Package ‘ibd’

April 24, 2019

Version 1.5

Date 2019-04-24

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. The package contains function to generate A- and D-efficient binary incomplete block designs with given numbers of treatments, number of blocks and block size. The package also 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. Package also 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 2019-04-24 07:10:03 UTC

R topics documented:

`aov.ibd` *Analysis of variance, least square means and contrast analysis of data from a block design*

Description

This function 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 orthoghonal, joint test is performed.
`details`	Logical, if details=TRUE then all objects including lm object from lm(), emmGrid 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 "ABCDEFGHIJ"
`...`	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

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

Author(s)

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

Examples

```
data(ibddata)
aov.ibd(y~factor(trt)+factor(blk),data=ibddata)
contrast=matrix(c(1,-1,0,0,0,0,0,0,0,0,0,1,-1,0,0,0,0,0),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

This function 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)

B N 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)
A_eff(N)
```

`bibd` *Balanced incomplete block design for given parameters*

Description

This function generates a balanced incomplete block design with given number of treaments(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,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
`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)

B N 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,pbar=FALSE)
```

`btib` *balanced treatment incomplete block designs*

Description

This function 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)

B N 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

This function 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
`lambda0`	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)

B N 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 from given treatment by block incidence matrix of a block design*

Description

This function 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)

B N 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)
```

`design_to_N` *block design to treatment by block incidence matrix*

Description

This function generates treatment by block incidence matrix from a given block design

Usage

```
design_to_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)

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

Examples

```
d=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(d)
```

D_eff — *D-efficiency of a binary incomplete block design*

Description

This function computes lower bound to D-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

```
D_eff(N)
```

Arguments

N	treatment by block incidence matrix

Value

Deff	Lower bound to D-efficiency

Author(s)

B N 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

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

Usage

```
ibd(v,b,k,NNPo,ntrial,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
`pbar`	progress bar

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

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)

B N 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

```
v=9
b=12
k=3
ibd(v,b,k,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) comparions*

Description

This function 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)

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

`is.connected` *Connctedness of a binary incomplete block design*

Description

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

Usage

```
is.connected(N)
```

Arguments

`N`	incidence matrix

Value

`connected`	Connctedness

Author(s)

B N 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.connected(N)
```

`is.equir` *Equi-replicateness a binary incomplete block design*

Description

This function checks whether an incomplete block design is equi-replicated or not. Treatment by block incidence matrix of the design is to be supplied as input to the function. If the design is equir-eplicated, it returns a value of 1 else it returns 0.

Usage

```
is.equir(N)
```

Arguments

`N`	incidence matrix

Value

`equir` equi-replicated

Author(s)

B N 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.equir(N)
```

`is.orthogonal` *orthogonality a block design*

Description

This function 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)

B N 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.orthogonal(N)
```

is.proper — *proper binary incomplete block design*

Description

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

Usage

```
is.proper(N)
```

Arguments

N	incidence matrix

Value

proper	proper

Author(s)

B N 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.proper(N)
```

is.vb — *Variance balancedness of a binary incomplete block design*

Description

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

Usage

```
is.vb(N)
```

Arguments

N	incidence matrix

Value

vb variance balanced

Author(s)

B N 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 *Treatment by block incidence matrix from given block design*

Description

This function generates the block contents from a given treatment by block incidence matrix

Usage

```
N_to_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 numbered as 1, 2, ..., v

Author(s)

B N 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)
```

Index