Package ‘blocksdesign’

September 30, 2014

Type Package

Title Nested Block Designs for Unstructured Treatments

Version 1.0

Date 2014-09-16

Author R. N. Edmondson

Maintainer Rodney Edmondson <rodney.edmondson@gmail.com>

Description Nested block designs for unstructured treatment sets where blocks can be repeatedly nested and treatments can have different levels of replication. Blocks strata are optimized hierarchically with each set of nested blocks optimized within the levels of the preceding set. Block sizes are equal if the number of blocks exactly divides the number of plots,otherwise they differ by at most one plot. The design output is a data table giving a randomised allocation of treatments to blocks together with a plan table showing treatments in blocks and a set of blocks-by-treatments incidence matrices, one for each blocks stratum.

License GPL(>= 2)

URL http://www.expdesigns.co.uk

Suggests knitr

VignetteBuilder knitr

Imports crossdes

R topics documented:

blocksdesign-package *Blocks design package*

Description

The blocksdesign package provides functionality for the construction of block designs for unstructured treatment sets with arbitrary levels of replication and arbitrary depth of nesting.

Details

Block designs group experimental units into homogeneous blocks to provide maximum precision for treatment comparisons. The most basic type of block design are complete randomised blocks where each block contains one or more complete sets of treatments. Complete randomized blocks are excellent for small designs but for larger designs, the variability within blocks may become too large for reliable treatment comparison and then it may be desirable to sub-divide the complete blocks into smaller incomplete blocks.

Traditionally, nested block designs in large experiments have used a single stratum of nested blocks contained within a set of complete main blocks. The complexity of design and analysis of designs with more than a single stratum of nesting has made multi-stratum nesting infeasible for practical experiments. However, modern software such as the lme4 mixed model package (Bates et al 2014) and the availability of modern design algorithms have largely eliminated these restrictions and multi-stratum nesting for large block designs is now entirely feasible.

The advantage of multi-stratum nesting is that random variability can be captured across a range of block sizes and this allows for more realistic modelling of block effects compared with single stratum nesting. The blocksdesign package is a general purpose tool that provides for the construction of general block designs where treatments can have any number of levels of replication and blocks can be nested to any feasible depth of nesting. Where designs have one or more levels of nesting, blocks are optimized hierarchically with each successive set of nested blocks optimized within the blocks of the preceding set.

The main function is blocks which is used to generate the actual required design. The output from blocks includes a data frame of the block and treatment factors for each plot, a data frame of the allocation of treatments to plots for each block in the design, blocks-by-treatments incidence matrices for each stratum in the design and an A-efficiency factor for each stratum in the design, together with an efficiency upper bound, where available.

The secondary function efficiencies takes the design output from the blocks function and uses it to construct tables of efficiency factors for each pairwise treatment difference in each stratum, as required.

The subsidiary function upper_bounds estimates A-efficiency upper bounds for regular block designs with equally replicated treatments and equal block sizes.

Further discussion of multi-stratum nesting can be found in the package vignette at: vignette("blocksdesign")

References

Bates, D., Maechler, M., Bolker, B. and Walker, S. (2014). lme4: Linear mixed-effects models using Eigen and S4. R package version 1.1-6. http://CRAN.R-project.org/package=lme4

`blocks` *Block designs*

Description

`blocks` function constructs nested blocks for unstructured treatments with arbitrary levels of replication and arbitrary depth of nesting

Usage

```
blocks(treatments, replicates, blocklevels = hcf, searches = min(64,
  floor(4096/nunits)), seed = NULL)
```

Arguments

`treatments` A list of the number of treatments for each treatment set in the design. Each treatment number must have a matching replication number in the `replicates` list

`replicates` A list of the number of replicates for each treatment set in the design. Each replication number must have a matching treatment number in the `treatments` list

`blocklevels` A hierarchical list of nested blocks where the first number is the number of main blocks and the remaining numbers, if any, are the numbers of blocks nested in each preceding block. The default is the hcf of the replication numbers.

`searches` The number of local optima searched during a design optimization. The default is the minimum of 64 or the integer quotient of 4096 divided by the number of plots.

`seed` An integer seed for initializing the random number generator where a design must be reproducible. The default is a random seed.

Details

`blocks` constructs nested block designs for unstructured treatment sets where treatments can have any arbitrary replication, not necessarily all equal, and blocks can have any feasible depth of nesting.

Treatment and replication numbers are defined by the `treatments` and `replicates` parameter lists. These lists must be of equal length and each matching pair of numbers in the two lists represents a treatment set where the `treatments` list gives the number of treatments in the set and the `replicates` list gives the replication of the set.

Any number of treatment sets is allowed and the treatments are numbered consecutively according to the ordering of the treatment sets in the parameter lists (see the examples).

Blocks are defined by the `blocklevels` list which is a hierarchical list of nested blocks. The first number is the number of main blocks and the succesive numbers, if any, are the numbers of blocks nested in each preceding block. The cumulative product of the levels for any stratum is the total number of blocks in that stratum. The default value for the `blocklevels` list is a single number equal to the highest common factor (hcf) of the replication numbers, which gives an orthogonal blocks design with the maximum possible number of othogonal blocks.

Block sizes in any given stratum will be equal if the cumulative number of blocks exactly divides the number of plots otherwise they will be as near equal as possible and will never differ by more than a single unit.

Lattice designs where the number of treatments is the square of the block size v and the number of replicates is k+2 or less and k mutually orthogonal latin squares (MOLS) of size v*v exist are constructed algebraically. Lattice designs exist for any v if k = 1 and for any prime or prime power v if k < v and for v = 10 and k = 2. Prime-power MOLS are constructed by using the MOLS function of the `crossdes` package (Sailer 2013).

All other designs are constructed algorithmically by a swapping algorithm that maximizes the determinant of the information matrix (D-optimality).

Designs are optimized hierarchically with the blocks of each new set optimized within the blocks of the preceding set.

Designs are fully randomized with treatments randomized within blocks and each set of nested blocks randomized within the preceding set of blocks.

Value

`Design` Data frame showing the block and treatment factors for each plot

`Plan` Data frame showing the allocation of treatments to plots for each block in the design

`Incidences` List of blocks-by-treatments incidence matrices, one for each stratum in the design

`Efficiencies` Data frame showing the A-efficiency factor for each stratum in the design together with an upper bound, where available

`seed` Numerical seed for random number generator

References

Sailer, M. O. (2013). crossdes: Construction of Crossover Designs. R package version 1.1-1. http://CRAN.R-project.org/package=crossdes

Examples

```
# 3 treatments with 2 reps, 2 treatments with 4 reps, 4 treatments with 3 reps
# the replication hcf is 1 and the default design is a completely randomized design
blocks(treatments=c(3,2,4),replicates=c(2,4,3))

# 50 treatments with 4 reps in 4 complete randomized blocks
blocks(treatments=50,replicates=4)

# as above but with 4 main blocks and 5 nested blocks in each main block
blocks(treatments=50,replicates=4,blocklevels=c(4,5))

# as above but with 20 additional single replicate treatments, one to each block
blocks(treatments=c(50,20),replicates=c(4,1),blocklevels=c(4,5))
```

```
# 64 treatments with 2 reps and 2 main blocks with five 2-level nested factors
blocks(treatments=64,replicates=2,blocklevels=c(2,2,2,2,2,2),searches=12)

# concurrence matrices of 36 treatments with 3 reps and 3 main blocks with 6 nested blocks
crossprod(blocks(treatments=36,replicates=3,blocklevels=c(3,6))$Incidences[[2]])

# concurrence matrix for 13 treatments with 4 reps and 13 treatments with one rep in 13 blocks
crossprod(blocks(c(13,13),c(4,1),13,searches=100)$Incidences[[1]])
```

efficiencies *Efficiencies of pairwise differences*

Description

efficiencies function tabulates the efficiencies of pairwise treatment difference for each stratum of a design

Usage

```
efficiencies(Design)
```

Arguments

Design	A design data frame generated by blocks

Details

Tabulates the efficiency of pairwise treatment difference for each statum of any design built by the blocks function. The tables provide detailed information such as the minimum efficiency of all possible treatment comparisons. For equi-replicate designs, the harmonic mean of the pairwise efficiency factors will equal the A-efficiency factor, as calculated by the blocks function.

Value

Efficiencies	List of treatments efficiency matrices for all pairwise differences for each stratum of the design

Examples

```
# 4 replicates of 50 treatments in complete randomized blocks
efficiencies(blocks(treatments=50,replicates=4,blocklevels=c(4,5))$Design)
```

upper_bounds *Efficiency bounds*

Description

upper_bounds finds upper A-efficiency bounds for regular block designs.

Usage

```
upper_bounds(nplots, ntrts, nblocks)
```

Arguments

nplots	The total number of plots in the design
ntrts	The total number of treatments in the design
nblocks	The total number of blocks in the design

Details

Upper bounds for the A-efficiency of regular nested block designs (see Chapter 2.8 of John and Williams 1995). Non-trivial bounds are calculated for regular block designs with equal block sizes and equal replication. All other designs return NA.

References

John, J. A. and Williams, E. R. (1995). Cyclic and Computer Generated Designs. Chapman and Hall, London.

Examples

```
# 50 plots, 10 treatments and 10 blocks for a design with 5 replicates and blocks of size 5
upper_bounds(nplots=50,ntrts=10,nblocks=10)
```

Index