

# Package ‘ibd’

May 8, 2026

**Version** 1.6

**Date** 2024-01-23

**Title** Incomplete Block Designs

**Author** B N Mandal [aut, cre]

**Maintainer** B N Mandal <mandal.stat@gmail.com>

**Depends** R (>= 3.1.1)

**Imports** lpSolve, car, emmeans, multcomp

**Suggests** multcompView

**Description** A collection of several utility functions related to binary incomplete block designs. Contains function to generate A- and D-efficient binary incomplete block designs with given numbers of treatments, number of blocks and block size. Contains function to generate an incomplete block design with specified concurrence matrix. There are functions to generate balanced treatment incomplete block designs and incomplete block designs for test versus control treatments comparisons with specified concurrence matrix. Allows performing analysis of variance of data and computing estimated marginal means of factors from experiments using a connected incomplete block design. Tests of hypothesis of treatment contrasts in incomplete block design set up is supported.

**License** GPL (>= 2)

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2024-01-23 09:50:02 UTC

## Contents

aov.ibd . . . . .	2
A_eff . . . . .	3
bibd . . . . .	4
btib . . . . .	6
btib1 . . . . .	7
Cmatrix . . . . .	8
design_to_N . . . . .	9
D_eff . . . . .	10

ibd . . . . .	10
ibddata . . . . .	12
ibdtvc . . . . .	12
is.connected . . . . .	14
is.equir . . . . .	14
is.orthogonal . . . . .	15
is.proper . . . . .	16
is.vb . . . . .	16
N_to_design . . . . .	17
randomize . . . . .	18

## Index 19

---

aov.ibd	<i>Analysis of Variance, Estimated Marginal Means and Contrast Analysis of Data from An Incomplete Block Design</i>
---------	---

---

### Description

Performs intrablock analysis of variance of data from experiments using a block design. It also computes estimated marginal means of the factor variables (e.g. treatments) and optionally estimates and tests the contrasts of factor variables (e.g. treatments).

### Usage

```
aov.ibd(formula, specs, data, contrast, joint=FALSE, details=FALSE, sort=TRUE, by=NULL,
alpha=0.05, Letters = "ABCDEFGHIJ", ...)
```

### Arguments

formula	A formula specifying the model of the form response~treatment+block or response~block+treatment. Make sure the treatment and blocks are factor variables.
specs	A character vector specifying the names of the factors over which estimated marginal means are desired
data	A data frame in which the variables specified in the formula will be found. If missing, the variables are searched for in the standard way.
contrast	A matrix whose rows are contrasts of factors (e.g. treatments)
joint	If contrast argument has more than one row, then whether a joint test of the contrasts will be performed. Default is FALSE. If joint=TRUE, a check is performed whether the contrasts are pairwise orthogonal or not and then if orthogonal, joint test is performed.
details	Logical, if details=TRUE then all objects including lm object from lm(), emm-Grid object from emmeans() are returned. Default is FALSE.
sort	Logical value determining whether the least square means are sorted before the comparisons are produced. Default is TRUE.

by	Character value giving the name or names of variables by which separate families of comparisons are tested. If NULL, all means are compared.
alpha	Numeric value giving the significance level for the comparisons
Letters	Characters to be used for compact letter display of groups of factor variables over which least square means are computed. Default is english alphabet capital letters "ABCDEFGHJIJ"
...	Not used

### Details

The function makes use of `lm()` function in R and `Anova()` function in `car` package with specification of Type III sum of squares and `emmeans()`, `contrast()` functions in `emmeans()` package, `cld()` function in `multcomp` package and combines the results in a single place.

### Value

Returns a list with following components

<code>lm.obj</code>	An object of class <code>lm</code> if <code>details=TRUE</code>
<code>ANOVA.table</code>	ANOVA table from the fitted <code>lm</code> object
<code>EMMEANS</code>	Estimated marginal means means with compact letter display
<code>contrast.analysis</code>	Contrast analysis result if contrast matrix was supplied

### Author(s)

Baidya Nath Mandal <mandal.stat@gmail.com>

### Examples

```
data(ibddata)
aov.ibd(y~factor(trt)+factor(blk), data=ibddata)
contrast=matrix(c(1,-1,0,0,0,0,0,0,0,0,1,-1,0,0,0,0), nrow=2, byrow=TRUE)
aov.ibd(y~factor(trt)+factor(blk), specs="trt", data=ibddata, contrast=contrast)
```

---

A\_eff

*A-efficiency of A Binary Incomplete Block Design*

---

### Description

Computes lower bound to A-efficiency of a binary incomplete block design. Treatment by block incidence matrix of the design is to be supplied as input to the function.

### Usage

`A_eff(N)`

**Arguments**

N Treatment by block incidence matrix

**Value**

Aeff A-efficiency

**Author(s)**

Baidya Nath Mandal <mandal.stat@gmail.com>

**Examples**

```
N=matrix(c(1,0,0,0,1,0,1,0,0,0,1,0,1,1,0,0,1,0,1,1,0,1,1,0,1,0,1,1,0,0,0,0,1,1,0,0,0,1,1,1,0,0,0,1,0,0,
1,0,1,1,0,0),nrow=7,byrow=TRUE)
A_eff(N)
```

---

bibd

*Balanced Incomplete Block Design for Given Parameters*

---

**Description**

Generates a balanced incomplete block design with given number of treatments ( $v$ ), number of blocks ( $b$ ), number of replications ( $r$ ), block size ( $k$ ) and number of concurrences ( $\lambda$ ).

**Usage**

```
bibd(v,b,r,k,lambda,ntrial=5,pbar=FALSE)
```

**Arguments**

v number of treatments

b number of blocks

r number of replications

k block size

lambda number of concurrences

ntrial number of trials. Default value is 5.

pbar logical value indicating whether progress bar will be displayed or not. Default is FALSE

**Value**

v	number of treatments
b	number of blocks
r	number of replications
k	block size
lambda	number of concurrences
design	block contents in a b by k matrix
N	treatments by blocks incidence matrix of the generated design
NNP	concurrence matrix of the generated design
Aeff	Lower bound to the A-efficiency of the generated design
Deff	Lower bound to the D-efficiency of the generated design

**Note**

The function works best for values of number of treatments (v) up to 30 and block size (k) up to 10. However, for block size (k) up to 3, much larger values of number of treatments (v) may be used.

**Author(s)**

Baidya Nath Mandal <mandal.stat@gmail.com>

**References**

Mandal, B. N., Gupta, V. K. and Parsad, R. (2013). Application of optimization techniques for construction of incomplete block designs. Project report, IASRI, New Delhi.

Mandal, B. N., Gupta, V. K., & Parsad, R. (2014). Efficient Incomplete Block Designs Through Linear Integer Programming. American Journal of Mathematical and Management Sciences, 33(2), 110-124.

Mandal, B. N. (2015). Linear integer programming approach to construction of balanced incomplete block designs. Communications in Statistics-Simulation and Computation, 44:6, 1405-1411.

**Examples**

```
bibd(7,7,3,3,1)
bibd(9,12,4,3,1)
```

---

 btib

*Balanced Treatment Incomplete Block Designs*


---

**Description**

Generates a balanced treatment incomplete block design for specified parameters.

**Usage**

```
btib(v,b,r,r0,k,lambda,lambda0,ntrial=5,pbar=FALSE)
```

**Arguments**

v	number of test treatments
b	number of blocks
r	number of replications of test treatments
r0	number of replications of the control treatment
k	block size
lambda	number of concurrences among test treatments
lambda0	number of concurrences between test treatments and the control treatment
ntrial	number of trials. Default is 5.
pbar	logical value indicating whether progress bar will be displayed or not. Default is FALSE.

**Value**

v	number of test treatments
b	number of blocks
r	number of replications of test treatments
r0	number of replications of the control treatment
k	block size
lambda	number of concurrences among test treatments
lambda0	number of concurrences between test treatments and the control treatment
design	generated block design
N	treatment by block incidence matrix of the generated block design
NNP	concurrence matrix of the generated design
Aeff	A-efficiency of the generated design

**Note**

The function works best for values of number of treatments (v) up to 30 and block size (k) up to 10. However, for block size (k) up to 3, much larger values of number of treatments (v) may be used.

**Author(s)**

Baidya Nath Mandal <mandal.stat@gmail.com>

**References**

Mandal, B. N., Gupta, V. K. and Parsad, R. (2013). Application of optimization techniques for construction of incomplete block designs. Project report, IASRI, New Delhi.

Mandal, B. N., Gupta, V. K., & Parsad, R. (2014). Balanced treatment incomplete block designs through integer programming, *Communications in Statistics - Theory and Methods*, 46:8, 3728-3737.

**Examples**

```
btib(4,6,3,6,3,1,3,10)
```

---

 btib1

---

*Balanced Treatment Incomplete Block Designs*


---

**Description**

Generates a balanced treatment incomplete block design for specified parameters by searching all possible combinations.

**Usage**

```
btib1(v,b,r,r0,k,lambda,lambda0)
```

**Arguments**

v	number of test treatments
b	number of blocks
r	number of replications of test treatments
r0	number of replications of the control treatment
k	block size
lambda	number of concurrences among test treatments
lambda0	number of concurrences between test treatments and control treatment

**Value**

v	number of test treatments
b	number of blocks
r	number of replications of test treatments
r0	number of replications of control treatment
k	block size

$\lambda$	number of concurrences among test treatments
$\lambda_{0}$	number of concurrences between test treatments and control treatment
design	generated block design
N	treatment by block incidence matrix of the generated block design
NNP	concurrence matrix of the generated design
Aeff	A-efficiency of the generated design

**Note**

The function works best for values of number of treatments ( $v$ ) up to 30 and block size ( $k$ ) up to 10. However, for block size ( $k$ ) up to 3, much larger values of number of treatments ( $v$ ) may be used.

**Author(s)**

Baidya Nath Mandal <mandal.stat@gmail.com>

**References**

- Mandal, B. N., Gupta, V. K. and Parsad, R. (2013). Application of optimization techniques for construction of incomplete block designs. Project report, IASRI, New Delhi.
- MANDAL, B. N., GUPTA, V. K. and PARSAD, R. (2012). Generation of Binary Incomplete Block Design with a Specified Concurrence Matrix. Journal of Statistics & Applications, 7.

**Examples**

btib(4,6,3,6,3,1,3)

---

Cmatrix

*Information Matrix of a Block Design*

---

**Description**

Gives the information matrix from a given treatment by block incidence matrix of a block design

**Usage**

Cmatrix(N)

**Arguments**

N treatment by block incidence matrix

**Value**

Cmatrix v by v information matrix where v is number of treatments

**Author(s)**

Baidya Nath Mandal <mandal.stat@gmail.com>

**Examples**

```
N = matrix(c(1,0,0,0,1,0,1,0,0,0,1,0,1,1,0,0,1,0,1,1,0,1,0,1,1,0,0,0,0,1,1,0,0,0,1,1,1,0,0,0,1,0,0,
,1,0,1,1,0,0),nrow=7,byrow=TRUE)
Cmatrix(N)
Information.Matrix(N)
```

---

design\_to\_N

*Block Design to Treatment by Block Incidence Matrix*

---

**Description**

Generates treatment by block incidence matrix from a given block design

**Usage**

```
design_to_N(design)
N(design)
```

**Arguments**

design            design

**Value**

N                    A treatment by block incidence matrix of order v by b with elements as 0 and 1 where v is the number of treatments and b is the number of blocks

**Author(s)**

Baidya Nath Mandal <mandal.stat@gmail.com>

**Examples**

```
design = matrix(c(1,4,6,5,6,7,3,4,5,2,4,7,1,3,7,2,3,6,1,2,5),nrow=7,byrow=TRUE)
design_to_N(design)
# or alternatively
N(design)
```

---

D\_eff *D-efficiency of a Binary Incomplete Block Design*

---

**Description**

Computes lower bound to D-efficiency of a binary incomplete block design

**Usage**

D\_eff(N)

**Arguments**

N treatment by block incidence matrix

**Value**

Deff lower bound to D-efficiency

**Author(s)**

Baidya Nath Mandal <mandal.stat@gmail.com>

**Examples**

```
N=matrix(c(1,0,0,0,1,0,1,0,0,0,1,0,1,1,0,0,1,0,1,1,0,1,0,1,1,0,0,0,0,1,1,0,0,0,1,1,1,0,0,0,1,0,0,
,1,0,1,1,0,0),nrow=7,byrow=TRUE)
D_eff(N)
```

---

ibd *Binary Incomplete Block Design for Given v, b and k and Optionally, with a Specified Concurrence Matrix*

---

**Description**

Generates an A- and D- efficient binary incomplete block design with given number of treatments(v), number of blocks(b) and block size(k) and optionally with a specified concurrence matrix(NNP).

**Usage**

ibd(v,b,k,NNPo,ntrial=5,pbar=FALSE)

**Arguments**

v	number of treatments
b	number of blocks
k	block size
NNPo	optionally, desired concurrence matrix. If not specified, a nearly balanced concurrence matrix is obtained automatically.
ntrial	number of trials. Default is 5.
pbar	progress bar. Default is FALSE.

**Value**

v	number of treatments
b	number of blocks
k	block size
NNP	specified concurrence matrix
N	incidence matrix of the generated design
design	block contents in a b by k matrix
conc.mat	concurrence matrix of the generated design
A.efficiency	lower bound to A-efficiency of the generated design
D.efficiency	lower bound to D-efficiency of the generated design
time.taken	time taken to generate the design

**Note**

This function works best for values of number of treatments (v) up to 30 and block size (k) up to 10. However, for block size (k) up to 3, much larger values of number of treatments (v) may be used.

**Author(s)**

Baidya Nath Mandal <mandal.stat@gmail.com>

**References**

- Mandal, B. N., Gupta, V. K. and Parsad, R. (2013). Application of optimization techniques for construction of incomplete block designs. Project report, IASRI, New Delhi.
- Mandal, B. N., Gupta, V. K., & Parsad, R. (2014). Efficient Incomplete Block Designs Through Linear Integer Programming. American Journal of Mathematical and Management Sciences, 33(2), 110-124.

**Examples**

```
ibd(v = 7, b = 7, k = 4, pbar=FALSE)
```

---

ibddata

*Data from an Experiment using Incomplete Block Design*

---

**Description**

Data from an experiment using incomplete block design

**Usage**

```
data("ibddata")
```

**Format**

A data frame with 36 observations on the following 3 variables.

trt Treatments

blk Blocks

y The response variable

**Details**

The experiment used a balanced incomplete block design.

**References**

Dey,A. (1986). Theory of block designs. Wiley Eastern Limited, New Delhi.

**Examples**

```
data(ibddata)
```

---

ibdtvc

*Incomplete Block Design for Test vs Control(s) Comparisons*

---

**Description**

Generates an incomplete block design for test vs control(s) comparisons with specified parameters and concurrence matrix.

**Usage**

```
ibdtvc(v1,v2,b,k,NNPo,ntrial=5,pbar=FALSE)
```

**Arguments**

v1	number of test treatments
v2	number of control treatments
b	number of blocks
k	block size
NNPo	desired concurrence matrix
ntrial	number of trials, default is 5
pbar	logical value indicating whether progress bar will be displayed. Default is FALSE.

**Value**

v1=v1,v2=v2,b=b,k=k,design=design,N=N, NNP=NNP,Aeff=Aeff)

v1	number of test treatments
v2	number of control treatments
b	number of blocks
k	block size
design	generated block design
N	treatment by block incidence matrix of the generated block design
NNP	concurrence matrix of the generated design

**Author(s)**

Baidya Nath Mandal <mandal.stat@gmail.com>

**References**

Mandal, B. N., Gupta, V.K. and Parsad, R. (2013). Binary Incomplete Block Designs with a Specified Concurrence Matrix through Integer Programming, to be submitted for publication

**Examples**

```
NNPo=matrix(c(7,3,3,3,3,3,3,3,7,3,3,3,3,3,3,7,3,3,3,3,3,3,7,3,3,3,3,3,3,7,
3,3,3,3,3,3,7,3,3,3,3,3,3,3,9,9,3,3,3,3,3,9,9),nrow=8,byrow=TRUE)
ibdtvc(6,2,15,4,NNPo)
```



**Value**

equir                    equi-replicated

**Author(s)**

Baidya Nath Mandal <mandal.stat@gmail.com>

**Examples**

```
N=matrix(c(1,0,0,0,1,0,1,0,0,0,1,0,1,1,0,0,1,0,1,1,0,1,0,1,1,0,1,1,0,0,0,0,1,1,0,0,0,1,1,1,0,
0,0,1,0,0,1,0,1,1,0,0),nrow=7,byrow=TRUE)
is.equir(N)
```

---

is.orthogonal

*Orthogonality a Block Design*

---

**Description**

Checks whether an incomplete block design is orthogonal or not. Treatment by block incidence matrix of the design is to be supplied as input to the function. If the design is orthogonal, it returns a value of 1 else it returns 0.

**Usage**

```
is.orthogonal(N)
```

**Arguments**

N                    incidence matrix

**Value**

orthogonal            orthogonal

**Author(s)**

Baidya Nath Mandal <mandal.stat@gmail.com>

**Examples**

```
N=matrix(c(1,0,0,0,1,0,1,0,0,0,1,0,1,1,0,0,1,0,1,1,0,1,0,1,1,0,1,1,0,0,0,0,1,1,0,0,0,1,1,1,0,
0,0,1,0,0,1,0,1,1,0,0),nrow=7,byrow=TRUE)
is.orthogonal(N)
```



**Value**

vb                    variance balanced

**Author(s)**

Baidya Nath Mandal <mandal.stat@gmail.com>

**Examples**

```
N=matrix(c(1,0,0,0,1,0,1,0,0,0,1,0,1,1,0,0,1,0,1,1,0,1,0,1,1,0,0,0,0,1,1,0,0,0,1,1,1,0,
0,0,1,0,0,1,0,1,1,0,0),nrow=7,byrow=TRUE)
is.vb(N)
```

---

N\_to\_design

*Block Design from Given Treatment by Block Incidence Matrix*

---

**Description**

Generates the block contents from a given treatment by block incidence matrix

**Usage**

```
N_to_design(N)
design(N)
```

**Arguments**

N                    treatment by block incidence matrix

**Value**

design                a matrix with number of rows equal to number of blocks and number of columns equal to block size. Constant block size is assumed. Treatments are labelled as 1, 2, ..., v.

**Author(s)**

Baidya Nath Mandal <mandal.stat@gmail.com>

**Examples**

```
N=matrix(c(1,0,0,0,1,0,1,0,0,0,1,0,1,1,0,0,1,0,1,1,0,1,0,1,1,0,0,0,0,1,1,0,0,0,1,
1,1,0,0,0,1,0,0,1,0,1,1,0,0),nrow=7,byrow=TRUE)
N_to_design(N)
design(N)
```

---

`randomize`*Randomize a block design*

---

**Description**

Randomize a given block design

**Usage**

```
randomize(design)
```

**Arguments**

`design`            `design`

**Value**

`design`            Block design with a constant block size

**Author(s)**

Baidya Nath Mandal <mandal.stat@gmail.com>

**Examples**

```
design = matrix(c(1,4,6,5,6,7,3,4,5,2,4,7,1,3,7,2,3,6,1,2,5),nrow=7,byrow=TRUE)
randomize(design)
```

# Index

- \* **A-efficiency**
    - A\_eff, 3
    - bibd, 4
    - btib, 6
    - btib1, 7
    - ibd, 10
  - \* **Connectedness**
    - is.connected, 14
  - \* **D-efficiency**
    - bibd, 4
    - D\_eff, 10
    - ibd, 10
  - \* **EMMEANS**
    - aov.ibd, 2
  - \* **Estimated marginal means**
    - aov.ibd, 2
  - \* **Type III sum of squares**
    - aov.ibd, 2
  - \* **analysis of variance**
    - aov.ibd, 2
  - \* **balanced incomplete block design**
    - aov.ibd, 2
    - bibd, 4
  - \* **block design**
    - is.orthogonal, 15
  - \* **concurrence matrix**
    - bibd, 4
    - ibd, 10
  - \* **contrast analysis**
    - aov.ibd, 2
  - \* **datasets**
    - ibddata, 12
  - \* **equi-replicated**
    - is.equir, 14
  - \* **incidence matrix**
    - bibd, 4
    - Cmatrix, 8
    - design\_to\_N, 9
    - ibd, 10
    - N\_to\_design, 17
    - randomize, 18
  - \* **incomplete block design**
    - A\_eff, 3
    - aov.ibd, 2
    - btib, 6
    - btib1, 7
    - Cmatrix, 8
    - D\_eff, 10
    - design\_to\_N, 9
    - ibd, 10
    - ibdtvc, 12
    - is.connected, 14
    - is.equir, 14
    - is.proper, 16
    - is.vb, 16
    - N\_to\_design, 17
    - randomize, 18
  - \* **information matrix**
    - Cmatrix, 8
  - \* **orthogonal**
    - is.orthogonal, 15
  - \* **proper**
    - is.proper, 16
  - \* **test vs control(s) comparison**
    - btib, 6
    - btib1, 7
    - ibdtvc, 12
  - \* **variance balanced**
    - is.vb, 16
- A\_eff, 3  
aov.ibd, 2  
  
bibd, 4  
btib, 6  
btib1, 7  
  
Cmatrix, 8  
D\_eff, 10

`design(N_to_design)`, 17  
`design_to_N`, 9

`ibd`, 10  
`ibddata`, 12  
`ibdtvc`, 12  
Information.Matrix (Cmatrix), 8  
`is.connected`, 14  
`is.equir`, 14  
`is.orthogonal`, 15  
`is.proper`, 16  
`is.vb`, 16

`N(design_to_N)`, 9  
`N_to_design`, 17

`randomize`, 18