Experimental Designs with **agricolae**

Felipe de Mendiburu[1], Muhammad Yaseen[2]

2020-05-01

Contents

1. Professor of the Academic Department of Statistics and Informatics of the Faculty of Economics and Planning.National University Agraria La Molina-PERU.

2. Department of Mathematics and Statistics, University of Agriculture Faisalabad, Pakistan.

---

1 Experimental Designs

The package **agricolae** presents special functions for the creation of the field book for experimental designs. Due to the random generation, this package is quite used in agricultural research.

For this generation, certain parameters are required, as for example the name of each treatment, the number of repetitions, and others, according to the design (Le Clerg et al., 1962; Cochran and Cox, 1992; Kuehl, 2000; Montgomery, 2002). There are other parameters of random generation, as the seed to reproduce the same random generation or the generation method (See the reference manual of **agricolae**).

Important parameters in the generation of design:

series:	A constant that is used to set numerical tag blocks , eg number = 2, the labels will be : 101, 102, for the first row or block, 201, 202, for the following , in the case of completely randomized design, the numbering is sequencial.
design:	Some features of the design requested agricolae be applied specifically to design.ab(factorial) or design.split (split plot) and their possible values are: "rcbd", "crd" and "lsd".
seed:	The seed for the random generation and its value is any real value, if the value is zero, it has no reproducible generation, in this case copy of value of the outdesign$parameters.
kinds:	the random generation method, by default "Super-Duper".
first:	For some designs is not required random the first repetition, especially in the block design, if you want to switch to random, change to TRUE.
randomization:	TRUE or FALSE. If false, randomization is not performed

Output design:

parameters:	the input to generation design, include the seed to generation random, if seed=0, the program generate one value and it is possible reproduce the design.
book:	field book
statistics:	the information statistics the design for example efficiency index, number of treatments.
sketch:	distribution of treatments in the field.
The enumeration of the plots	zigzag is a function that allows you to place the numbering of the plots in the direction of serpentine: The zigzag is output generated by one design: blocks, Latin square, graeco, split plot, strip plot, into blocks factorial, balanced incomplete block, cyclic lattice, alpha and augmented blocks.
fieldbook:	output zigzag, contain field book.

1.1 Completely Randomized Design (CRD)

It generates completely a randomized design with equal or different repetition. "Random" uses the methods of number generation in **R**.The seed is by `set.seed(seed, kinds)`. They only require the names of the treatments and the number of their repetitions and its parameters are:

```
str(design.crd)
```

```
function (trt, r, serie = 2, seed = 0, kinds = "Super-Duper", randomization = TRUE)
 - attr(*, "srcref")= 'srcref' int [1:8] 47 1 80 1 1 1 47 80
  ..- attr(*, "srcfile")=Classes 'srcfilecopy', 'srcfile' <environment: 0x557496b753e8>
```

```
trt <- c("A", "B", "C")
repeticion <- c(4, 3, 4)
outdesign <- design.crd(trt,r=repeticion,seed=777,serie=0)
book1 <- outdesign$book
head(book1)
```

```
  plots r trt
```

```
1     1 1   C
2     2 1   A
3     3 1   B
4     4 2   A
5     5 3   A
6     6 2   C
```

Excel:write.csv(book1,"book1.csv",row.names=FALSE)

1.2 Randomized Complete Block Design (RCBD)

It generates field book and sketch to Randomized Complete Block Design. "Random" uses the methods of number generation in **R**.The seed is by `set.seed(seed, kinds)`. They require the names of the treatments and the number of blocks and its parameters are:

```
str(design.rcbd)
```

```
function (trt, r, serie = 2, seed = 0, kinds = "Super-Duper", first = TRUE,
    continue = FALSE, randomization = TRUE)
 - attr(*, "srcref")= 'srcref' int [1:8] 51 1 83 1 1 1 51 83
  ..- attr(*, "srcfile")=Classes 'srcfilecopy', 'srcfile' <environment: 0x5574969321d8>
```

```
trt <- c("A", "B", "C","D","E")
repeticion <- 4
outdesign <- design.rcbd(trt,r=repeticion, seed=-513, serie=2)
# book2 <- outdesign$book
book2<- zigzag(outdesign) # zigzag numeration
print(outdesign$sketch)
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,] "E"  "B"  "D"  "A"  "C"
[2,] "B"  "A"  "D"  "C"  "E"
[3,] "C"  "E"  "A"  "B"  "D"
[4,] "D"  "C"  "E"  "B"  "A"
```

```
print(matrix(book2[,1],byrow = TRUE, ncol = 5))
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]  101  102  103  104  105
[2,]  205  204  203  202  201
[3,]  301  302  303  304  305
[4,]  405  404  403  402  401
```

1.3 Latin Square Design

It generates Latin Square Design. "Random" uses the methods of number generation in **R**.The seed is by `set.seed(seed, kinds)`. They require the names of the treatments and its parameters are:

```
str(design.lsd)
```

```
function (trt, serie = 2, seed = 0, kinds = "Super-Duper", first = TRUE,
    randomization = TRUE)
 - attr(*, "srcref")= 'srcref' int [1:8] 44 1 84 1 1 1 44 84
  ..- attr(*, "srcfile")=Classes 'srcfilecopy', 'srcfile' <environment: 0x557496bf3e30>
```

```
trt <- c("A", "B", "C", "D")
outdesign <- design.lsd(trt, seed=543, serie=2)
print(outdesign$sketch)
```

```
     [,1] [,2] [,3] [,4]
[1,] "B"  "C"  "A"  "D"
[2,] "D"  "A"  "C"  "B"
[3,] "C"  "D"  "B"  "A"
[4,] "A"  "B"  "D"  "C"
```

1.3.1 Serpentine enumeration

```
book <- zigzag(outdesign)
print(matrix(book[,1],byrow = TRUE, ncol = 4))
```

```
     [,1] [,2] [,3] [,4]
[1,]  101  102  103  104
[2,]  204  203  202  201
[3,]  301  302  303  304
[4,]  404  403  402  401
```

1.4 Graeco-Latin Designs

A graeco-latin square is a $k \times k$ pattern that permits the study of k treatments simultaneously with three different blocking variables, each at k levels. The function is only for squares of the odd numbers and even numbers (4, 8, 10 and 12). They require the names of the treatments of each factor of study and its parameters are:

```
str(design.graeco)
```

```
function (trt1, trt2, serie = 2, seed = 0, kinds = "Super-Duper",
    randomization = TRUE)
 - attr(*, "srcref")= 'srcref' int [1:8] 53 1 145 1 1 1 53 145
  ..- attr(*, "srcfile")=Classes 'srcfilecopy', 'srcfile' <environment: 0x557495dc21a8>
```

```
trt1 <- c("A", "B", "C", "D")
trt2 <- 1:4
outdesign <- design.graeco(trt1,trt2, seed=543, serie=2)
print(outdesign$sketch)
```

```
     [,1]   [,2]   [,3]   [,4]
[1,] "NA 2" "NA 4" "NA 3" "NA 1"
[2,] "NA 3" "NA 1" "NA 2" "NA 4"
[3,] "NA 1" "NA 3" "NA 4" "NA 2"
[4,] "NA 4" "NA 2" "NA 1" "NA 3"
```

1.4.1 Serpentine enumeration

```
book <- zigzag(outdesign)
print(matrix(book[,1],byrow = TRUE, ncol = 4))
```

```
     [,1] [,2] [,3] [,4]
[1,]  101  102  103  104
```

```
[2,]  204  203  202  201
[3,]  301  302  303  304
[4,]  404  403  402  401
```

1.5 Youden Square Design

Such designs are referred to as Youden squares since they were introduced by Youden (1937) after Yates (1936) considered the special case of column equal to number treatment minus 1. "Random" uses the methods of number generation in **R**. The seed is by `set.seed(seed, kinds)`. They require the names of the treatments of each factor of study and its parameters are:

```
str(design.youden)
```

```
function (trt, r, serie = 2, seed = 0, kinds = "Super-Duper", first = TRUE,
    randomization = TRUE)
 - attr(*, "srcref")= 'srcref' int [1:8] 50 1 96 1 1 1 50 96
  ..- attr(*, "srcfile")=Classes 'srcfilecopy', 'srcfile' <environment: 0x5574960d3128>
```

```
varieties<-c("perricholi","yungay","maria bonita","tomasa")
r<-3
outdesign <-design.youden(varieties,r,serie=2,seed=23)
print(outdesign$sketch)
```

```
     [,1]           [,2]           [,3]
[1,] "maria bonita" "tomasa"       "perricholi"
[2,] "yungay"       "maria bonita" "tomasa"
[3,] "perricholi"   "yungay"       "maria bonita"
[4,] "tomasa"       "perricholi"   "yungay"
```

```
book <- outdesign$book
print(book) # field book.
```

```
   plots row col    varieties
1    101   1   1 maria bonita
2    102   1   2       tomasa
3    103   1   3   perricholi
4    201   2   1       yungay
5    202   2   2 maria bonita
6    203   2   3       tomasa
7    301   3   1   perricholi
8    302   3   2       yungay
9    303   3   3 maria bonita
10   401   4   1       tomasa
11   402   4   2   perricholi
12   403   4   3       yungay
```

```
print(matrix(as.numeric(book[,1]),byrow = TRUE, ncol = r))
```

```
     [,1] [,2] [,3]
[1,]  101  102  103
[2,]  201  202  203
[3,]  301  302  303
[4,]  401  402  403
```

1.5.1 Serpentine enumeration

```
book <- zigzag(outdesign)
print(matrix(as.numeric(book[,1]),byrow = TRUE, ncol = r))
```

```
     [,1] [,2] [,3]
[1,]  101  102  103
[2,]  203  202  201
[3,]  301  302  303
[4,]  403  402  401
```

1.6 Balanced Incomplete Block Designs (BIBD)

Creates Randomized Balanced Incomplete Block Design. "Random" uses the methods of number generation in **R**. The seed is by `set.seed(seed, kinds)`. They require the names of the treatments and the size of the block and its parameters are:

```
str(design.bib)
```

```
function (trt, k, r = NULL, serie = 2, seed = 0, kinds = "Super-Duper",
    maxRep = 20, randomization = TRUE)
 - attr(*, "srcref")= 'srcref' int [1:8] 64 1 145 1 1 1 64 145
  ..- attr(*, "srcfile")=Classes 'srcfilecopy', 'srcfile' <environment: 0x5574946dc490>
```

```
trt <- c("A", "B", "C", "D", "E" )
k <- 4
outdesign <- design.bib(trt,k, seed=543, serie=2)
```

```
Parameters BIB
==============
Lambda     : 3
treatmeans : 5
Block size : 4
Blocks     : 5
Replication: 4

Efficiency factor 0.9375

<<< Book >>>
```

```
book5 <- outdesign$book
outdesign$statistics
```

```
       lambda treatmeans blockSize blocks r Efficiency
values      3          5         4      5 4     0.9375
```

```
outdesign$parameters
```

```
$design
[1] "bib"

$trt
[1] "A" "B" "C" "D" "E"
```

```
$k
[1] 4

$serie
[1] 2

$seed
[1] 543

$kinds
[1] "Super-Duper"
```

According to the produced information, they are five blocks of size 4, being the matrix:

```
outdesign$sketch
```

```
     [,1] [,2] [,3] [,4]
[1,] "B"  "C"  "E"  "A"
[2,] "C"  "D"  "B"  "A"
[3,] "A"  "D"  "E"  "B"
[4,] "E"  "C"  "D"  "B"
[5,] "D"  "C"  "E"  "A"
```

It can be observed that the treatments have four repetitions. The parameter lambda has three repetitions, which means that a couple of treatments are together on three occasions. For example, B and E are found in the blocks I, II and V.

1.6.1 Serpentine enumeration

```
book <- zigzag(outdesign)
matrix(book[,1],byrow = TRUE, ncol = 4)
```

```
     [,1] [,2] [,3] [,4]
[1,]  101  102  103  104
[2,]  204  203  202  201
[3,]  301  302  303  304
[4,]  404  403  402  401
[5,]  501  502  503  504
```

1.7 Cyclic Designs

They require the names of the treatments, the size of the block and the number of repetitions. This design is used for 6 to 30 treatments. The repetitions are a multiple of the size of the block; if they are six treatments and the size is 3, then the repetitions can be 6, 9, 12, etc. and its parameters are:

```
str(design.cyclic)
```

```
function (trt, k, r, serie = 2, rowcol = FALSE, seed = 0, kinds = "Super-Duper",
    randomization = TRUE)
 - attr(*, "srcref")= 'srcref' int [1:8] 51 1 198 1 1 1 51 198
  ..- attr(*, "srcfile")=Classes 'srcfilecopy', 'srcfile' <environment: 0x55749605eee0>
```

```
trt <- c("A", "B", "C", "D", "E", "F" )
outdesign <- design.cyclic(trt,k=3, r=6, seed=543, serie=2)
```

```
cyclic design
Generator block basic:
1 2 4
1 3 2

Parameters
==================
treatmeans : 6
Block size : 3
Replication: 6
```

```
book6 <- outdesign$book
outdesign$sketch[[1]]
```

```
     [,1] [,2] [,3]
[1,] "F"  "D"  "C"
[2,] "C"  "B"  "E"
[3,] "D"  "E"  "A"
[4,] "B"  "E"  "F"
[5,] "A"  "F"  "C"
[6,] "B"  "A"  "D"
```

```
outdesign$sketch[[2]]
```

```
     [,1] [,2] [,3]
[1,] "A"  "F"  "E"
[2,] "A"  "C"  "B"
[3,] "A"  "F"  "B"
[4,] "C"  "D"  "E"
[5,] "E"  "D"  "F"
[6,] "D"  "C"  "B"
```

12 blocks of 4 treatments each have been generated.

1.7.1 Serpentine enumeration

```
book <- zigzag(outdesign)
array(book$plots,c(3,6,2))->X
t(X[,,1])
```

```
     [,1] [,2] [,3]
[1,]  101  102  103
[2,]  106  105  104
[3,]  107  108  109
[4,]  112  111  110
[5,]  113  114  115
[6,]  118  117  116
```

```
t(X[,,2])
```

```
     [,1] [,2] [,3]
[1,]  201  202  203
[2,]  206  205  204
[3,]  207  208  209
```

```
[4,]  212  211  210
[5,]  213  214  215
[6,]  218  217  216
```

1.8 Lattice Designs

SIMPLE and **TRIPLE** lattice designs. It randomizes treatments in $k \times k$ lattice. They require a number of treatments of a perfect square; for example 9, 16, 25, 36, 49, etc. and its parameters are:

```
str(design.lattice)
```

```
function (trt, r = 3, serie = 2, seed = 0, kinds = "Super-Duper",
    randomization = TRUE)
 - attr(*, "srcref")= 'srcref' int [1:8] 41 1 117 1 1 1 41 117
  ..- attr(*, "srcfile")=Classes 'srcfilecopy', 'srcfile' <environment: 0x557496d70010>
```

They can generate a simple lattice (2 rep.) or a triple lattice (3 rep.) generating a triple lattice design for 9 treatments 3×3

```
trt<-letters[1:9]
outdesign <-design.lattice(trt, r = 3, serie = 2, seed = 33,
    kinds =  "Super-Duper")
```

```
Lattice design,  triple   3 x 3

Efficiency factor
(E ) 0.7272727

<<< Book >>>
```

```
book7 <- outdesign$book
outdesign$parameters
```

```
$design
[1] "lattice"

$type
[1] "triple"

$trt
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i"

$r
[1] 3

$serie
[1] 2

$seed
[1] 33

$kinds
[1] "Super-Duper"
```

```
outdesign$sketch
```

```
$rep1
     [,1] [,2] [,3]
[1,] "g"  "c"  "a"
[2,] "f"  "b"  "h"
[3,] "i"  "e"  "d"

$rep2
     [,1] [,2] [,3]
[1,] "g"  "f"  "i"
[2,] "a"  "h"  "d"
[3,] "c"  "b"  "e"

$rep3
     [,1] [,2] [,3]
[1,] "g"  "h"  "e"
[2,] "c"  "f"  "d"
[3,] "a"  "b"  "i"
```

```
head(book7)
```

```
  plots r block trt
1   101 1     1   g
2   102 1     1   c
3   103 1     1   a
4   104 1     2   f
5   105 1     2   b
6   106 1     2   h
```

1.8.1 Serpentine enumeration

```
book <- zigzag(outdesign)
array(book$plots,c(3,3,3)) -> X
t(X[,,1])
```

```
     [,1] [,2] [,3]
[1,]  101  102  103
[2,]  106  105  104
[3,]  107  108  109
```

```
t(X[,,2])
```

```
     [,1] [,2] [,3]
[1,]  201  202  203
[2,]  206  205  204
[3,]  207  208  209
```

```
t(X[,,3])
```

```
     [,1] [,2] [,3]
[1,]  301  302  303
[2,]  306  305  304
[3,]  307  308  309
```

1.9 Alpha Designs

Generates an alpha designs starting from the alpha design fixing under the series formulated by Patterson and Williams. These designs are generated by the alpha arrangements. They are similar to the lattice designs, but the tables are rectangular s by k (with s blocks and $k < s$ columns. The number of treatments should be equal to $s \times k$ and all the experimental units $r \times s \times k$ (r replications) and its parameters are:

```
str(design.alpha)
```

```
function (trt, k, r, serie = 2, seed = 0, kinds = "Super-Duper",
    randomization = TRUE)
 - attr(*, "srcref")= 'srcref' int [1:8] 71 1 213 1 1 1 71 213
  ..- attr(*, "srcfile")=Classes 'srcfilecopy', 'srcfile' <environment: 0x5574958be6c8>
```

```
trt <- letters[1:15]
outdesign <- design.alpha(trt,k=3,r=2,seed=543)
```

```
Alpha Design (0,1) - Serie  I

Parameters Alpha Design
=======================
Treatmeans : 15
Block size : 3
Blocks     : 5
Replication: 2

Efficiency factor
(E ) 0.6363636

<<< Book >>>
```

```
book8 <- outdesign$book
outdesign$statistics
```

```
       treatments blocks Efficiency
values         15      5  0.6363636
```

```
outdesign$sketch
```

```
$rep1
     [,1] [,2] [,3]
[1,] "i"  "g"  "m"
[2,] "f"  "o"  "h"
[3,] "n"  "j"  "b"
[4,] "a"  "c"  "k"
[5,] "e"  "l"  "d"

$rep2
     [,1] [,2] [,3]
[1,] "g"  "f"  "k"
[2,] "e"  "j"  "a"
[3,] "m"  "c"  "l"
[4,] "n"  "d"  "o"
```

```
[5,] "i"  "h"  "b"
```

```
# codification of the plots
A<-array(book8[,1], c(3,5,2))
t(A[,,1])
```

```
     [,1] [,2] [,3]
[1,]  101  102  103
[2,]  104  105  106
[3,]  107  108  109
[4,]  110  111  112
[5,]  113  114  115
```

```
t(A[,,2])
```

```
     [,1] [,2] [,3]
[1,]  201  202  203
[2,]  204  205  206
[3,]  207  208  209
[4,]  210  211  212
[5,]  213  214  215
```

1.9.1 Serpentine enumeration

```
book <- zigzag(outdesign)
A<-array(book[,1], c(3,5,2))
t(A[,,1])
```

```
     [,1] [,2] [,3]
[1,]  101  102  103
[2,]  106  105  104
[3,]  107  108  109
[4,]  112  111  110
[5,]  113  114  115
```

```
t(A[,,2])
```

```
     [,1] [,2] [,3]
[1,]  201  202  203
[2,]  206  205  204
[3,]  207  208  209
[4,]  212  211  210
[5,]  213  214  215
```

1.10 Augmented Block Designs

These are designs for two types of treatments: the control treatments (common) and the increased treatments. The common treatments are applied in complete randomized blocks, and the increased treatments, at random. Each treatment should be applied in any block once only. It is understood that the common treatments are of a greater interest; the standard error of the difference is much smaller than when between two increased ones in different blocks. The function `design.dau()` achieves this purpose and its parameters are:

```
str(design.dau)
```

```
function (trt1, trt2, r, serie = 2, seed = 0, kinds = "Super-Duper",
    name = "trt", randomization = TRUE)
 - attr(*, "srcref")= 'srcref' int [1:8] 57 1 96 1 1 1 57 96
  ..- attr(*, "srcfile")=Classes 'srcfilecopy', 'srcfile' <environment: 0x557494d21b10>
```

```
rm(list=ls())
trt1 <- c("A", "B", "C", "D")
trt2 <- c("t","u","v","w","x","y","z")
outdesign <- design.dau(trt1, trt2, r=5, seed=543, serie=2)
book9 <- outdesign$book
with(book9,by(trt, block,as.character))
```

```
block: 1
[1] "C" "B" "v" "D" "t" "A"
------------------------------------------------------------
block: 2
[1] "D" "u" "A" "B" "x" "C"
------------------------------------------------------------
block: 3
[1] "B" "y" "C" "A" "D"
------------------------------------------------------------
block: 4
[1] "A" "B" "C" "D" "w"
------------------------------------------------------------
block: 5
[1] "z" "A" "C" "D" "B"
```

1.10.1 Serpentine enumeration

```
book <- zigzag(outdesign)
with(book,by(plots, block, as.character))
```

```
block: 1
[1] "101" "102" "103" "104" "105" "106"
------------------------------------------------------------
block: 2
[1] "206" "205" "204" "203" "202" "201"
------------------------------------------------------------
block: 3
[1] "301" "302" "303" "304" "305"
------------------------------------------------------------
block: 4
[1] "405" "404" "403" "402" "401"
------------------------------------------------------------
block: 5
[1] "501" "502" "503" "504" "505"
```

```
head(book)
```

```
  plots block trt
1   101     1   C
2   102     1   B
3   103     1   v
```

```
4   104     1   D
5   105     1   t
6   106     1   A
```

For augmented ompletely randomized design, use the function design.crd().

1.11 Split Plot Designs

These designs have two factors, one is applied in plots and is defined as **trt1** in a randomized complete block design; and a second factor as **trt2** , which is applied in the subplots of each plot applied at random. The function design.split() permits to find the experimental plan for this design and its parameters are:

```
str(design.split)
```

```
function (trt1, trt2, r = NULL, design = c("rcbd", "crd", "lsd"),
    serie = 2, seed = 0, kinds = "Super-Duper", first = TRUE, randomization = TRUE)
 - attr(*, "srcref")= 'srcref' int [1:8] 46 1 102 1 1 1 46 102
  ..- attr(*, "srcfile")=Classes 'srcfilecopy', 'srcfile' <environment: 0x5574968a4c98>
```

1.11.1 Aplication

```
trt1<-c("A","B","C","D")
trt2<-c("a","b","c")
outdesign <-design.split(trt1,trt2,r=3,serie=2,seed=543)
book10 <- outdesign$book
head(book10)
```

```
  plots splots block trt1 trt2
1   101      1     1    D    b
2   101      2     1    D    a
3   101      3     1    D    c
4   102      1     1    B    a
5   102      2     1    B    b
6   102      3     1    B    c
```

```
p<-book10$trt1[seq(1,36,3)]
q<-NULL
for(i in 1:12)
q <- c(q,paste(book10$trt2[3*(i-1)+1],book10$trt2[3*(i-1)+2], book10$trt2[3*(i-1)+3]))
```

1.11.2 In plots:

```
print(t(matrix(p,c(4,3))))
```

```
     [,1] [,2] [,3] [,4]
[1,] "D"  "B"  "A"  "C"
[2,] "B"  "C"  "A"  "D"
[3,] "D"  "B"  "A"  "C"
```

1.11.3 In sub plots (split plot)

```
print(t(matrix(q,c(4,3))))
```

```
     [,1]    [,2]    [,3]    [,4]
[1,] "b a c" "a b c" "c a b" "c b a"
[2,] "b c a" "a b c" "b c a" "a c b"
[3,] "c a b" "b c a" "c a b" "a c b"
```

1.11.4 Serpentine enumeration

```
book <- zigzag(outdesign)
head(book,5)
```

```
  plots splots block trt1 trt2
1   101      1     1    D    b
2   101      2     1    D    a
3   101      3     1    D    c
4   102      1     1    B    a
5   102      2     1    B    b
```

1.12 Strip-Plot Designs

These designs are used when there are two types of treatments (factors) and are applied separately in large plots, called bands, in a vertical and horizontal direction of the block, obtaining the divided blocks. Each block constitutes a repetition and its parameters are:

```
str(design.strip)
```

```
function (trt1, trt2, r, serie = 2, seed = 0, kinds = "Super-Duper",
    randomization = TRUE)
 - attr(*, "srcref")= 'srcref' int [1:8] 45 1 84 1 1 1 45 84
  ..- attr(*, "srcfile")=Classes 'srcfilecopy', 'srcfile' <environment: 0x5574964f5cd8>
```

1.12.1 Aplication

```
trt1<-c("A","B","C","D")
trt2<-c("a","b","c")
outdesign <-design.strip(trt1,trt2,r=3,serie=2,seed=543)
book11 <- outdesign$book
head(book11)
```

```
  plots block trt1 trt2
1   101     1    D    b
2   102     1    D    a
3   103     1    D    c
4   104     1    B    b
5   105     1    B    a
6   106     1    B    c
```

```
t3<-paste(book11$trt1, book11$trt2)
B1<-t(matrix(t3[1:12],c(4,3)))
B2<-t(matrix(t3[13:24],c(3,4)))
B3<-t(matrix(t3[25:36],c(3,4)))
print(B1)
```

```
     [,1]  [,2]  [,3]  [,4]
[1,] "D b" "D a" "D c" "B b"
```

```
[2,] "B a" "B c" "A b" "A a"
[3,] "A c" "C b" "C a" "C c"
```

```
print(B2)
```

```
     [,1]  [,2]  [,3]
[1,] "C b" "C a" "C c"
[2,] "B b" "B a" "B c"
[3,] "A b" "A a" "A c"
[4,] "D b" "D a" "D c"
```

```
print(B3)
```

```
     [,1]  [,2]  [,3]
[1,] "A c" "A b" "A a"
[2,] "B c" "B b" "B a"
[3,] "D c" "D b" "D a"
[4,] "C c" "C b" "C a"
```

1.12.2 Serpentine enumeration

```
book <- zigzag(outdesign)
head(book)
```

```
  plots block trt1 trt2
1   101     1    D    b
2   102     1    D    a
3   103     1    D    c
4   106     1    B    b
5   105     1    B    a
6   104     1    B    c
```

```
array(book$plots,c(3,4,3))->X
t(X[,,1])
```

```
     [,1] [,2] [,3]
[1,]  101  102  103
[2,]  106  105  104
[3,]  107  108  109
[4,]  112  111  110
```

```
t(X[,,2])
```

```
     [,1] [,2] [,3]
[1,]  201  202  203
[2,]  206  205  204
[3,]  207  208  209
[4,]  212  211  210
```

```
t(X[,,3])
```

```
     [,1] [,2] [,3]
[1,]  301  302  303
[2,]  306  305  304
[3,]  307  308  309
[4,]  312  311  310
```

1.13 Factorial

The full factorial of n factors applied to an experimental design (CRD, RCBD and LSD) is common and this procedure in **agricolae** applies the factorial to one of these three designs and its parameters are:

```
str(design.ab)
```

```
function (trt, r = NULL, serie = 2, design = c("rcbd", "crd", "lsd"),
    seed = 0, kinds = "Super-Duper", first = TRUE, randomization = TRUE)
 - attr(*, "srcref")= 'srcref' int [1:8] 51 1 107 1 1 1 51 107
  ..- attr(*, "srcfile")=Classes 'srcfilecopy', 'srcfile' <environment: 0x557496394130>
```

To generate the factorial, you need to create a vector of levels of each factor, the method automatically generates up to 25 factors and r repetitions.

```
trt <- c (4,2,3) # three factors with  4,2 and 3 levels.
```

to crd and rcbd designs, it is necessary to value r as the number of repetitions, this can be a vector if unequal to equal or constant repetition (recommended).

```
trt<-c(3,2) # factorial 3x2
outdesign <-design.ab(trt, r=3, serie=2)
book12 <- outdesign$book
head(book12) # print of the field book
```

```
  plots block A B
1   101     1 2 2
2   102     1 2 1
3   103     1 3 2
4   104     1 1 2
5   105     1 1 1
6   106     1 3 1
```

1.13.1 Serpentine enumeration

```
book <- zigzag(outdesign)
head(book)
```

```
  plots block A B
1   101     1 2 2
2   102     1 2 1
3   103     1 3 2
4   104     1 1 2
5   105     1 1 1
6   106     1 3 1
```

factorial $2 \times 2 \times 2$ with 5 replications in completely randomized design.

```
trt<-c(2,2,2)
crd<-design.ab(trt, r=5, serie=2,design="crd")
names(crd)
```

```
[1] "parameters" "book"
```

```
crd$parameters
```

```
$design
```

```
[1] "factorial"

$trt
[1] "1 1 1" "1 1 2" "1 2 1" "1 2 2" "2 1 1" "2 1 2" "2 2 1" "2 2 2"

$r
[1] 5 5 5 5 5 5 5 5

$serie
[1] 2

$seed
[1] 1923434691

$kinds
[1] "Super-Duper"

[[7]]
[1] TRUE

$applied
[1] "crd"
```

```
head(crd$book)
```

```
  plots r A B C
1   101 1 2 2 2
2   102 2 2 2 2
3   103 1 2 1 1
4   104 1 1 2 1
5   105 1 1 1 1
6   106 2 1 2 1
```

References

Cochran, W. G., and Cox, G. M. (1992). *Experimental Designs.* John Wiley & Sons, New York.

Kuehl, R. O. (2000). *Designs of Experiments: Statistical Principles of Research Design and Analysis.* Duxbury Press.

Le Clerg, E. L., Leonard, W. H., Erwin, L., Warren, H. L., and Andrew, G. C. (1962). *Field Plot Techniques.* Burgess Publishing Company, Minneapolis, Minnesota.

Montgomery, D. C. (2002). *Design and Analysis of Experiments.* John Wiley & Sons, New York.