

# Package ‘OPDOE’

January 20, 2025

**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  
``Optimal Experimental Design with R" by Rasch D. et. al (ISBN 9781439816974).

**Imports** mvtnorm, orthopolynom, nlme, crossdes, polynom

**Depends** gmp

**License** GPL (>= 2)

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2018-03-17 22:49:18 UTC

## Contents

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

size_c.three_way . . . . .	15
size_n.one_way.model_1 . . . . .	17
size_n.three_way . . . . .	18
size_n.two_way . . . . .	20
triangular.test . . . . .	22
triangular.test.norm . . . . .	23
triangular.test.prop . . . . .	25
update.triangular.test . . . . .	27

<b>Index</b>	<b>28</b>
--------------	-----------

---

cattle	<i>Cattle data</i>
--------	--------------------

---

## 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] 132 128 135 121 138 173 166 172 176 169 ...

## Author(s)

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

## References

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt: Optimal Experimental Design with R, Chapman and Hall/CRC, 2011

## Examples

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

---

design.reg.polynom      *Design for Polynomial Regression*

---

### Description

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

Needs specification of order of polynomial, 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

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt: Optimal Experimental Design with R, Chapman and Hall/CRC, 2011

**Examples**

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

---

design.regression      *Regression Design Object*

---

**Description**

An `design.regression` object is created with [design.regression.polynom](#)

**Arguments**

A `triangular.test` object is a list of

model	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**

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt: Optimal Experimental Design with R, Chapman and Hall/CRC, 2011

**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	<i>male / female heights data</i>
---------	-----------------------------------

---

**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**

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt: Optimal Experimental Design with R, Chapman and Hall/CRC, 2011

**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

*Hemp data*

---

**Description**

age and height of hemp plants.

**Usage**

data(hemp)

**Format**

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

x a numeric vector

y a numeric vector

**References**

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt: Optimal Experimental Design with R, Chapman and Hall/CRC, 2011

---

OPDOE-undocumented

*(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**

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt: Optimal Experimental Design with R, Chapman and Hall/CRC, 2011

---

```
print.design.regression
```

*Prints a regression design object*

---

### Description

Print method for a [design.regression](#) object.

### Usage

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

### Arguments

x	<a href="#">design.regression</a> object
epl	integer, entries per line
...	additional print arguments

### Author(s)

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

### References

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt: Optimal Experimental Design with R, Chapman and Hall/CRC, 2011

### See Also

[design.regression](#)

---

```
print.triangular.test
```

*Print method for Triangular Test Objects*

---

### Description

Prints a `triangular.test` object.

### Usage

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

**Arguments**

x                    [triangular.test](#) object  
 ...                additional paramters for print

**Author(s)**

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

**References**

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt: Optimal Experimental Design with R, Chapman and Hall/CRC, 2011

**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 abcABC, 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 describiung 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
c	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 "minimin" 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

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt: Optimal Experimental Design with R, Chapman and Hall/CRC, 2011

### 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="minimin")
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,
```

```

        alpha=0.05, beta=0.1, delta=0.5, cases="maximin")
size.anova(model="aBxC", hypothesis="a",
           assumption="sigma_AC=0, b=c", a=6, n=2,
           alpha=0.05, beta=0.1, delta=0.5, cases="minimin")

size.anova(model="a>B>c", hypothesis="c", a=6, b=2, c=4,
           alpha=0.05, beta=0.1, delta=0.5, case="maximin")
size.anova(model="a>B>c", hypothesis="c", a=6, b=20, c=4,
           alpha=0.05, beta=0.1, delta=0.5, case="maximin")

size.anova(model="a>B>c", hypothesis="c", a=6, b=NA, c=4,
           alpha=0.05, beta=0.1, delta=0.5, case="maximin")

size.anova(model="(axb)>c", hypothesis="a", a=6, b=5, c=4,
           alpha=0.05, beta=0.1, delta=0.5, case="maximin")
size.anova(model="(axb)>c", hypothesis="a", a=6, b=5, c=4,
           alpha=0.05, beta=0.1, delta=0.5, case="minimin")

size.anova(model="(axb)>c", hypothesis="a", a=6, b=5, c=4,
           alpha=0.05, beta=0.1, delta=0.5, case="maximin")
size.anova(model="(axb)>c", hypothesis="a", a=6, b=5, c=4,
           alpha=0.05, beta=0.1, delta=0.5, case="minimin")

```

---

size_a.three_way	<i>Three-way analysis of variance – mixed classification (<math>A \times B</math>) <math>\succ</math> <math>C</math> model III and VII</i>
------------------	--

---

### Description

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

### Usage

```

size_a.three_way_mixed_cxbina.model_3_c(alpha, beta, delta, b, c, n, cases)
size_a.three_way_mixed_cxbina.model_7_c(alpha, beta, delta, b, c, n, cases)
size_ab.three_way_mixed_cxbina.model_7_c(alpha, beta, delta, c, n, cases)

```

### 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 "minimin" sizes are to be determined

**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**

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt: Optimal Experimental Design with R, Chapman and Hall/CRC, 2011

**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, 5, 4, 1, "minimin")
size_ab.three_way_mixed_cxbina.model_7_c(0.05,0.1,0.50, 5,2, "maximin")
size_ab.three_way_mixed_cxbina.model_7_c(0.05,0.1,0.50, 5,2, "minimin")
```

---

size_b.three_way	<i>Three-way analysis of variance – nested and mixed classification <math>A \succ B \succ C</math> and <math>(A \times B) \succ C</math> model III, IV and VII</i>
------------------	--

---

**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**

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt: Optimal Experimental Design with R, Chapman and Hall/CRC, 2011

**See Also**

[size.anova](#)

**Examples**

```
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")
```

---

size_b.two_way	<i>Design for Two-Way ANOVA</i>
----------------	---------------------------------

---

**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 "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**

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt: Optimal Experimental Design with R, Chapman and Hall/CRC, 2011

**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")

```

---

size_bc.three_way	<i>Three-way analysis of variance – cross classification (A in B) x C – model IV, Three-way analysis of variance – mixed classification (A in B) x C model VI</i>
-------------------	---

---

**Description**

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

**Usage**

```

size_bc.three_way_cross.model_4_a_case1(alpha, beta, delta, a, n, cases)
size_bc.three_way_cross.model_4_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.

**Author(s)**

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

**References**

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt: Optimal Experimental Design with R, Chapman and Hall/CRC, 2011

**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_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_c.three_way	<i>Three-way analysis of variance – several cross-, nested and mixed classifications.</i>
------------------	---

---

**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)
size_c.three_way_nested.model_5_b     (alpha, beta, delta, a, b, n, cases)
size_c.three_way_nested.model_7_b     (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**

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt: Optimal Experimental Design with R, Chapman and Hall/CRC, 2011

**See Also**

[size.anova](#)

**Examples**

```
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")

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")

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")

size_c.three_way_mixed_ab_in_c.model_5_axb(0.05, 0.1, 0.5, 6, 5, 1, "maximin")
size_c.three_way_mixed_ab_in_c.model_5_axb(0.05, 0.1, 0.5, 6, 5, 1, "minimin")

size_c.three_way_mixed_ab_in_c.model_5_b(0.05, 0.1, 0.5, 6, 5, 1, "maximin")
```

```

size_c.three_way_mixed_ab_in_c.model_5_b(0.05, 0.1, 0.5, 6, 5, 1, "minimin")

size_c.three_way_mixed_ab_in_c.model_6_b(0.05, 0.1, 0.5, 6, 5, 1, "maximin")
size_c.three_way_mixed_ab_in_c.model_6_b(0.05, 0.1, 0.5, 6, 5, 1, "minimin")

size_c.three_way_mixed_cxbina.model_5_a(0.05, 0.1, 0.5, 6, 5, 2, "maximin")
size_c.three_way_mixed_cxbina.model_5_a(0.05, 0.1, 0.5, 6, 5, 2, "minimin")

size_c.three_way_mixed_cxbina.model_5_b(0.05, 0.1, 0.5, 6, 5, 2, "maximin")
size_c.three_way_mixed_cxbina.model_5_b(0.05, 0.1, 0.5, 6, 5, 2, "minimin")

size_c.three_way_mixed_cxbina.model_7_b(0.05, 0.1, 0.5, 6, 5, 2, "maximin")
size_c.three_way_mixed_cxbina.model_7_b(0.05, 0.1, 0.5, 6, 5, 2, "minimin")

size_c.three_way_nested.model_5_a(0.05, 0.1, 0.5, 6, 5, 2, "maximin")
size_c.three_way_nested.model_5_a(0.05, 0.1, 0.5, 6, 5, 2, "minimin")

size_c.three_way_nested.model_5_b(0.05, 0.1, 0.5, 6, 5, 2, "maximin")
size_c.three_way_nested.model_5_b(0.05, 0.1, 0.5, 6, 5, 2, "minimin")

size_c.three_way_nested.model_7_b(0.05, 0.1, 0.5, 6, 4, 1, "maximin")
size_c.three_way_nested.model_7_b(0.05, 0.1, 0.5, 6, 4, 1, "minimin")

```

---

size\_n.one\_way.model\_1

*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 [size.anova](#) which allows a cleaner notation.

**Author(s)**

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

**References**

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt: Optimal Experimental Design with R, Chapman and Hall/CRC, 2011

**See Also**

[size.anova](#)

**Examples**

```
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")
```

---

size\_n.three\_way      *Design for Three-Way ANOVA*

---

**Description**

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

**Usage**

```
size_n.three_way_cross.model_1_a      (alpha, beta, delta, a, b, c, cases)
size_n.three_way_cross.model_1_axb    (alpha, beta, delta, a, b, c, cases)
size_n.three_way_cross.model_1_axbxc (alpha, beta, delta, a, b, c, cases)
size_n.three_way_mixed_ab_in_c.model_1_a (alpha, beta, delta, a, b, c, cases)
size_n.three_way_mixed_ab_in_c.model_1_b (alpha, beta, delta, a, b, c, cases)
size_n.three_way_mixed_ab_in_c.model_1_c (alpha, beta, delta, a, b, c, cases)
size_n.three_way_mixed_ab_in_c.model_3_c (alpha, beta, delta, a, b, c, cases)
size_n.three_way_mixed_ab_in_c.model_4_c (alpha, beta, delta, a, b, c, cases)
size_n.three_way_mixed_cxbina.model_1_a (alpha, beta, delta, a, b, c, cases)
size_n.three_way_mixed_cxbina.model_1_axc (alpha, beta, delta, a, b, c, cases)
size_n.three_way_mixed_cxbina.model_1_b (alpha, beta, delta, a, b, c, cases)
size_n.three_way_mixed_cxbina.model_1_bxc (alpha, beta, delta, a, b, c, cases)
size_n.three_way_mixed_cxbina.model_1_c (alpha, beta, delta, a, b, c, cases)
size_n.three_way_mixed_cxbina.model_3_b (alpha, beta, delta, a, b, c, cases)
```

size_n.three_way_mixed_cxbina.model_3_bxc	(alpha, beta, delta, a, b, c, cases)
size_n.three_way_nested.model_1_a	(alpha, beta, delta, a, b, c, cases)
size_n.three_way_nested.model_1_b	(alpha, beta, delta, a, b, c, cases)
size_n.three_way_nested.model_1_c	(alpha, beta, delta, a, b, c, cases)
size_n.three_way_nested.model_3_b	(alpha, beta, delta, a, b, c, cases)
size_n.three_way_nested.model_3_c	(alpha, beta, delta, a, b, c, cases)
size_n.three_way_nested.model_4_a	(alpha, beta, delta, a, b, c, cases)
size_n.three_way_nested.model_8_c	(alpha, beta, delta, a, b, c, 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
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**

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt: Optimal Experimental Design with R, Chapman and Hall/CRC, 2011

**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_axb(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way_cross.model_1_axbxc(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way_cross.model_1_axbxc(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_b(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way_mixed_ab_in_c.model_1_b(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way_mixed_ab_in_c.model_1_c(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way_mixed_ab_in_c.model_1_c(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way_mixed_ab_in_c.model_3_c(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way_mixed_ab_in_c.model_3_c(0.05, 0.1, 0.5, 6, 5, 4, "minimin")
size_n.three_way_mixed_ab_in_c.model_4_c(0.05, 0.1, 0.5, 6, 5, 4, "maximin")
size_n.three_way_mixed_ab_in_c.model_4_c(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")

```

**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**

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt: Optimal Experimental Design with R, Chapman and Hall/CRC, 2011

**See Also**

[size.anova](#)

**Examples**

```

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	<i>Triangular Test Object</i>
-----------------	-------------------------------

---

**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 <code>triangular.test.prop</code>
p1	...
p2	...
mu0	parameter describing the Null hypothesis, see <code>triangular.test.norm</code>
mu1	...
mu2	...
result	character, outcome of the test, "H0" or "H1"
step	total number of steps

and some more components for internal use.

**Author(s)**

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

**References**

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt: Optimal Experimental Design with R, Chapman and Hall/CRC, 2011

**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**

<code>x</code>	initial data for group x, at least 1 entry.
<code>y</code>	initial data for group y, at least 1 entry for a two sample test, otherwise omitted.
<code>mu0</code>	specifies Null and alternative hypothesis, see Details below.
<code>mu1</code>	specifies Null and alternative hypothesis, see Details below.
<code>mu2</code>	specifies Null and alternative hypothesis, see Details below.
<code>delta</code>	The minimum difference to be detected, alternative way to specify $\mu_2 = \mu_1 + \delta$ , see above, use either this or <code>mu2</code> .
<code>sigma</code>	prior sigma.
<code>sigma2</code>	prior sigma for group 2 if different than for group 1.
<code>alpha</code>	Risk of 1st kind
<code>beta</code>	Risk of 2nd kind
<code>plot</code>	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  $\mu_1 = \mu_2 = \mu_1$  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 `triangular.test`, to be used for later update steps.

**Note**

A two-sided test may be specified by supplying both `mu1` and `mu2`, even unsymmetric if needed.

**Author(s)**

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

**References**

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt: Optimal Experimental Design with R, Chapman and Hall/CRC, 2011

**See Also**

`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,y=male[4])
tt <- update(tt,x=female[5])
tt <- update(tt,y=male[5])
```

```

tt <- update(tt,x=female[6])
tt <- update(tt,y=male[6])
tt <- update(tt,x=female[7])
tt <- update(tt,y=male[7])
# 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

```

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 $p_2 = p_1 + \text{delta}$ , see above, use either this or p2.

## Details

One-sample:

This function performs a one- or two-sided sequential Test for  $p = p_1$  versus

$p > p_2$ , if  $p_2 > p_1$  (one-sided)

$p < p_2$ , if  $p_2 < p_1$  (one-sided)

$p < p_0$  or  $p > p_2$ , if  $p_2 > p_1$  and  $p_0 < p_1$  (two-sided, possibly unsymmetric)

Two-sample:

This function performs a one- or two-sided sequential Test for equal proportions  $p_1 = p_2 = p_0$  versus

$p_1 > p_2$ , if  $p_2 > p_1$  (one-sided)

$p_1 < p_2$ , if  $p_2 < p_1$  (one-sided)

$p_1 < p_0$  or  $p_1 > p_2$ , if  $p_2 > p_1$  and  $p_0 < p_1$  (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

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt: Optimal Experimental Design with R, Chapman and Hall/CRC, 2011

### See Also

[triangular.test](#), [triangular.test.norm](#), [update.triangular.test](#)

### Examples

```
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,p0=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)
```

---

`update.triangular.test`*Print method for Triangular Test Objects*

---

**Description**

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

**Usage**

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

**Arguments**

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

**Author(s)**

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

**References**

Dieter Rasch, Juergen Pilz, L.R. Verdooren, Albrecht Gebhardt: Optimal Experimental Design with R, Chapman and Hall/CRC, 2011

**See Also**

[triangular.test.norm](#), [triangular.test.prop](#)

# Index

- \* **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
- \* **datasets**
  - cattle, 2
  - had, 4
  - heights, 5
  - hemp, 6
- \* **doe**
  - OPDOE-undocumented, 6
- \* **lm**
  - design.reg.polynom, 3
  - design.regression, 4
  - print.design.regression, 7
- \* **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, 2
- conf.matrix (OPDOE-undocumented), 6
- Confint.mean.two\_stage  
(OPDOE-undocumented), 6
- conversion.low\_high.two\_levels  
(OPDOE-undocumented), 6
- decide.triangular.test  
(OPDOE-undocumented), 6
- defining\_relation.fractional\_factorial.two\_levels  
(OPDOE-undocumented), 6
- delta.t.test (OPDOE-undocumented), 6
- design.reg.polynom, 3
- design.regression, 3, 4, 7
- design.regression.polynom, 4
- 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
- hadamard.matrix (OPDOE-undocumented), 6
- hadamard.table (OPDOE-undocumented), 6
- heights, 5
- hemp, 6
- 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  
(OPDOE-undocumented), 6
- paley (OPDOE-undocumented), 6
- paley2 (OPDOE-undocumented), 6
- plot.triangular.test  
(OPDOE-undocumented), 6
- print.bibd (OPDOE-undocumented), 6
- print.design.regression, 7
- print.triangular.test, 7
- seqadd.confint (OPDOE-undocumented), 6
- seqadd.normal (OPDOE-undocumented), 6
- size.anova, 8, 11–16, 18, 19, 21
- size.comparing.variances  
(OPDOE-undocumented), 6
- size.comparing\_probs.mcnemar  
(OPDOE-undocumented), 6
- size.multiple\_t.test  
(OPDOE-undocumented), 6
- size.prop.confint (OPDOE-undocumented),  
6
- size.prop.test (OPDOE-undocumented), 6
- size.prop\_test.two\_sample  
(OPDOE-undocumented), 6
- size.selection.bechhofer  
(OPDOE-undocumented), 6
- size.seq\_select.mean  
(OPDOE-undocumented), 6
- size.t.test (OPDOE-undocumented), 6
- size.variance.confint  
(OPDOE-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.two\_way, 13
- size\_b.two\_way\_cross.mixed\_model\_a\_fixed\_a  
(size\_b.two\_way), 13
- size\_b.two\_way\_nested.b\_random\_a\_fixed\_a  
(size\_b.two\_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\_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  
(OPDOE-undocumented), 6
- size\_n.regII.ci\_exp  
(OPDOE-undocumented), 6

- 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.two\_way\_nested.a\_random\_b\_fixed\_b  
(size\_n.two\_way), 20
- size\_n.two\_way\_nested.model\_1\_test\_factor\_a  
(size\_n.two\_way), 20
- size\_n.two\_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, 23, 26, 27
- triangular.test.prop, 8, 22–24, 25, 27
- update.triangular.test, 24, 26, 27
- walsh (OPDOE-undocumented), 6