Package ‘crossdes’

October 12, 2022

Date 2022-04-04
Title Construction of Crossover Designs
Version 1.1-2
Author Martin Oliver Sailer
Depends AlgDesign, gtools
Description Contains functions for the construction of carryover balanced crossover designs. In addition contains functions to check given designs for balance.
Maintainer Bjoern Bornkamp <bbnkmp@mail.de>
License GPL-2
NeedsCompilation no
Repository CRAN
Date/Publication 2022-04-04 21:50:02 UTC

R topics documented:
crossdes-package .................................................. 2
allcombs .......................................................... 3
balmin.RMD ....................................................... 4
biertest.d4 ......................................................... 5
biertest.dat ......................................................... 6
biertest.f2 ........................................................ 6
des.MOLS ........................................................ 7
find.BIB ........................................................... 8
get.plan ........................................................... 9
isCbalanced ....................................................... 11
isGYD ............................................................. 12
MOLS .............................................................. 13
williams .......................................................... 15
williams.BIB ...................................................... 16

Index 18
crossdes-package  

*Design and Analysis of dose-finding studies*

### Description

The crossdes package provides functions for the construction of carryover balanced crossover designs. In addition contains functions to check given designs for balance.

### Details

- **Package:** crossdes
- **Type:** Package
- **Version:** 1.1-1
- **Date:** 2013-03-18
- **License:** GPL-2

The main functions are:

- `get.plan`: Menu-driven selection of crossover designs
- `allcombs`: Construct crossover design with all possible treatment orders
- `williams`: Construct a Williams design
- `williams.BIB`: Construct a crossover design based on a combination of balanced incomplete block designs and Williams designs.
- `des.MOLS`: Construct crossover design based on mutually orthogonal Latin Squares
- `balmin.RMD`: Construct balanced minimal repeated measurements crossover design
- `isCbalanced`: Check whether a crossover design is balanced for first order carryover effects
- `isGYD`: Check whether a crossover design is balanced

### Author(s)

Martin Oliver Sailer

### References

- Earlier version of the package:

- Overview on cross-over designs:

- Wakeling, I.N. and MacFie, H.J.H. (1995): Designing consumer trials balanced for first and higher orders of carry-over effect when only a subset of k samples from t may be tested. *Food Quality and Preference* 6, 299-308.
allcombs

Balanced Row-Column Design with all Possible Treatment Orders

Description

The function constructs a row-column design with subjects as rows and periods as columns. Each subject gets each treatment at most once. All possible treatment orders are assigned to the subjects.

Usage

allcombs(trt, k)

Arguments

trt

An integer > 1. Number of treatments (products) to be tested.

k

An integer \( \leq \text{trt} \). Number of periods for each subject.

Details

The design is a carryover balanced generalized Youden design that is uniform on the columns. The treatments are numbered 1, \( \ldots, \text{trt} \). The entry \((i, j)\) of the design corresponds to the treatment the \(i\)-th subject gets in the \(j\)-th period.

Value

A matrix with \( \frac{\text{trt}!}{(\text{trt} - k)!} \) rows and \( k \) columns representing the experimental design.

Note

Requires the package gtools.

Author(s)

Oliver Sailer

References


Wakeling, I.N. and MacFie, H.J.H. (1995): Designing consumer trials balanced for first and higher orders of carry-over effect when only a subset of \( k \) samples from \( t \) may be tested. Food Quality and Preference 6, 299-308.

See Also

get.plan
Examples

# Design for 4 treatments assigned in 3 periods.
# All possible treatment orders occur.
allcombs(4, 3)

Description

The function constructs a row-column design with subjects as rows and periods as columns. The design is incomplete, i.e. no subject gets all the treatments. The design is balanced for carryover effects but will in general not be a balanced block design.

Usage

balmin.RMD(trt, n, p)

Arguments

trt  An integer >1 giving the number of treatments (products) to be tested.
n  An integer >1 giving the number of subjects (assessors) in the study.
p  An integer >1 giving the number of periods for each subject.

Details

A necessary and sufficient condition for the existence of such a design is that \( \frac{(trt-1)}{(p-1)} \) be a positive integer. In this case \( n = \frac{trt(trt-1)}{(p-1)} \). In the resulting design the treatments are numbered 1, \ldots, trt. The entry \((i, j)\) of the design corresponds to the treatment the \(i\)-th subject gets in the \(j\)-th period.

Value

A matrix with \(n\) rows and \(p\) columns representing the experimental design.

Author(s)

Oliver Sailer

References

Wakeling, I.N. and MacFie, H.J.H. (1995): Designing consumer trials balanced for first and higher orders of carry-over effect when only a subset of \(k\) samples from \(t\) may be tested. Food Quality and Preference 6, 299-308.
biertest.d4

See Also

get.plan

Examples

balmin.RMD(10,30,4) # a balanced minimal RMD
balmin.RMD(11,22,6) # another balanced minimal RMD

biertest.d4


Description

The design is a balanced incomplete block design with rows as blocks. If one assumes that there is a preperiod, i.e. that each assessor is given the treatment of the fifth period before the experiment starts, then the design is carryover balanced.

Usage

data(biertest.d4)

Format

A matrix with 12 rows corresponding to the assessors, 5 columns corresponding to the five periods.

Details

The five treatments (products) to be tested are numbered 1,...,5.

Source


Examples

data(biertest.d4)
**biertest.dat**  
*The Beer Testing Data in Kunert (1998)*

**Description**

The data comes from a beer testing experiment where assessors had to rate the bitterness of five beers.

**Usage**

```r
data(biertest.dat)
```

**Format**

A matrix with 12 rows corresponding to the assessors, 5 columns corresponding to the five periods.

**Details**

The possible values for the ratings lie in the interval [0, 12]. Note that the data were reordered after the experiment.

**Source**


**Examples**

```r
data(biertest.dat)
```

---

**biertest.f2**  
*Experimental Design for the Beer Testing Data in Kunert (1998)*

**Description**

The design is a row-column design. It is actually a generalized latin square design that is also carryover balanced.

**Usage**

```r
data(biertest.f2)
```

**Format**

A matrix with 10 rows corresponding to the first 10 assessors in a fictional uniformity trial, 5 columns corresponding to the five periods.
Details

There are five products to be tested, numbered 1, …, 5.

Source


Examples

data(biertest.f2)

des.MOLS

Description

The function constructs row-column designs based on complete sets of mutually orthogonal latin squares. Each subject may get each treatment at most once. The design is a generalized Youden design that is also balanced for carryover effects.

Usage

des.MOLS(trt, k = trt)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>trt</td>
<td>A prime power less than 100. The number of treatments (products) to be tested.</td>
</tr>
<tr>
<td>k</td>
<td>An integer ≤ trt. Number of periods for each subject.</td>
</tr>
</tbody>
</table>

Details

A complete set of mutually orthogonal latin squares is constructed using Galois Fields. The rows of the designs represent the treatment orders for the subjects. If an incomplete design with k columns is needed, only the first k columns of the designs are considered. The treatments are numbered 1, …, trt. The entry (i, j) of the design corresponds to the treatment the i-th subject gets in the j-th period.

Value

A matrix with trt(trt − 1) rows and k columns representing the experimental design.

Author(s)

Oliver Sailer
References

Wakeling, I.N. and MacFie, H.J.H. (1995): Designing consumer trials balanced for first and higher orders of carry-over effect when only a subset of k samples from t may be tested. Food Quality and Preference 6, 299-308.


See Also

get.plan, MOLS

Examples

des.MOLS(7,7)
des.MOLS(8,5)

find.BIB

Generation of Balanced Incomplete Block Designs Using the Package AlgDesign

Description

The function optBlock of the library AlgDesign is used to search for balanced incomplete block designs (BIBDs). The design is assigned to a matrix where rows represent blocks (subjects) and columns represent periods.

Usage

find.BIB(trt, b, k, iter = 30)

Arguments

trt An integer > 1 giving the number of treatments of the design.
b An integer > 1 giving the number of rows (subjects) of the design.
k An integer > 1 giving the number of columns (periods) of the design.
iter The number of iterations of the function optBlock

Details

The function optBlock tries to find a D-optimal block design for the specified parameters. The resulting design need not be a BIBD. The necessary conditions for the existence are that \( \frac{bk}{trt} \) and \( \frac{bk(k-1)}{trt(trt-1)} \) positive integers. They are NOT checked automatically. Even if they are fulfilled, there need not be a BIBD. If no BIBD is found, the function is iterated. If no BIBD is found after iter iterations, the search is terminated. The resulting design should be checked by the user applying isGYD.
Value

A matrix that represents the experimental design.

Note

As indicated above, the returned design is not necessarily a BIBD design.

Author(s)

Oliver Sailer

References


See Also

get.plan, optBlock

Examples

find.BIB(10,30,4)  # BIBD
find.BIB(3,3,3)    # BIBD
find.BIB(5,5,3, 100)  # There is no BIBD for these parameters
# isGYD(find.BIB(5,5,3, 100))

Description

This menu based function constructs simple experimental designs for repeated measurements with one or two block variables. It is assumed that each subject is assigned to each treatment at most once. A maximum number of subjects in the study is also requested. Five possible construction methods available. These construction methods and the characteristics of the resulting designs are described in Wakeling and MacFie (1995). See also Jones and Kenward (1989), Ch. 5, for a discussion of these designs. The function is demonstrated in more detail in Sailer (2005).

Usage

get.plan(trt, k = trt, maxsub = 1000)

Arguments

trt            An integer > 1, giving the number of treatments.
k            An integer in \{2, ..., trt\} giving the number of periods.
maxsub       The maximum number of subjects available.
Details

The five types of designs are: designs based on all possible treatment orders ("all.combin"), Williams designs ("williams"), designs based on mutually orthogonal latin squares ("des.MOLS"), a combination of balanced incomplete block designs (BIBDs) and Williams designs ("williams.BIB") by Patterson (1951) and the balanced minimal designs of Afsarinejad ("balmin.RMD"). Some designs are only available for special combinations of treatment number and number of periods. Other designs may require too many subjects. Therefore, the possible choices available for the submitted values of trt, k and maxsub are determined. If there is no design available, the parameters may be changed interactively. If more than one design type is available the user has to choose one. The minimum number of subjects required for the designs is given and maybe a criterion for selecting a design. All types of designs are balanced for first-order carryover effects. All types except the balanced minimal RMDs are also balanced block designs. The user may want to construct a design for a multiple of the minimum number of subjects required to get closer to the preferred number of subjects. Once the design is chosen the design is displayed. In practice the labels for the treatments and subjects should be randomized before the design is used. The treatments are numbered 1,..,trt. The entry (i, j) of the design corresponds to the treatment the i-th subject gets in the j-th period.

Value

A matrix representing the experimental design.

Warning

For the construction of designs that combine BIBDs with Williams designs, the function find.BIB is called to search for a BIBD. If the necessary conditions for the existence of a BIBD are fulfilled, this approach always returns a design. This design will however not always be a BIBD. When using the Patterson approach, please check the resulting design for balance using isGYD and isCbalanced.

It should be noted that this is a computational problem only, not a problem of the theoretical approach of Patterson (1951).

Note

The "All combinations" approach requires the package gtools.

Author(s)

Oliver Sailer

References

isCbalanced  

Checking Block Designs for Carryover Balance

Description

The function checks whether a block design is balanced for first order carryover effects (residual effects). The user specifies whether there is a preperiod. The design is checked and the left neighbour incidence matrix is given.

Usage

isCbalanced(d, preperiod = FALSE)

Arguments

d
A matrix with entries 1,...,trt representing the experimental design with rows as blocks (subjects). The columns represent periods.

preperiod
Logical flag. TRUE if there is a preperiod. In this case, each subject experiences in the first period the residual effect of the treatment of the last period (i.e. the last period precedes the first period, i.e. the plots in the last period are left neighbours of the plots in the first period). FALSE if there are no residual effects in the first period.

See Also

allcombs, balmin.RMD, des.MOLS, williams, williams.BIB

Examples

## Not run:
get.plan(10,4,60)
  # "williams.BIB" or "balmin.RMD"
get.plan(7,7,7000)
  # "all.combin", "williams" or "des.MOLS", "williams" requires
  # only 14 subjects, "all.combin" requires 5040.
get.plan(5,5,5)
  # Increase maxsub

## End(Not run)
Details

The design is said to be carryover balanced (balanced for first order carryover effects), if each treatment is preceeded by all other treatments equally often and if no treatment is preceeded by itself. If the design is balanced, this is stated.

Value

1 Logical flag. TRUE if the design is carryover balanced. This is not displayed on the screen.

2 Left neighbour incidence matrix. The \((i, j)\)-th element is the number of times that treatment \(i\) preceeds treatment \(j\).

Author(s)

Oliver Sailer

See Also

isGYD

Examples

d1 <- matrix( c(1,2,3,4,1,1,1,1), 4,2)
d2 <- matrix( c(1:4,2:4,1,4,1:3,3,4,1),ncol=4)
d3 <- matrix( rep(1:3,each=2), ncol=2)
isCbalanced(d1)
isCbalanced(d1,TRUE)
isCbalanced(d2)
isCbalanced(d3,TRUE)

Description

A function to check a simple block or a row-column design for balance. The rows and columns of the design are blocking variables. It is checked which type of balance the design fulfills. Optionally, incidence and concurrence matrices are given.

Usage

isGYD(d, tables=FALSE, type=TRUE)

Arguments

d A matrix representing the experimental design. The treatments must be numbered 1,...,\(drt\).
tables Logical flag. If TRUE, incidence matrices are displayed.
type Logical flag. If TRUE, the type of design is displayed.
Details

A design is said to be a balanced block design if the following three conditions hold: i) Each treatment appears equally often in the design. ii) The design is binary in the sense that each treatment appears in each block either \( n \) or \( n+1 \) times where \( n \) is an integer. iii) The number of concurrences of treatments \( i \) and \( j \) is the same for all pairs of distinct treatments \((i, j)\). Here the blocks are either rows or columns.

A design that has less columns (rows) than treatments is said to be incomplete with respect to rows (columns). A design that is balanced with respect to both rows and columns is called a generalized Youden design (GYD). A GYD for which each treatment occurs equally often in each row (column) is called uniform on the rows (columns). If both conditions hold, it is called a generalized latin square. A design where each treatment occurs exactly once in each row and column is called a latin square.

Value

A list containing information about balance in rows and columns as well as incidence and concurrence matrices for the design.

Author(s)

Oliver Sailer

See Also

isCbalanced

Examples

d1 <- matrix( c(1,2,3,4,1,1,1,1), 4,2)
# d1 is not balanced
d2 <- matrix( c(1:4,2:4,1,4,1:3,3,4,1),ncol=4)
# d2 is a latin square
d3 <- matrix( rep(1:3,each=2), ncol=2)
# d3 is a balanced incomplete block design.
d1
isGYD(d1,tables=TRUE)
d2
isGYD(d2,tables=TRUE)
d3
isGYD(d3,tables=TRUE)

MOLS

Construction of Complete Sets of Mutually Orthogonal Latin Squares

Description

The function constructs sets of mutually orthogonal latin squares (MOLS) using Galois fields. The construction works for prime powers only.
Usage

MOLS(p, n, primpol = GF(p, n)[][2][][1, ])

Arguments

- p: A prime number less than 100.
- n: A positive integer.
- primpol: A primitive polynomial of the Galois Field GF(p^n).

Details

If \( t = p^n \) is a prime power, then \( t - 1 \) latin squares of order \( t \) are constructed. The elements of the squares are numbered 1, \ldots, \( t \). These squares are mutually orthogonal, i.e. if any two of them are superimposed, the resulting array will contain each ordered pair \((i, j)\), \(i, j \in \{1, \ldots, t\}\) exactly once. The squares are in standard order, i.e. the first row is always equal to \((1, \ldots, t)\). A primitive polynomial may be constructed automatically using the internal function GF.

Value

For \( t = p^n \), an array that contains \( t - 1 \) latin squares is returned.

Author(s)

Oliver Sailer

References


See Also

des.MOLS

Examples

MOLS(7,1) # 6 mutually orthogonal latin squares of order 7
MOLS(2,3) # 7 mutually orthogonal latin squares of order 8
williams

Construction of Williams Designs

Description
The function constructs williams designs. Williams designs are row-column designs. They are used if each of the treatments in the study is given to each of the subjects. If the number of treatments to be tested is even, the design is a latin square, otherwise it consists of two latin squares.

Usage
williams(trt)

Arguments
trt An integer > 1, giving the number of treatments in the design.

Details
The resulting design is a (generalized) latin square that is also balanced for first order carryover effects. Carryover balance is achieved with very few subjects. In the experimental design the treatments are numbered 1,...,trt. The entry (i,j) of the design corresponds to the treatment the i-th subject gets in the j-th period.

Value
A matrix representing the experimental design.

Author(s)
Oliver Sailer

References
Wakeling, I.N. and MacFie, H.J.H. (1995): Designing consumer trials balanced for first and higher orders of carry-over effect when only a subset of k samples from t may be tested. Food Quality and Preference 6, 299-308.

See Also
get.plan

Examples
williams(3)
williams(10)
williams.BIB

Construction of Carryover Balanced Designs Based on Balanced Incomplete Block Designs

Description

Patterson (1951) combined balanced incomplete block designs (BIBDs) with Williams designs to get carryover balanced generalized Youden designs.

Usage

williams.BIB(d)

Arguments

d  A matrix representing a BIBD. Rows represent blocks (subjects).

Details

For each row of the design, a Williams design is constructed using the treatments of that row. The rows of the resulting designs are then combined. The treatments are numbered 1, ..., trt. The entry \((i, j)\) of the design corresponds to the treatment the \(i\)-th subject gets in the \(j\)-th period.

Value

A matrix representing the experimental design.

Warning

The resulting design is only balanced properly if the input design actually is a BIBD. This is NOT checked automatically. You have to do this by yourself, e.g. by applying isGYD to your design.

Note

BIBDs may be generated using find.BIB.

Author(s)

Oliver Sailer

References

Wakeling, I.N. and MacFie, H.J.H. (1995): Designing consumer trials balanced for first and higher orders of carry-over effect when only a subset of \(k\) samples from \(t\) may be tested. Food Quality and Preference 6, 299-308.
williams.BIB

See Also

get.plan, isGYD, find.BIB, williams

Examples

d <- matrix( rep(1:3,each=2), ncol=2)
# # check for balance
# isGYD(d)
williams.BIB(d)
Index

* datasets
  biertest.d4, 5
  biertest.dat, 6
  biertest.f2, 6

* design
  allcombs, 3
  balmin.RMD, 4
  des.MOLS, 7
  find.BIB, 8
  get.plan, 9
  isCbalanced, 11
  isGYD, 12
  MOLS, 13
  williams, 15
  williams.BIB, 16

* package
  crossdes-package, 2
    allcombs, 2, 3, 11
    balmin.RMD, 2, 4, 11
    biertest.d4, 5
    biertest.dat, 6
    biertest.f2, 6
    crossdes (crossdes-package), 2
    crossdes-package, 2
    des.MOLS, 2, 7, 11, 14
    find.BIB, 8, 17
    get.plan, 2, 3, 5, 8, 9, 9, 15, 17
    isCbalanced, 2, 11, 13
    isGYD, 2, 12, 12, 17
    MOLS, 8, 13
    optBlock, 9
    williams, 2, 11, 15, 17
    williams.BIB, 2, 11, 16