
size_b.three_way_mixed_ab_in_c.model_3_a(0.05, 0.1, 0.5, 6, 5, 1, "maximin")
size_b.three_way_mixed_ab_in_c.model_3_a(0.05, 0.1, 0.5, 6, 5, 1, "minimin")
size_b.three_way_mixed_cxbina.model_4_a(0.05, 0.1, 0.5, 6, 4, 1, "maximin")
size_b.three_way_mixed_cxbina.model_4_a(0.05, 0.1, 0.5, 6, 4, 1, "minimin")
size_b.three_way_mixed_cxbina.model_4_c(0.05, 0.1, 0.5, 6, 4, 1, "maximin")
size_b.three_way_mixed_cxbina.model_4_c(0.05, 0.1, 0.5, 6, 4, 1, "minimin")
size_b.three_way_mixed_cxbina.model_4_axc(0.05, 0.1, 0.5, 6, 4, 1, "maximin")
size_b.three_way_mixed_cxbina.model_4_axc(0.05, 0.1, 0.5, 6, 4, 1, "minimin")
size_b.three_way_nested.model_6_a(0.05, 0.1, 0.5, 6, 4, 2, "maximin")
size_b.three_way_nested.model_6_a(0.05, 0.1, 0.5, 6, 4, 2, "minimin")
```
Description

Returns the optimal number of observations per level of factor B.

Usage

size_b.two_way_cross.mixed_model.a_fixed.a(alpha, beta, delta, a, n, cases)
size_b.two.way_nested.b_random.a_fixed.a(alpha, beta, delta, a, cases)

Arguments

alpha Risk of 1st kind
beta Risk of 2nd kind
delta The minimum difference to be detected
a Number of levels of fixed factor A
n Number of replications
cases Specifies whether the "maximin" or "minimax" sizes are to be determined

Details

see chapter 3 in the referenced book

Value

Integer giving the size.

Note

Better use size.anova which allows a cleaner notation.

Author(s)

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt, Minghui Wang

References


See Also

size.anova
Examples

size_b.two_way_cross.mixed_model_a_fixed_a(0.05, 0.1, 1, 6, 1, "maximin")
size_b.two_way_cross.mixed_model_a_fixed_a(0.05, 0.1, 1, 6, 1, "minimin")
size_b.two_way_cross.mixed_model_a_fixed_a(0.05, 0.1, 1, 6, 2, "maximin")
size_b.two_way_cross.mixed_model_a_fixed_a(0.05, 0.1, 1, 6, 2, "minimin")
size_b.two_way_nested.b_random_a_fixed_a(0.05, 0.1, 1, 6, "maximin")
size_b.two_way_nested.b_random_a_fixed_a(0.05, 0.1, 1, 6, "minimin")

Description

Returns the optimal number of levels for factor B and C.

Usage

size_bc.three_way.cross.model_T_a_case1(alpha, beta, delta, a, n, cases)
size_bc.three_way.cross.model_T_a_case2(alpha, beta, delta, a, n, cases)
size_bc.three_way.mixed_cxbina.model_6_a_case1(alpha, beta, delta, a, n, cases)
size_bc.three_way.mixed_cxbina.model_6_a_case2(alpha, beta, delta, a, n, cases)

Arguments

alpha Risk of 1st kind
beta Risk of 2nd kind
delta The minimum difference to be detected
a Number of levels of fixed factor A
n Number of replications
cases Specifies whether the "maximin" or "minimin" sizes are to be determined

Details

see chapter 3 in the referenced book

Value

Integers giving the sizes.

Note

Better use size.anova which allows a cleaner notation.
size_c.three_way

Author(s)
Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt, Minghui Wang

References

See Also
size.anova

Examples
size.bc.three_way_cross.model_4_a_case1(0.05, 0.1, 0.5, 6, 2, "maximin")
size.bc.three_way_cross.model_4_a_case1(0.05, 0.1, 0.5, 6, 2, "minimin")
size.bc.three_way_cross.model_4_a_case1(0.05, 0.1, 1, 6, 2, "maximin")
size.bc.three_way_cross.model_4_a_case1(0.05, 0.1, 1, 6, 2, "minimin")
size.bc.three_way_cross.model_4_a_case2(0.05, 0.1, 0.5, 6, 2, "maximin")
size.bc.three_way_cross.model_4_a_case2(0.05, 0.1, 0.5, 6, 2, "minimin")
size.bc.three_way_cross.model_4_a_case2(0.05, 0.1, 1, 6, 2, "maximin")
size.bc.three_way_cross.model_4_a_case2(0.05, 0.1, 1, 6, 2, "minimin")
size.bc.three_way_cross.model_6_a_case1(0.05, 0.1, 0.5, 6, 2, "maximin")
size.bc.three_way_cross.model_6_a_case1(0.05, 0.1, 0.5, 6, 2, "minimin")
size.bc.three_way_cross.model_6_a_case2(0.05, 0.1, 0.5, 6, 2, "maximin")
size.bc.three_way_cross.model_6_a_case2(0.05, 0.1, 0.5, 6, 2, "minimin")
size.bc.three_way_mixed_cxbina.model_6_a_case1(0.05, 0.1, 0.5, 6, 2, "maximin")
size.bc.three_way_mixed_cxbina.model_6_a_case1(0.05, 0.1, 0.5, 6, 2, "minimin")
size.bc.three_way_mixed_cxbina.model_6_a_case2(0.05, 0.1, 0.5, 6, 2, "maximin")
size.bc.three_way_mixed_cxbina.model_6_a_case2(0.05, 0.1, 0.5, 6, 2, "minimin")
size.bc.three_way_nested.model_5_a (alpha, beta, delta, a, b, n, cases)
size.bc.three_way_nested.model_5_a (alpha, beta, delta, a, b, n, cases)
size.bc.three_way_nested.model_5_a (alpha, beta, delta, a, b, n, cases)
size.bc.three_way_nested.model_5_a (alpha, beta, delta, a, b, n, cases)
size.bc.three_way_nested.model_5_a (alpha, beta, delta, a, b, n, cases)

size_c.three_way

Three-way analysis of variance – several cross-, nested and mixed classifications.

Description
Returns the optimal number of levels for .

Usage
size_c.three_way_cross.model_3_a (alpha, beta, delta, a, b, n, cases)
size_c.three_way_cross.model_3_axb (alpha, beta, delta, a, b, n, cases)
size_c.three_way_mixed_ab_in_c.model_5_a (alpha, beta, delta, a, b, n, cases)
size_c.three_way_mixed_ab_in_c.model_5_axb (alpha, beta, delta, a, b, n, cases)
size_c.three_way_mixed_ab_in_c.model_5_b (alpha, beta, delta, a, b, n, cases)
size_c.three_way_mixed_ab_in_c.model_6_b (alpha, beta, delta, a, b, n, cases)
size_c.three_way_mixed_cxbina.model_5_a (alpha, beta, delta, a, b, n, cases)
size_c.three_way_mixed_cxbina.model_5_b (alpha, beta, delta, a, b, n, cases)
size_c.three_way_mixed_cxbina.model_7_b (alpha, beta, delta, a, b, n, cases)
size_c.three_way_nested.model_5_a (alpha, beta, delta, a, b, n, cases)
Arguments

- **alpha**: Risk of 1st kind
- **beta**: Risk of 2nd kind
- **delta**: The minimum difference to be detected
- **a**: Number of levels of fixed factor A
- **b**: Number of levels of fixed factor B
- **n**: Number of replications
- **cases**: Specifies whether the "maximin" or "minimin" sizes are to be determined

Details

see chapter 3 in the referenced book

Value

integer, desired size of factor C

Note

Better use `size.anova` which allows a cleaner notation.

Author(s)

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt, Minghui Wang

References


See Also

`size.anova`

Examples

```r
size_c.three_way_cross.model_3_a(0.05, 0.1, 0.5, 6, 5, 2, "maximin")
size_c.three_way_cross.model_3_a(0.05, 0.1, 0.5, 6, 5, 2, "minimin")
```

```r
size_c.three_way_cross.model_3_axb(0.05, 0.1, 0.5, 6, 5, 2, "maximin")
size_c.three_way_cross.model_3_axb(0.05, 0.1, 0.5, 6, 5, 2, "minimin")
```

```r
size_c.three_way_mixed_ab_in_c.model_5_a(0.05, 0.1, 0.5, 6, 5, 1, "maximin")
size_c.three_way_mixed_ab_in_c.model_5_a(0.05, 0.1, 0.5, 6, 5, 1, "minimin")
```
Design for One-Way ANOVA

Description
Returns the optimal number of observations per level of factor A.

Usage
size_n.one_way.model_1(alpha, beta, delta, a, cases)

Arguments
alpha  
Risk of 1st kind
beta  
Risk of 2nd kind
delta  
The minimum difference to be detected
a  
Number of levels of fixed factor A
cases  
Specifies whether the "maximin" or "minimin" sizes are to be determined
Details

see chapter 3 in the referenced book

Value

Integer giving the size.

Note

Better use \texttt{size.anova} which allows a cleaner notation.

Author(s)

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt, Minghui Wang

References


See Also

\texttt{size.anova}

Examples

\begin{verbatim}
size_n.one_way.model_1(0.05,0.1,2,4, "maximin")
size_n.one_way.model_1(0.05,0.1,2,4, "minimin")
\end{verbatim}
Arguments

- alpha: Risk of 1st kind
- beta: Risk of 2nd kind
- delta: The minimum difference to be detected
- a: Number of levels of fixed factor A
- b: Number of levels of fixed factor B
- c: Number of levels of fixed factor C
- cases: Specifies whether the "maximin" or "maximin" sizes are to be determined

Details

see chapter 3 in the referenced book

Value

Integer giving the size.

Note

Better use `size.anova` which allows a cleaner notation.

Author(s)

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt, Minghui Wang

References


See Also

`size.anova`
Examples

size_n.three_way.cross.model_1_a(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way.cross.model_1_a(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way.cross.model_1_axb(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way.cross.model_1_axbc(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way.cross.model_1_axbc(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way.mixed_ab_in_c.model_1_a(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way.mixed_ab_in_c.model_1_a(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way.mixed_ab_in_c.model_1_axb(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way.mixed_ab_in_c.model_1_axb(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way.mixed_ab_in_c.model_1_axbc(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way.mixed_ab_in_c.model_1_axbc(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way.mixed_ab_in_c.model_1_axbc(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way.mixed_ab_in_c.model_1_axbc(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way.mixed_ab_in_c.model_1_a(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way.mixed_ab_in_c.model_1_a(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way.mixed_ab_in_c.model_1_axb(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way.mixed_ab_in_c.model_1_axb(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way.mixed_ab_in_c.model_1_axbc(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way.mixed_ab_in_c.model_1_axbc(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way.mixed_ab_in_c.model_1_axbc(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way.mixed_ab_in_c.model_1_axbc(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way.mixed_cxbina.model_1_a(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way.mixed_cxbina.model_1_a(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way.mixed_cxbina.model_1_axc(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way.mixed_cxbina.model_1_axc(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way.mixed_cxbina.model_1_b(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way.mixed_cxbina.model_1_b(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way.mixed_cxbina.model_1_bxc(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way.mixed_cxbina.model_1_bxc(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way.mixed_cxbina.model_1_c(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way.mixed_cxbina.model_1_c(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way.mixed_cxbina.model_3_b(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way.mixed_cxbina.model_3_b(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way.mixed_cxbina.model_3_bxc(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way.mixed_cxbina.model_3_bxc(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way.nested.model_1_a(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way.nested.model_1_a(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way.nested.model_1_b(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way.nested.model_1_b(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way.nested.model_1_c(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way.nested.model_1_c(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way.nested.model_3_b(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way.nested.model_3_b(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way.nested.model_3_c(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way.nested.model_3_c(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way.nested.model_4_c(0.05, 0.1, 0.5, 6, NA, 4, "maximin")
size_n.three_way.nested.model_4_c(0.05, 0.1, 0.5, 6, NA, 4, "minimin")
size_n.three_way.nested.model_8_c(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way.nested.model_8_c(0.05, 0.1, 0.5, 6, 5, 4, "minimin")

size_n.two_way

Design for Two-Way ANOVA
size_n.two_way

Description
Returns the optimal number of observations per level of factor A.

Usage
size_n.two_way_cross.model_1_a(alpha, beta, delta, a, b, cases)
size_n.two_way_cross.model_1_axb(alpha, beta, delta, a, b, cases)
size_n.two_way_nested.model_1_test_factor_a(alpha, beta, delta, a, b, cases)
size_n.two_way_nested.model_1_test_factor_b(alpha, beta, delta, a, b, cases)
size_n.two_way_nested.a_random_b_fixed_b(alpha, beta, delta, a, b, cases)

Arguments
alpha Risk of 1st kind
beta Risk of 2nd kind
delta The minimum difference to be detected
a Number of levels of fixed factor A
b Number of levels of fixed factor B
cases Specifies whether the "maximin" or "maximin" sizes are to be determined

Details
see chapter 3 in the referenced book

Value
Integer giving the size.

Note
Better use size.anova which allows a cleaner notation.

Author(s)
Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt, Minghui Wang

References

See Also
size.anova
Examples

```r
size_n.two_way_cross.model_1_a(0.05, 0.1, 1, 6, 4, "maximin")
size_n.two_way_cross.model_1_a(0.05, 0.1, 1, 6, 4, "minimin")
size_n.two_way_cross.model_1_axb(0.05, 0.1, 1, 6, 4, "maximin")
size_n.two_way_cross.model_1_axb(0.05, 0.1, 1, 6, 4, "minimin")
size_n.two_way_nested.model_1_test_factor_a(0.05, 0.1, 1, 6, 4, "maximin")
size_n.two_way_nested.model_1_test_factor_a(0.05, 0.1, 1, 6, 4, "minimin")
size_n.two_way_nested.a_random_b_fixed_b(0.05, 0.1, 1, 2, 10, "maximin")
size_n.two_way_nested.a_random_b_fixed_b(0.05, 0.1, 1, 2, 10, "minimin")
size_n.two_way_nested.a_random_b_fixed_b(0.05, 0.1, 1, 3, 10, "maximin")
size_n.two_way_nested.a_random_b_fixed_b(0.05, 0.1, 1, 3, 10, "minimin")
size_n.two_way_nested.a_random_b_fixed_b(0.05, 0.1, 1, 10, 10, "maximin")
size_n.two_way_nested.a_random_b_fixed_b(0.05, 0.1, 1, 10, 10, "minimin")
```

### triangular.test  
**Triangular Test Object**

#### Description

An `triangular.test` object is created with `triangular.test.norm` or `triangular.test.prop`

#### Arguments

A `triangular.test` object is a list of

- `x`: data for group 1
- `y`: data for group 2
- `n`: size of group 1
- `m`: size of group 2
- `alpha`: risk of 1st kind
- `beta`: risk of 2nd kind
- `dist`: character, either "normal" or "bernoulli", describing the type of triangular test
- `sample`: character, "one" or "two"
- `kind`: character, "one-sided" or "two-sided"
- `p0`: parameter describing the Null hypothesis, see `triangular.test.prop`
- `p1`: ...
- `p2`: ...
- `mu0`: parameter describing the Null hypothesis, see `triangular.test.norm`
- `mu1`: ...
- `mu2`: ...
- `result`: character, outcome of the test, "H0" or "H1"
- `step`: total number of steps and some more components for internal use.
triangular.test.norm

Author(s)
Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt

References

See Also
triangular.test.norm, triangular.test.prop

---

triangular.test.norm  Triangular Test for Normal Data

Description
Performs a sequential test, compares means of two normally distributed groups.

Usage
triangular.test.norm(x, y = NULL, mu0 = NULL, mu1, mu2 = NULL, delta = NULL, sigma = NULL, sigma2 = NULL, alpha = 0.05, beta = 0.1, plot = TRUE)

Arguments

- **x**: initial data for group x, at least 1 entry.
- **y**: initial data for group y, at least 1 entry for a two sample test, otherwise omitted.
- **mu0**: specifies Null and alternative hypothesis, see Details below.
- **mu1**: specifies Null and alternative hypothesis, see Details below.
- **mu2**: specifies Null and alternative hypothesis, see Details below.
- **delta**: The minimum difference to be detected, alternative way to specify mu2=m1+delta, see above, use either this or mu2.
- **sigma**: prior sigma.
- **sigma2**: prior sigma for group 2 if different than for group 1.
- **alpha**: Risk of 1st kind
- **beta**: Risk of 2nd kind
- **plot**: logical, indicates whether a initial plot should be generated.
Details

One-sample:
This function performs a one- or two-sided sequential Test for \( \mu = \mu_1 \) versus
\( \mu > \mu_2 \), if \( \mu_2 > \mu_1 \) (one-sided)
\( \mu < \mu_2 \), if \( \mu_2 < \mu_1 \) (one-sided)
\( \mu < \mu_0 \) or \( \mu > \mu_2 \), if \( \mu_2 > \mu_1 \) and \( \mu_0 < \mu_1 \) (two-sided, possibly unsymmetric)

Two-sample:
This function performs a one- or two-sided sequential Test for equal means \( \mu_1 = \mu_1 \mu_2 = \mu_1 \) in both groups versus
\( \mu_2 > \mu_2 \), if \( \mu_2 > \mu_1 \) (one-sided)
\( \mu_2 < \mu_2 \), if \( \mu_2 < \mu_1 \) (one-sided)
\( \mu_2 < \mu_0 \) or \( \mu_2 > \mu_2 \), if \( \mu_2 > \mu_1 \) and \( \mu_0 < \mu_1 \) (two-sided, possibly unsymmetric)

Value

An object of class \texttt{triangular.test}, to be used for later update steps.

Note

A two-sided test may be specified by supplying both \( \mu_1 \) and \( \mu_2 \), even unsymmetric if needed.

Author(s)

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt

References


See Also

\texttt{triangular.test, triangular.test.prop, update.triangular.test}

Examples

data(heights)
attach(heights)
# a symmetric two sided alternative:
tt <- triangular.test.norm(x=female[1:3],
y=male[1:3], mu1=170,mu2=176,mu0=164,
alpha=0.05, beta=0.2,sigma=7)
# Test is yet unfinished, add the remaining values step by step:
tt <- update(tt,x=female[4])
tt <- update(tt,x=male[4])
tt <- update(tt,x=female[5])
tt <- update(tt,y=male[5])
triangular.test.prop

```r
# Test is finished now
# an unsymmetric two sided alternative:
tt2 <- triangular.test.norm(x=female[1:3],
y=male[1:3], mu1=170,mu2=180,mu0=162,
alpha=0.05, beta=0.2,sigma=7)
tt2 <- update(tt2,x=female[4])
```

---

**triangular.test.prop**  
**Triangular Test for Bernoulli Data**

**Description**

Performs a sequential test, compares probabilities in two groups.

**Usage**

```r
triangular.test.prop(x, y = NULL, p0 = NULL, p1 = NULL, p2 = NULL, alpha = 0.05, beta = 0.1, delta = NULL, plot = TRUE)
```

**Arguments**

- `x`  
  initial data for group x, at least 1 entry, values restricted to 0 and 1.
- `y`  
  initial data for group y, at least 1 entry for a two sample test, otherwise omitted, values restricted to 0 and 1.
- `p0`  
  specifies Null and alternative hypothesis, see Details below.
- `p1`  
  specifies Null and alternative hypothesis, see Details below.
- `p2`  
  specifies Null and alternative hypothesis, see Details below.
- `alpha`  
  Risk of 1st kind
- `beta`  
  Risk of 2nd kind
- `plot`  
  logical, indicates whether a initial plot should be generated.
- `delta`  
  The minimum difference to be detected, alternative way to specify p2=p1+delta, see above, use either this or p2.

**Details**

**One-sample:**

This function performs a one- or two-sided sequential Test for \( p = p1 \) versus \( p > p2, \text{if } p2 > p1 \) (one-sided)  
\( p < p2, \text{if } p2 < p1 \) (one-sided)  
\( p < p0 \text{ or } p > p2, \text{if } p2 > p1 \text{ and } p0 < p1 \) (two-sided, possibly unsymmetric)
Two-sample:
This function performs a one- or two-sided sequential Test for equal proportions \( p_1 = p_1 \) \( p_2 = p_2 \) versus

\[ p_2 > p_2, \text{ if } p_2 > p_1 \text{ (one-sided)} \]

\[ p_2 < p_2, \text{ if } p_2 < p_1 \text{ (one-sided)} \]

\[ p_2 < \phi \text{ or } p_2 > p_2, \text{ if } p_2 > p_1 \text{ and } \phi < p_1 \text{ (two-sided, possibly unsymmetric)} \]

Value
An object of class `triangular.test`, to be used for later update steps.

Note
A two-sided test may be specified by supplying both \( p_1 \) and \( p_2 \), even unsymmetric if needed.

Author(s)
Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt

References

See Also
`triangular.test`, `triangular.test.norm`, `update.triangular.test`

Examples
```r
data(heights)
attach(heights)
male180 <- as.integer(male>180)
female164 <- as.integer(female>164)
sum(male180)/length(male180)
tt <- triangular.test.prop(x=female164[1:3],
y=male180[1:3], p1=0.4,p2=0.8,\(\phi\)=0.1,
alpha=0.05, beta=0.2)
tt <- update(tt,x=female164[4])
tt <- update(tt,y=male180[4])
tt <- update(tt,x=female164[5])
sum(female164)/length(female164)
```
**Description**

Updates a triangular.test object and executes one or more steps in the sequence of tests.

**Usage**

```r
## S3 method for class 'triangular.test'
update(object, x=NULL, y=NULL, initial=FALSE, plot="last", recursive=FALSE, ...)
```

**Arguments**

- `object`  
  *triangular.test* object
- `x`  
  data for group 1
- `y`  
  data for group 2
- `initial`  
  logical, used internally for creating a *triangular.test* object
- `plot`  
  character, "all": plot all intermediate steps, "last": plot only the last state
- `recursive`  
  logical, used internally to decide whether a plot should be generated (will be omitted if recursively called)
- `...`  
  additional parameters for `update`

**Author(s)**

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt

**References**


**See Also**

*triangular.test.norm, triangular.test.prop*
Index

*Topic anova
  size.anova, 8
  size_a.three_way, 10
  size_b.three_way, 11
  size_b.two_way, 13
  size_bc.three_way, 14
  size_c.three_way, 15
  size_n.one_way.model_1, 17
  size_n.three_way, 18
  size_n.two_way, 20

*Topic datasets
  cattle, 2
  had, 4
  heights, 5
  hemp, 6

*Topic doe
  OPDOE-undocumented, 6

*Topic lm
  design.reg.polynom, 3
  design.regression, 4
  print.design.regression, 7

*Topic test
  print.triangular.test, 7
  triangular.test, 22
  triangular.test.norm, 23
  triangular.test.prop, 25
  update.triangular.test, 27

add.size.mean.two.stage
  (OPDOE-undocumented), 6

Beta (OPDOE-undocumented), 6
bibd (OPDOE-undocumented), 6

Cattle (OPDOE-undocumented), 6

conf.matrix (OPDOE-undocumented), 6
Confint.mean.two.stage
  (OPDOE-undocumented), 6
conversion.low.high.two.levels
  (OPDOE-undocumented), 6

design.regression.polynom
  (design.reg.polynom), 3

had, 4
had100 (had), 4
had116 (had), 4
had156 (had), 4
had172 (had), 4
had188 (had), 4
had236 (had), 4
had244 (had), 4
had52 (had), 4
had92 (had), 4

is.hadamard (OPDOE-undocumented), 6

jacobsthal (OPDOE-undocumented), 6

legendre.symbol (OPDOE-undocumented), 6

myGF (OPDOE-undocumented), 6
myredu.modp (OPDOE-undocumented), 6

ncp (OPDOE-undocumented), 6

OPDOE-undocumented, 6

optim.size.n.three_way_nested.model_4_a
  (OPDOE-undocumented), 6
optim_size_n.three_way_nested.model_4_c
(undocumented), 6
paley (undocumented), 6
paley2 (undocumented), 6
plot.triangular.test
(undocumented), 6
print.bibd (undocumented), 6
print.design.regression, 7
print.triangular.test, 7
seqadd.confint (undocumented), 6
seqadd.normal (undocumented), 6
size.anova, 8, 11–16, 18, 19, 21
size.comparing.variances
(undocumented), 6
size.comparing.probs.mc Nemar
(undocumented), 6
size.multiple_t.test
(undocumented), 6
size.prop.confint (undocumented), 6
size.prop.test (undocumented), 6
size.prop.test.two_sample
(undocumented), 6
size.selection.bechhofer
(undocumented), 6
size.seq.select.mean
(undocumented), 6
size.t.test (undocumented), 6
size.variance.confint
(undocumented), 6
size_a.three_way, 10
size_a.three_way.mixed.cxbina.model_3_c
(size_a.three_way), 10
size_a.three_way.mixed.cxbina.model_7_c
(size_a.three_way), 10
size_ab.three_way.mixed.cxbina.model_7_c
(size_a.three_way), 10
size_b.three_way, 11
size_b.three_way.mixed_ab_in_c.model_3_a
(size_b.three_way), 11
size_b.three_way.mixed.cxbina.model_4_a
(size_b.three_way), 11
size_b.three_way.mixed.cxbina.model_4_axc
(size_b.three_way), 11
size_b.three_way.mixed.cxbina.model_4_c
(size_b.three_way), 11
size_b.three_way.mixed.cxbina.model_7_c
(size_b.three_way), 11
size_b.three_way_nested.model_6_a
(size_b.three_way), 11
size_b.three_way_nested.model_6_b
(size_b.three_way), 11
size_b.three_way_nested.b.random_a_fixed_a
(size_b.three_way), 13
size_b.three_way_nested.b.random_a_fixed_a
(size_b.three_way), 13
size_bc.three_way, 14
size_bc.three_way.cross.model_4_a_case1
(size_bc.three_way), 14
size_bc.three_way.cross.model_4_a_case2
(size_bc.three_way), 14
size_bc.three_way.mixed.cxbina.model_6_a_case1
(size_bc.three_way), 14
size_bc.three_way.mixed.cxbina.model_6_a_case2
(size_bc.three_way), 14
size_c.three_way, 15
size_c.three_way.cross.model_3_a
(size_c.three_way), 15
size_c.three_way.cross.model_3_axb
(size_c.three_way), 15
size_c.three_way.mixed_ab_in_c.model_5_a
(size_c.three_way), 15
size_c.three_way.mixed_ab_in_c.model_5_axb
(size_c.three_way), 15
size_c.three_way.mixed_ab_in_c.model_5_b
(size_c.three_way), 15
size_c.three_way.mixed_ab_in_c.model_6_b
(size_c.three_way), 15
size_c.three_way.mixed.cxbina.model_5_a
(size_c.three_way), 15
size_c.three_way.mixed.cxbina.model_5_b
(size_c.three_way), 15
size_c.three_way.mixed.cxbina.model_7_b
(size_c.three_way), 15
size_c.three_way.mixed.cxbina.model_7_c
(size_c.three_way), 15
size_c.three_way_nested.model_5_a
(size_c.three_way), 15
size_c.three_way_nested.model_5_b
(size_c.three_way), 15
size_c.three_way_nested.model_7_b
(size_c.three_way), 15
size_n.one_way.model_1, 17
size_n.regII.ci_b1
(undocumented), 6
size_n.regII.ci_exp
(undocumented), 6
INDEX

size_n.regII.ci_rho  
(OPDOE-undocumented), 6
size_n.regII.test_rho  
(OPDOE-undocumented), 6
size_n.regII.test_rho2  
(OPDOE-undocumented), 6
size_n.regII.test_rho_2  
(OPDOE-undocumented), 6
size_n.three_way, 18
size_n.three_way_cross.model_1_a  
(size_n.three_way), 18
size_n.three_way_cross.model_1_axb  
(size_n.three_way), 18
size_n.three_way_cross.model_1_axbxc  
(size_n.three_way), 18
size_n.three_way_mixed_ab_in_c.model_1_a  
(size_n.three_way), 18
size_n.three_way_mixed_ab_in_c.model_1_axb  
(size_n.three_way), 18
size_n.three_way_mixed_ab_in_c.model_1_b  
(size_n.three_way), 18
size_n.three_way_mixed_ab_in_c.model_1_c  
(size_n.three_way), 18
size_n.three_way_mixed_ab_in_c.model_3_c  
(size_n.three_way), 18
size_n.three_way_mixed_ab_in_c.model_4_c  
(size_n.three_way), 18
size_n.three_way_mixed_cxbina.model_1_a  
(size_n.three_way), 18
size_n.three_way_mixed_cxbina.model_1_axc  
(size_n.three_way), 18
size_n.three_way_mixed_cxbina.model_1_b  
(size_n.three_way), 18
size_n.three_way_mixed_cxbina.model_1_bxc  
(size_n.three_way), 18
size_n.three_way_mixed_cxbina.model_1_c  
(size_n.three_way), 18
size_n.three_way_mixed_cxbina.model_3_b  
(size_n.three_way), 18
size_n.three_way_mixed_cxbina.model_3_bxc  
(size_n.three_way), 18
size_n.three_way_nested.model_1_a  
(size_n.three_way), 18
size_n.three_way_nested.model_1_b  
(size_n.three_way), 18
size_n.three_way_nested.model_1_c  
(size_n.three_way), 18
size_n.three_way_nested.model_3_b  
(size_n.three_way), 18
size_n.three_way_nested.model_3_c  
(size_n.three_way), 18
size_n.three_way_nested.model_4_a  
(size_n.three_way), 18
size_n.three_way_nested.model_4_c  
(OPDOE-undocumented), 6
size_n.three_way_nested.model_8_c  
(size_n.three_way), 18
size_n.two_way, 20
size_n.two_way_cross.model_1_a  
(size_n.two_way), 20
size_n.two_way_cross.model_1_axb  
(size_n.two_way), 20
size.n.three_way_nested.a.random_b_fixed_b  
(size_n.two_way), 20
size.n.three_way_nested.model_1_test_factor_a  
(size_n.two_way), 20
size.n.three_way_nested.model_1_test_factor_b  
(size_n.two_way), 20
sizes.confint.welch  
(OPDOE-undocumented), 6
sylvester  
(OPDOE-undocumented), 6
triangular.test, 8, 22, 24, 26, 27
triangular.test.norm, 8, 22, 23, 26, 27
triangular.test.prop, 8, 22–24, 25, 27
update.triangular.prop, 24, 26, 27
walsh  
(OPDOE-undocumented), 6