



Designing comparative experiments using R

(Chris Brien and Sam Rogers)

I. Concepts in experimental design

(Software and materials at
<https://tinyurl.com/BrienWorkshop>)

Tentative programme

09:00–10:00: Concepts in experimental design.

10:00–10:45: Orthogonal experimental design in \mathbb{R} .

10:45 –11:15: Morning tea

11:15–12:15: Nonorthogonal experimental design.

12:15–13:00: Nonorthogonal experimental design in \mathbb{R} .

13:00–13:45: Lunch

13:45–14:15: Nonorthogonal experimental design in \mathbb{R} (continued).

14:15–15:15: Advanced experimental design: multiphase and p-rep designs.

15:15 –15:45: Afternoon tea

15:45–17:00: Using \mathbb{R} for advanced experimental design.

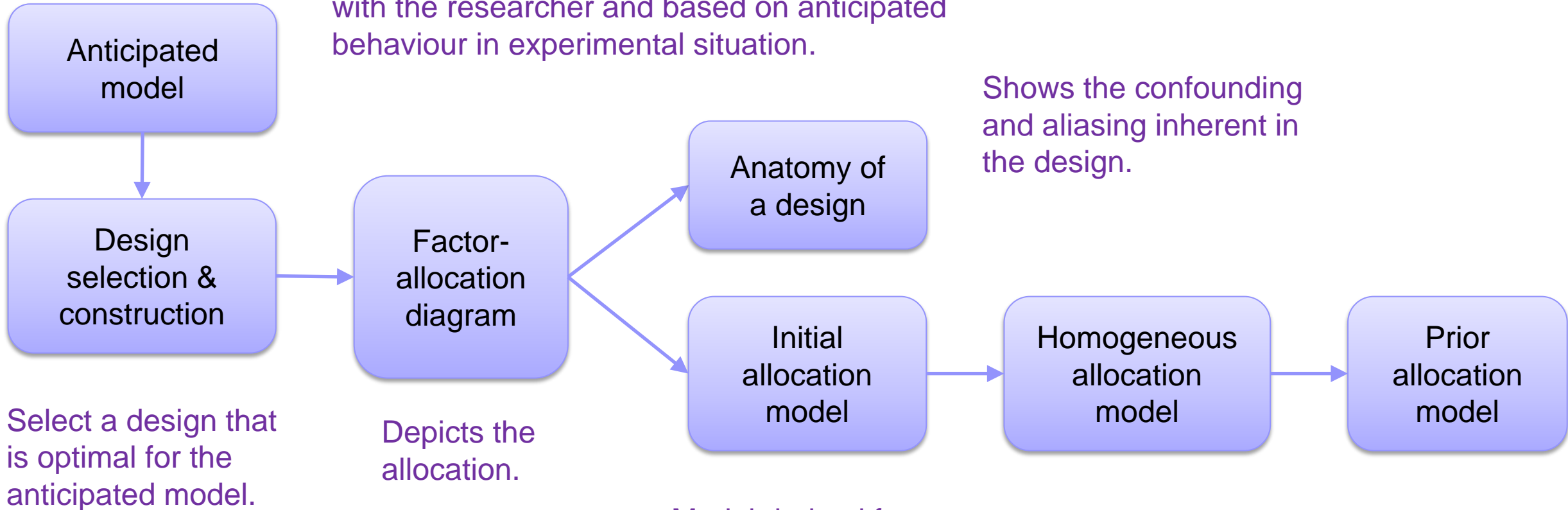
Outline

1. A paradigm for designing experiments.
2. Experiment on a 5 x 5 grid of plots.
3. Split-unit design.
4. Summary of constructing orthogonal designs.
5. Software and practical sessions.

1. A paradigm for designing experiments

(Brien & Demétrio, 2009;
Brien, 2017)

Anticipated model determined in consultation with the researcher and based on anticipated behaviour in experimental situation.



The design specifies the allocation of a set of factors, the allocated factors, to a second set of factors, the recipient factors. Allocation is usually by randomization, but not always.

Design optimality

- For comparative experiments, A-optimality is the favoured optimality criterion.
 - The definition of A-optimality is that it minimizes the total variance of the predictions or Prediction Error Variance (PEV) (Kiefer, 1959)
 - The PEV is the same as the average variance of pairwise differences (AVPD):
 - when terms to be optimized (Treatments) are fixed;
- Often fixed-model A-optimal designs are sought for comparative experiments:
 - All model terms are assume fixed, except the residuals.
 - Not just the treatments but blocks, row, columns and the like are fixed.

2. Experiment on a 5 x 5 grid of plots

- Suppose have 25 plots arranged in a grid of 5 rows × 5 columns.
- We want to assign 5 lines to the 25 plots.
- What design to use?
- Using the paradigm, we ask the question:
 - “what is the anticipated model?”

Case 1: row differences only

- Suppose that the researcher says that they are confident that there will be row differences, but column differences are very unlikely.
 - That is, the anticipated model is $\text{Lines} + \text{Rows} + \text{Rows:Columns}$ (units).
 - But, to formulate a model, it is necessary to identify the fixed and random terms?
 - Commonly, both Lines and Rows are fixed; Rows:Columns is random; i.e. a fixed-effects model is assumed.
 - The anticipated model becomes $\text{Lines} + \text{Rows} \mid \underline{\text{Rows:Columns}}$ (fixed terms are left of the '|'; the underline indicates an identity term).
- What is the optimal design for this model?
- It is known that a Randomized Complete Block (RCBD) is A-optimal for this model:
 - It minimizes the AVPD and so that is the design that will be used.
 - Of course, it is not usual to explicitly go through this process to choose a design for a situation as simple as this.
 - I argue that it is instructive to realize that choosing an optimal design for a model underscores what we usually do when designing an experiment.

RCBD on a grid of plots

■ For an RCBD:

- Allocated (treatment) factor is Lines.
- Recipient (unit) factors are Rows and Columns.
- Each line is applied once and only once in each row.
- Are Rows and Columns nested or crossed?
 - Given that Columns is not in the model, Rows is nested within Columns:
 - consistent differences between Columns across Rows are not anticipated;
 - instead variable differences within Rows are anticipated.
- Thus, the order of the treatments is randomized within each row.
- A method of achieving this randomization is:
 - take a systematic design for the allocated and recipient factors;
 - permute the recipient factors.

Plot of Lines

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
	1	2	3	4	5

Rows

Columns

Randomization by permutation of recipient factors

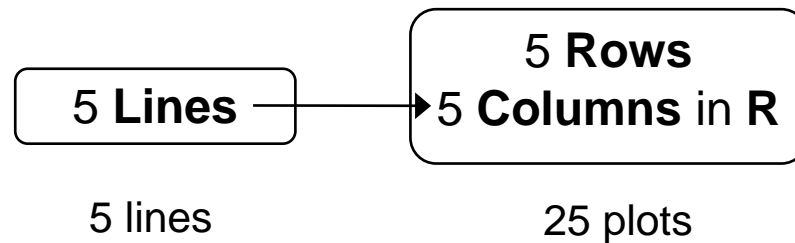
- Permutations for an RCBD with $b = 2$, $k = v = 2$.
- The allowable permutations are:
 - those that permute the blocks as a whole, and
 - those that permute the units within a block;
 - there are $b!(k!)^b = 2!(2!)^2 = 8$.

unit	Blocks	Units	Treatments	Permutation	Permuted	
					Blocks	Units
1	1	1	1	4	2	2
2	1	2	2	3	2	1
3	2	1	1	1	1	1
4	2	2	2	2	1	2

- Equivalent to Treatments randomization 1, 2, 2, 1.
- **designRandomize** implements this method of randomizing
 - The permuted Blocks and Units and the Treatments are put back into standard order.

RCBD on a 5 x 5 grid of plots (cont'd)

- What is its factor-allocation diagram for the design?



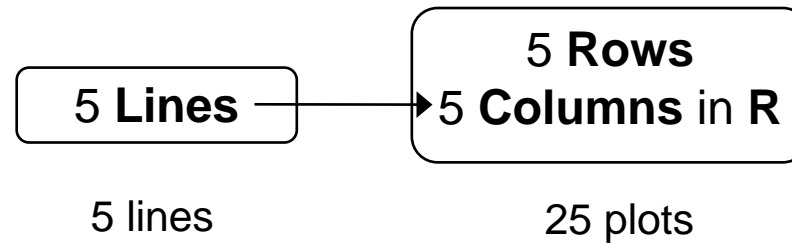
- Two sets of objects (uncapitalized names) with associated factors (capitalized names):
 - Allocated objects: 5 lines;
 - Allocated factor indexes lines: {Lines};
 - Recipient objects: 25 plots;
 - Recipient set of factors indexes plots: {Rows, Columns}.
- Columns are nested within Rows in the anticipated model and so are permuted within Rows to randomize Lines.
- This is in spite of the plots being in a grid, for which Rows and Columns are intrinsically crossed.

Plot of Lines

1	C	E	A	B	D
2	C	A	E	D	B
3	E	C	B	A	D
4	E	A	D	B	C
5	D	C	B	A	E
	1	2	3	4	5

Columns

Factor-allocation diagram for the RCBD (cont'd)



- One allocation (randomization):
 - a set of lines is allocated to a set of plots.
- The set of factors belonging to a set of objects, i.e. in each panel, forms a **tier**:
 - they have the same status in the allocation (randomization):
 - allocated or recipient.
- Textbook experiments are two-tiered:
- The factor-allocation diagram shows the EU and restrictions on randomization/allocation.



How to use `designRandomize` to get a layout for an RCBD on 5×5 grid

- There are many ways to approach this in R:
 - I chose to always start by creating a systematic `data.frame` to make the process more transparent.

`fac.gen` is convenient because it produces a `data.frame` containing factors `Rows` and `Columns` and another with `Lines`; they are combined

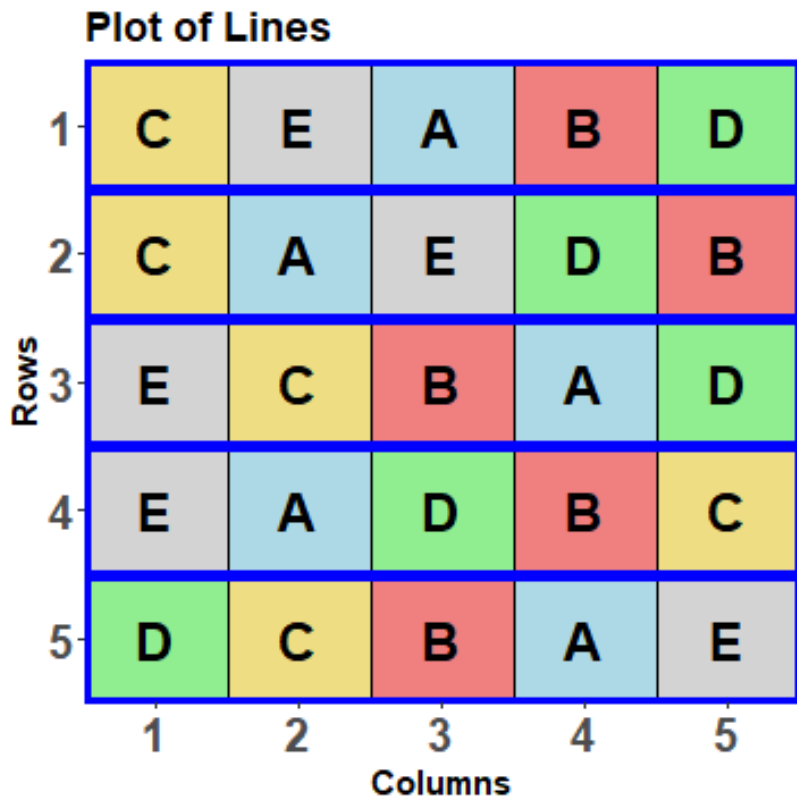
Order of `Rows` then `Columns` in `generate` means that `Columns` will move faster than `Rows`.

`Lines` is in a systematic order appropriate for an RCBD, consistent with the `Rows-Columns` order.

```
> b <- 5
> t <- 5
> RCBD.sys <- cbind(fac.gen(generate = list(Rows=b, Columns=t)),
+                  fac.gen(generate = list(Lines = LETTERS[1:t]), times = b))
> RCBD.lay <- designRandomize(allocated = RCBD.sys["Lines"],
+                             recipient = RCBD.sys[c("Rows", "Columns")],
+                             nested.recipients = list(Columns = "Rows"),
+                             seed = 1134)
> RCBD.lay
```

Note nesting of `Columns` within `Rows`.

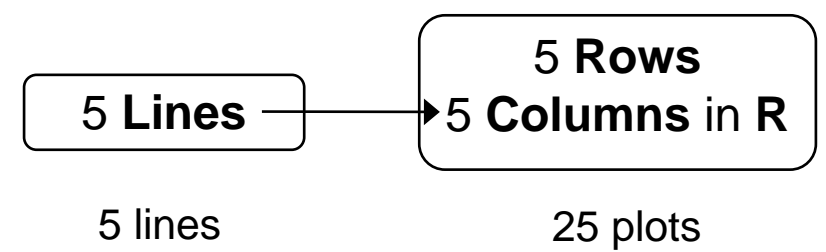
RCBD.lay



	Rows	Columns	Lines
1	1	1	C
2	1	2	E
3	1	3	A
4	1	4	B
5	1	5	D
6	2	1	C
7	2	2	A
8	2	3	E
9	2	4	D
10	2	5	B
11	3	1	E
12	3	2	C
13	3	3	B
14	3	4	A
15	3	5	D
16	4	1	E
17	4	2	A
18	4	3	D
19	4	4	B
20	4	5	C
21	5	1	D
22	5	2	C
23	5	3	B
24	5	4	A
25	5	5	E

- This randomization results from the specified permutations.
 - Columns within Rows
 - Rows

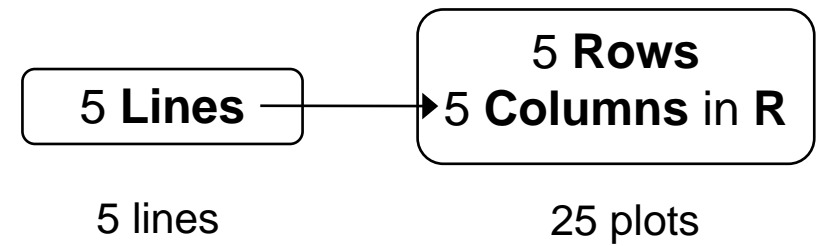
The initial allocation-based mixed model



- This model is based on the factor allocation diagram.
 - Take all combinations of the factors within a tier, subject to the restriction that if a factor is nested within another then the nesting factor must always be included in terms involving it.
 - The allocated (treatment) **terms** derived from the allocation factors are designated as fixed.
 - The recipient (unit) **terms** derived from the recipient factors are designated as random.

Lines | Rows + Rows:Columns
 - Because the allocation involved randomization, this model is equivalent to a randomization model (provided Rows is allowed to be negative).
 - This model and the anticipated model are different — here Rows is random.
 - The Rows terms could be moved to the fixed model to form a homogeneous allocation model, and this might become the prior allocation model.

Anatomy of the design



What is the purpose?

- To evaluate the design by establishing the confounding present in it.
- Provides insight into the analysis of the experiments based upon it.

What do you need?

- The tiers: the sets of factors in the panels.
 - plots tier: the recipient factors are {Rows, Columns};
 - lines tier: the allocated factor is {Lines}.
- The relationships between the factors within a tier.
 - Columns within Rows (given the allocation).
- The layout for the experiment, but no response values.

What do you get?

- A table showing the confounding relationships between **sources**.

Term versus source

- A term represents the differences between the levels of a (generalized or joint) factor in a model.
 - Mathematically it is of the form $\mathbf{X}\beta$ or $\mathbf{Z}\mathbf{u}$.
 - Its dimension is the number of columns of \mathbf{X} or \mathbf{Z} .
 - For example, factor Rows:Columns represents the difference between the combinations of Rows and Columns.
 - It has dimension bt : the \mathbf{Z}_{RC} matrix has bt columns. (The associated means projector $\mathbf{Z}_{RC}(\mathbf{Z}_{RC}^T \mathbf{Z}_{RC})^{-1} \mathbf{Z}_{RC}$ has dimension bt .)

Marginality relationships between terms

- A property of the column spaces of the incidence matrices for the terms.
 - One term is marginal to another if the column space of the first term is a subspace of that of the marginal term.
 - This property is independent of the replications of the levels of the factors that make up the term.
 - For example, the term for A is marginal to A:B irrespective of the number of values for each combination of the levels of A and B.
 - When nesting is explicit, because nested factors are numbered within the nesting factors (e.g. Columns within Rows),
 - then a term is marginal to another if its factors are a subset of those in the marginal term (e.g. Rows is marginal to Rows:Columns).

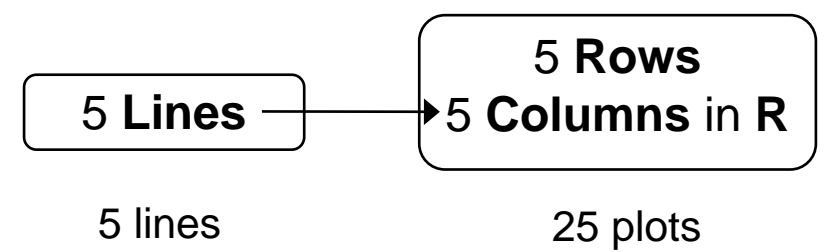
Term versus source

- A term represents the differences between the levels of a (generalized or joint) factor in a model.
 - For example, Rows:Columns has dimension bt , being the number of columns in the matrix \mathbf{Z}_{RC} . (Also, the associated projector $\mathbf{Z}_{RC}(\mathbf{Z}_{RC}^T \mathbf{Z}_{RC})^{-1} \mathbf{Z}_{RC}$ has dimension bt .)
- A source represents differences after marginal terms are eliminated.
 - Mathematically it is characterized by the projection matrix, \mathbf{Q} , that is the matrix of the quadratic form for its sum of squares, $\mathbf{y}^T \mathbf{Q} \mathbf{y}$.
 - Its dimension is the rank of \mathbf{Q} .
 - The source Columns[Rows] represents Rows:Columns differences, after the marginal Rows term has been eliminated, i.e. differences between Columns within Rows.
 - Its dimension is $bt - b = (t - 1)b$.
- For each term in a model there is a source in the anatomy.

Notation for sources

- When all factors and/or variables in a term are crossed,
 - They form an interaction, in which all are joined together with hashes (#),
 - e.g. A#B#C.
- When there is nesting between factors and/or variables in a term,
 - the nested factors are placed first and joined by hashes, while the nesting factors are enclosed in square brackets ([...]) and joined by colons (:).
- Rule for determining where factors occur in a source with nesting is:
 - only those factors in the term that nest any of the other factors must be in the square brackets joined by ':';
 - the rest are put to the left of the square brackets joined by '#'.
 - e.g. A#B[C:D], where C and D nest one or more of A and B;
here C:D indicates all observed combinations of C and D.

Anatomy of the design



- An anatomy is based on the allocation (in a factor-allocation diagram).
 - A formula per panel with nesting in panel incorporated.
 - Shorthand for terms (Wilkinson and Rogers, 1973):
 - $A/B = A + A:B$ ('/' is called the nesting operator)
 - $A*B = A + B + A:B$ ('*' is called the crossed operator)
- ```
> RCBD.canon <- designAnatomy(formulae = list(plots = ~ Rows/Columns,
+ lines = ~ Lines),
+ data = RCBD.lay)
```
- The terms for each formula are transformed into sources by **designAnatomy**.
    - To do this, it works out the marginality relationships between terms.

# Anatomy of the design

- Shows how Lines is confounded with the plots sources?

```
> RCBD.canon <- designAnatomy(formulae = list(plots = ~ Rows/Columns,
+ lines = ~ Lines),
+ data = RCBD.lay)
> summary(RCBD.canon)
```

Summary table of the decomposition for plots & lines

| Source.plots  | df1 | Source.lines | df2 | aefficiency | eefficiency | order |
|---------------|-----|--------------|-----|-------------|-------------|-------|
| Rows          | 3   |              |     |             |             |       |
| Columns[Rows] | 20  | Lines        | 4   | 1.0000      | 1.0000      | 1     |
|               |     | Residual     | 16  |             |             |       |

All these  
are one and  
so design is  
orthogonal.

Lines is confounded with  
Columns[Rows] (as expected).

The Residual measures differences between Columns  
within Rows. (not the interaction of Lines and Rows).

# How is it done?

- Get the Mean projectors for **terms** in each tier ( $\mathbf{X}[\mathbf{X}^T\mathbf{X}]^{-1}\mathbf{X}^T$  or  $\mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T$ ) :

- Allocated tier: Mean, Lines;  $\mathbf{M}_0, \mathbf{M}_L$ .
- Recipient tier: Mean, Rows, Rows:Columns;  $\mathbf{M}_0, \mathbf{M}_R, \mathbf{M}_{RC}$ .

$$\mathbf{X}_L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, \mathbf{M}_L = \frac{1}{2}\mathbf{X}_L^T\mathbf{X}_L = \begin{bmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{bmatrix}$$

- Form **source** projectors by orthogonalizing mean projectors for marginal terms:

- Allocated tier: Mean, Lines;  $\mathbf{Q}_0 = \mathbf{M}_0, \mathbf{Q}_L = \mathbf{M}_L - \mathbf{M}_0$ .
- Recipient tier: Mean, Rows, Columns[Rows];  $\mathbf{P}_0 = \mathbf{M}_0, \mathbf{P}_R = \mathbf{M}_R - \mathbf{P}_0, \mathbf{Q}_{C[R]} = \mathbf{P}_{RC} - \mathbf{Q}_0 - \mathbf{Q}_R$ .
- It can be shown that, for each tier, the **source** projectors are orthogonal.

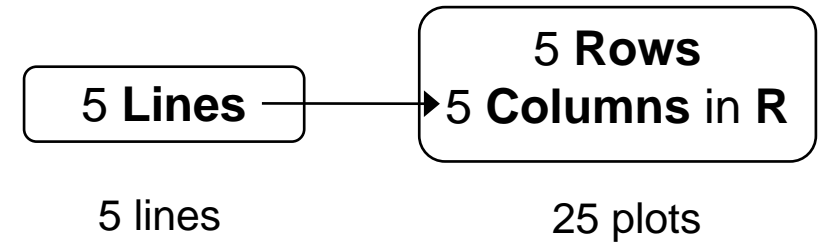
- Now investigate the relationship between all pairs of one lines source with one plot source:

- that is, compare the lines **source** projector ( $\mathbf{Q}_L$ ) with plots **source** projectors ( $\mathbf{P}_R, \mathbf{P}_{C[R]}$ ) through the nonzero eigenvalues of  $\mathbf{PQP}$  products:
  - $\mathbf{P}_R\mathbf{Q}_L\mathbf{P}_R$ , which has all zero eigenvalues;
  - $\mathbf{P}_{C[R]}\mathbf{Q}_L\mathbf{P}_{C[R]}$ , which has four eigenvalues all equal to one.
- The eigenvalues are the canonical efficiency factors.
- This information is summarized in the anatomy table.

# Canonical efficiency (eigenvalue) statistics

- A is the harmonic mean of the efficiency factors.
- M is the arithmetic mean of the efficiency factors.
- S is the variance of the efficiency factors.
- X is the maximum of the efficiency factors.
- E is the minimum of the efficiency factors.
- Order is the number of unique efficiency factors.
- DForthog is the number of efficiency factors equal to one.

# Anatomy of the design



All eigenvalues between **Lines** and **Rows** are zero so **Lines** is not confounded with **Rows**.

Summary table of the decomposition for plots & lines

| Source        | plots | df1 | Source   | lines | df2 | aefficiency | eefficiency | order |
|---------------|-------|-----|----------|-------|-----|-------------|-------------|-------|
| Rows          |       | 3   |          |       |     |             |             |       |
| Columns[Rows] | 20    |     | Lines    |       | 4   | 1.0000      | 1.0000      | 1     |
|               |       |     | Residual |       | 16  |             |             |       |

There are four eigenvalues between **Lines** and **Columns[Rows]** that are equal to one.

The three efficiency statistics are equal to one, indicating that all **Lines** information is confounded with **Columns[Rows]**.

So there are 4 df for **Lines** confounded with **Columns[Rows]**, i.e. all 4 df.

The advantage of this design is that, because no **Lines** information is confounded with **Rows**, **Rows** differences will not contribute to the variability of **Lines**.



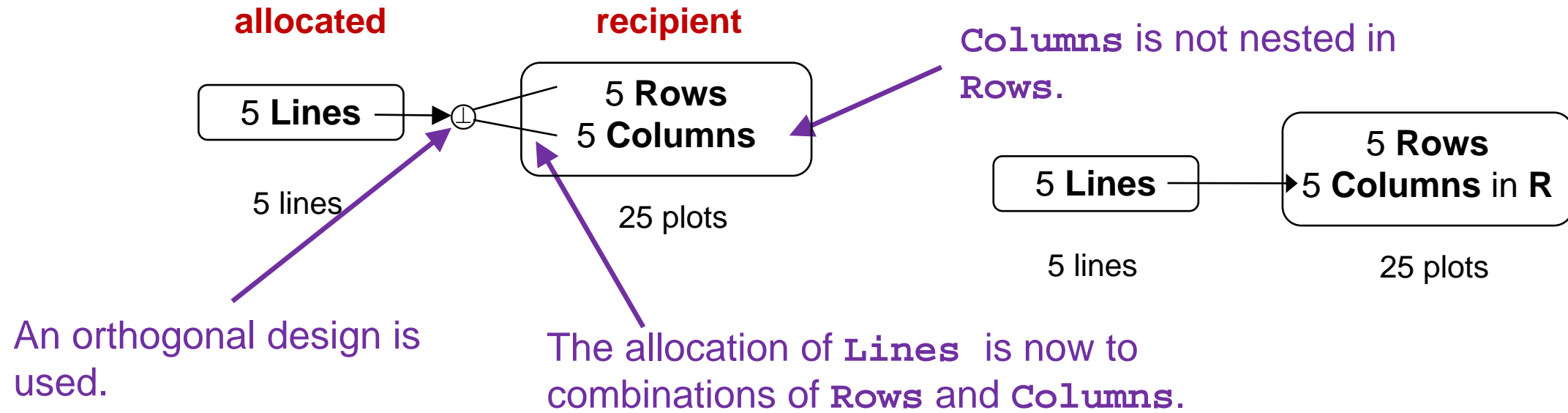
# Why anatomy?

- Is it not just a skeleton **anova** table?
  - Yes, it is, but ...
- An **anova** is used to analyse data.
  - So a skeleton **anova** is showing you how the analysis of some data will look.
- These days, I generally don't do **anovas**— I mainly do mixed model analyses of data.
- An **anatomy** is the analysis of a design, rather than of data.
  - It may be performed irrespective of the method to be used in analyzing the data.
  - Further, **anatomies** are based on the allocations for a design.
- So to emphasize this distinction I refer to the **anatomy** of a design.

## Case 2: Row and Column differences probable

- What is the anticipated model?
  - Anticipating row and column differences means that we should consider a model that includes Rows and Columns main effects.
  - Lines | Rows + Columns + Rows:Columns
    - (taking Rows and Columns to be random).
- Need a design that is A-optimal for this model.
  - The Latin Square Design (LSD) is such a design.
  - It is A-optimal whether Rows or Columns are fixed or random.
- Are Rows and Columns crossed or nested? Why?
  - Crossed because expect consistent differences between Rows (across Columns) and between Columns (across Rows) and this has been allowed for in the anticipated model.
  - In this case this is consistent with the intrinsic crossing of Rows and Columns.
  - It implies that Rows and Columns should be independently permuted.

# Factor allocation diagram for an LSD



- Again, the initial allocation model consists of terms from all combinations of the factors in each panel, taking into account any nesting:

**Lines | Rows + Columns + Rows:Columns**


- This model and the anticipated model are the same.



# LSD on 5 × 5 grid using designRandomize and designLatinSqrSys

```
> designLatinSqrSys(t)
[1] 1 2 3 4 5 2 3 4 5 1 3 4 5 1 2 4 5 1 2 3 5 1 2 3 4

b <- 5
t <- 5
> LSD.sys <- cbind(fac.gen(list(Rows=b, Columns=t)),
+ Lines = factor(designLatinSqrSys(t), labels = LETTERS[1:t]))
> LSD.lay <- designRandomize(allocated = LSD.sys["Lines"],
+ recipient = LSD.sys[c("Rows", "Columns")],
+ seed = 141)
```



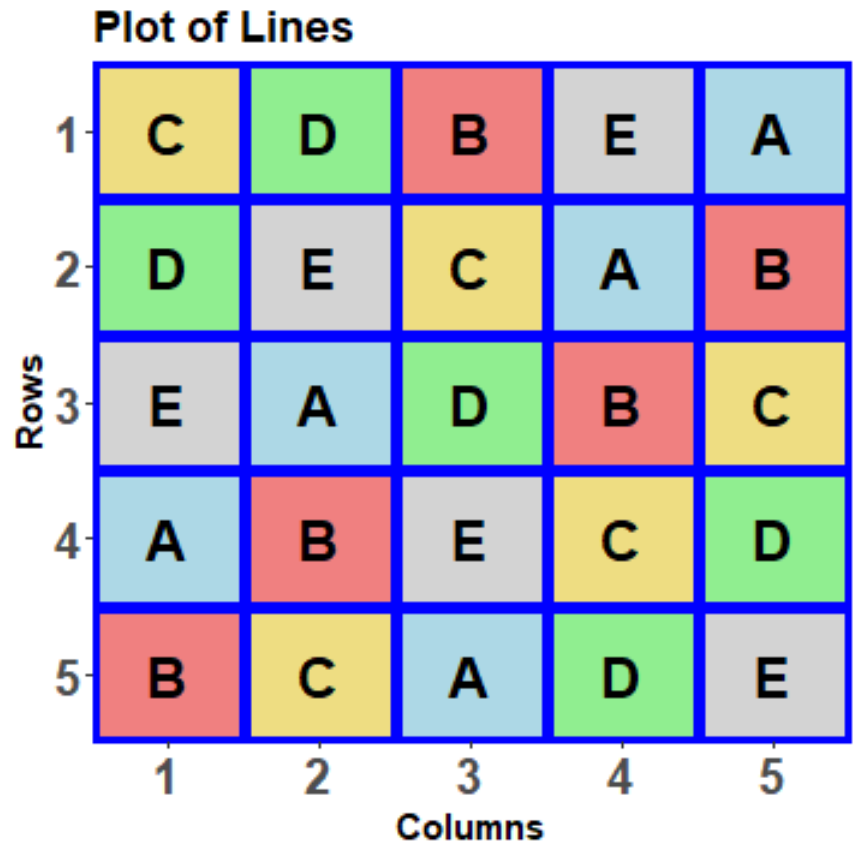
no nested.recipients, so  
Rows and Columns are crossed.

Columns are permuted  
and Rows are permuted.

## ■ Compare to RCBD

```
> RCBD.sys <- cbind(fac.gen(list(Rows=b, Columns=t)),
+ fac.gen(generate = list(Lines = LETTERS[1:t]), times = b))
> RCBD.lay <- designRandomize(allocated = RCBD.sys["Lines"],
+ recipient = RCBD.sys[c("Rows", "Columns")],
+ nested.recipients = list(Columns = "Rows"),
+ seed = 1134)
```

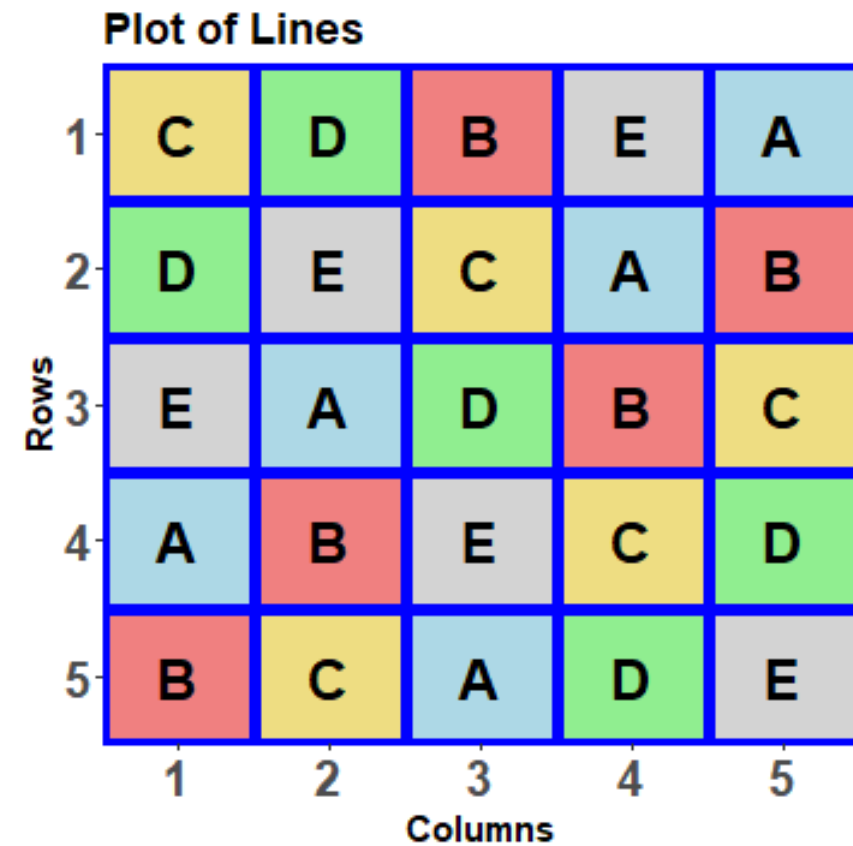
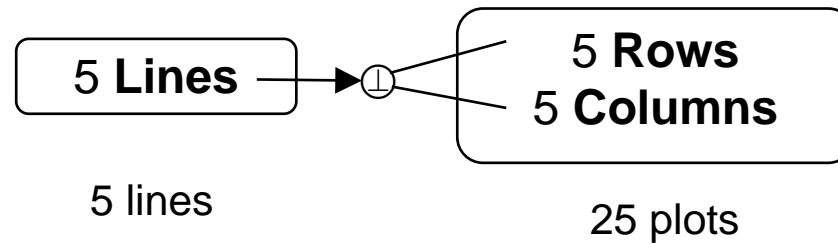
# LSD.lay



|    | Rows | Columns | Lines |
|----|------|---------|-------|
| 1  | 1    | 1       | C     |
| 2  | 1    | 2       | D     |
| 3  | 1    | 3       | B     |
| 4  | 1    | 4       | E     |
| 5  | 1    | 5       | A     |
| 6  | 2    | 1       | D     |
| 7  | 2    | 2       | E     |
| 8  | 2    | 3       | C     |
| 9  | 2    | 4       | A     |
| 10 | 2    | 5       | B     |
| 11 | 3    | 1       | E     |
| 12 | 3    | 2       | A     |
| 13 | 3    | 3       | D     |
| 14 | 3    | 4       | B     |
| 15 | 3    | 5       | C     |
| 16 | 4    | 1       | A     |
| 17 | 4    | 2       | B     |
| 18 | 4    | 3       | E     |
| 19 | 4    | 4       | C     |
| 20 | 4    | 5       | D     |
| 21 | 5    | 1       | B     |
| 22 | 5    | 2       | C     |
| 23 | 5    | 3       | A     |
| 24 | 5    | 4       | D     |
| 25 | 5    | 5       | E     |

- This randomization results from the specified permutations.
  - Columns
  - Rows

# What sources and confounding?



- The initial allocation model is:

Lines | Rows + Columns + Rows:Columns

- There is a source for each term:

- Allocated source: Lines;
- Recipient sources: Rows, Columns, Rows#Columns.

The interaction of Rows and Columns because both Rows and Columns in the model.

- Treats will be confounded with which recipient (unit) sources?

- Not with Rows or Columns;
- With Rows#Columns.

# Check properties using designAnatomy

```
> LSD.canon <- designAnatomy(formulae = list(plots = ~ Rows*Columns,
+
+
+ data = LSD.lay)
> summary(LSD.canon)
```

Reflects the factor-  
allocation diagram  
— note the ‘\*’.

Summary table of the decomposition for plots & lines

| Source.plots | df1 | Source.lines | df2 | aefficiency | eefficiency | order |
|--------------|-----|--------------|-----|-------------|-------------|-------|
| Rows         | 4   |              |     |             |             |       |
| Columns      | 4   |              |     |             |             |       |
| Rows#Columns | 16  | Lines        | 4   | 1.0000      | 1.0000      | 1     |
|              |     | Residual     | 12  |             |             |       |


Lines is confounded with only  
Rows#Columns.

# Comparing anatomies

```
> RCBD.canon <- designAnatomy(formulae = list(plots = ~ Rows/Columns,
+ lines = ~ Lines),
+ data = RCBD.lay)
```

| Source.plots | df1 | Source.lines | df2 | aefficiency | eefficiency | order |
|--------------|-----|--------------|-----|-------------|-------------|-------|
|--------------|-----|--------------|-----|-------------|-------------|-------|

|               |    |          |    |        |        |   |
|---------------|----|----------|----|--------|--------|---|
| Rows          | 3  |          |    |        |        |   |
| Columns[Rows] | 20 | Lines    | 4  | 1.0000 | 1.0000 | 1 |
|               |    | Residual | 16 |        |        |   |

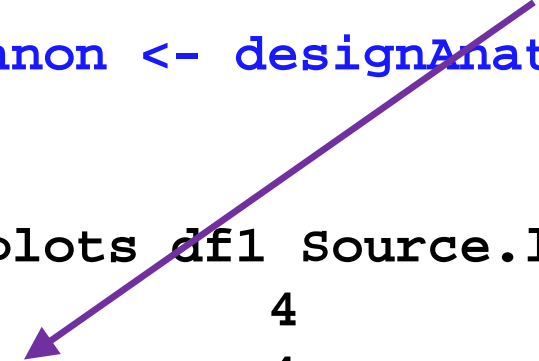


Columns within Rows includes Columns and so Residual has an extra 4 df — tradeoff.

```
> LSD.canon <- designAnatomy(formulae = list(plots = ~ Rows*Columns,
+ lines = ~ Lines),
+ data = LSD.lay)
```

| Source.plots | df1 | Source.lines | df2 | aefficiency | eefficiency | order |
|--------------|-----|--------------|-----|-------------|-------------|-------|
|--------------|-----|--------------|-----|-------------|-------------|-------|

|              |    |          |    |        |        |   |
|--------------|----|----------|----|--------|--------|---|
| Rows         | 4  |          |    |        |        |   |
| Columns      | 4  |          |    |        |        |   |
| Rows#Columns | 16 | Lines    | 4  | 1.0000 | 1.0000 | 1 |
|              |    | Residual | 12 |        |        |   |





# Recap of experiment on a 5 x 5 grid of plots

- For 25 plots arranged in a grid of 5 rows × 5 columns:
  - Rows and Columns are intrinsically crossed.
- But just because they are crossed there is no compunction to use a row-column design. The design could be any one of a:
  - Completely Randomized (CRD),
  - Randomized Complete Block (RCBD), or
  - Latin Square Design (LSD)?
- Which depends on whether the researcher anticipates appreciable (i) row and/or (ii) columns differences.
  - The researcher's assessment is encapsulated in the anticipated model.
- We have seen how the anticipated model influences:
  - the nesting and crossing relationships between the recipient factors (Rows and Columns);
  - hence, the permutations that are appropriate for randomizing a design; and
  - the confounding that results from the design.

### 3. Split-unit design

- They involve more than one treatment factor and so are factorial experiments.
- Designs in which main effects confounded with more variable units, such as large plots.
- Their defining attribute is that there is randomization to two different physical entities such that some main effects are randomized to the more variable entities.
- The **standard split-unit design** is one in which two factors, say A and B with  $a$  and  $b$  levels, respectively are assigned as follows:
  - one of the factors, A say, is randomized according to an RCBD with say  $r$  blocks and
  - each of the RCBD's  $ra$  units, called the **main units**, is split into  $b$  **subunits** (or split-units) and levels of B randomized independently to the subunits in each main unit. Altogether the experiment involves  $n = rab$  subunits.

# Split-unit principle

- Very flexible principle that can be used to generate a large number of different types of experiments.
- For example, the main units could be arranged in any of a CRD, RCBD, Latin square, BIBD, Youden Square
  - each unit of the design is subdivided into subunits.
- The subunits may utilize more complicated designs as well.
  - For example, the main units may be arranged in a RCBD each of which are subdivided in such a way as to allow a Latin Square to be placed in each main unit.
- Also, subunits can be split into subsubunits and subsubunits into ...
- Nor is one restricted to applying just one factor to each type of unit.
  - More than one factor can be randomized to main units, more than one to subunits and so on.
- The standard split-unit design is nearly the simplest possibility; only a CRD in the main units would be simpler.

# When to use a split-unit design

1. When the physical attributes of a factor require the use of larger units of experimental material than other factors.
  2. When it is desired to incorporate an additional factor into an experiment.
  3. When it is expected that differences amongst the levels of certain factors are larger than amongst those of other factors.
    - The levels of the factors with larger differences are randomized to main units.
    - One effect of this may be to increase the precision of comparisons between the levels of the other factors.
  4. When it is desired to ensure greater precision between some factors as compared to other factors.
    - Irrespective of the size of the differences between the main unit treatment factors, it is desired to increase the precision of some factors by assigning them to subunits.
- Note that the last two of these situations are utilising the anticipated greater variability of main units relative to subunits.
- That is, we are expecting the larger units to be more variable than the smaller units.
  - Generally, we expect a term will have more variability than those to which it is marginal (e.g. Rows are more variables than Rows:Columns).



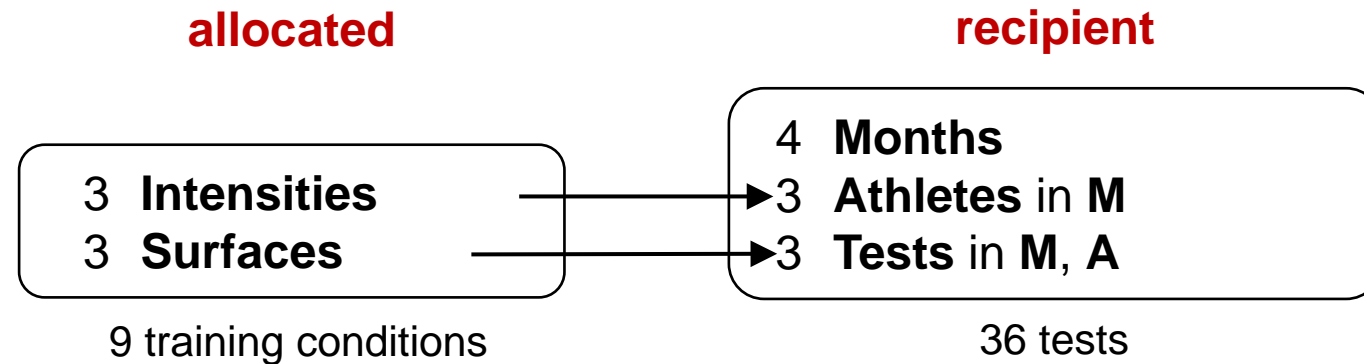
# A standard athlete training experiment

Peeling et al. (2009) ; Brien, Harch, Correll, Bailey (2011)

- 9 training conditions to be investigated:
  - combinations of 3 surfaces and 3 intensities of training.
- Testing is to be conducted over 4 Months:
  - In each month, 3 endurance athletes are to be recruited;
  - Each athlete will undergo 3 tests, separated by 7 days, under 3 different training conditions.
- On completion of each test, the athlete's heart rate is measured.
- Anticipated model, determined with the researcher:
  - Intensities + Surfaces + Intensities:Surfaces | Months + Months:Athletes + Months:Athletes:Tests.
  - Consistent differences between Athletes across Months is unlikely because different athletes are involved each Month.
  - Consistent differences between Tests across Athletes are not anticipated.

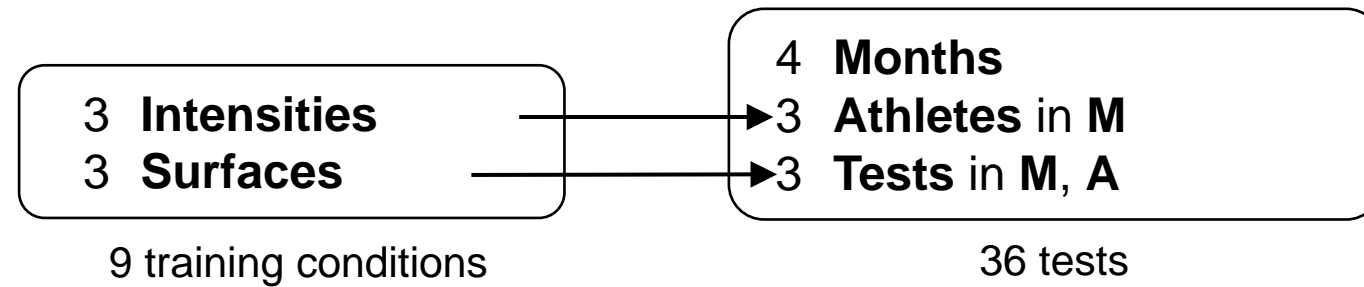
# Factor-allocation diagram for the standard athlete training experiment

- Given the experimental set-up and the anticipated model, the tiers are as given in the following panels with the nesting relationships shown:



- Assume the prime interest is in surface differences:
  - intensities are only included to observe the surfaces over a range of intensities.
- Also expect variability of Months:Athletes to be greater than Months:Athletes:Tests
  - Months:Athletes is marginal to Months:Athletes:Tests.
- Use a split-unit design (as per situation 4).
  - Assign Surfaces to Tests so that Surfaces has greater precision .

# Factor-allocation diagram for the standard athlete training experiment (cont'd)



- One allocation (randomization):
  - a set of training conditions is allocated to a set of tests using a single permutation of the tests.
  - The recipient tier is {Months, Athletes, Tests};
  - The allocated tier is {Intensities, Surfaces}.
- The initial allocation model is the same as the anticipated model.
- Because all allocation is by randomization the initial allocation model is equivalent to a randomization model.



# A randomized layout using designRandomize

Order of **Months** then **Athletes** then **Tests** means that **Tests** will change faster than **Athletes**, which will change faster than **Months**.

Similarly for **Intensities** and **Surfaces**.

Supplied to **allocated** is **Intensities** and **Surfaces** in a systematic order.

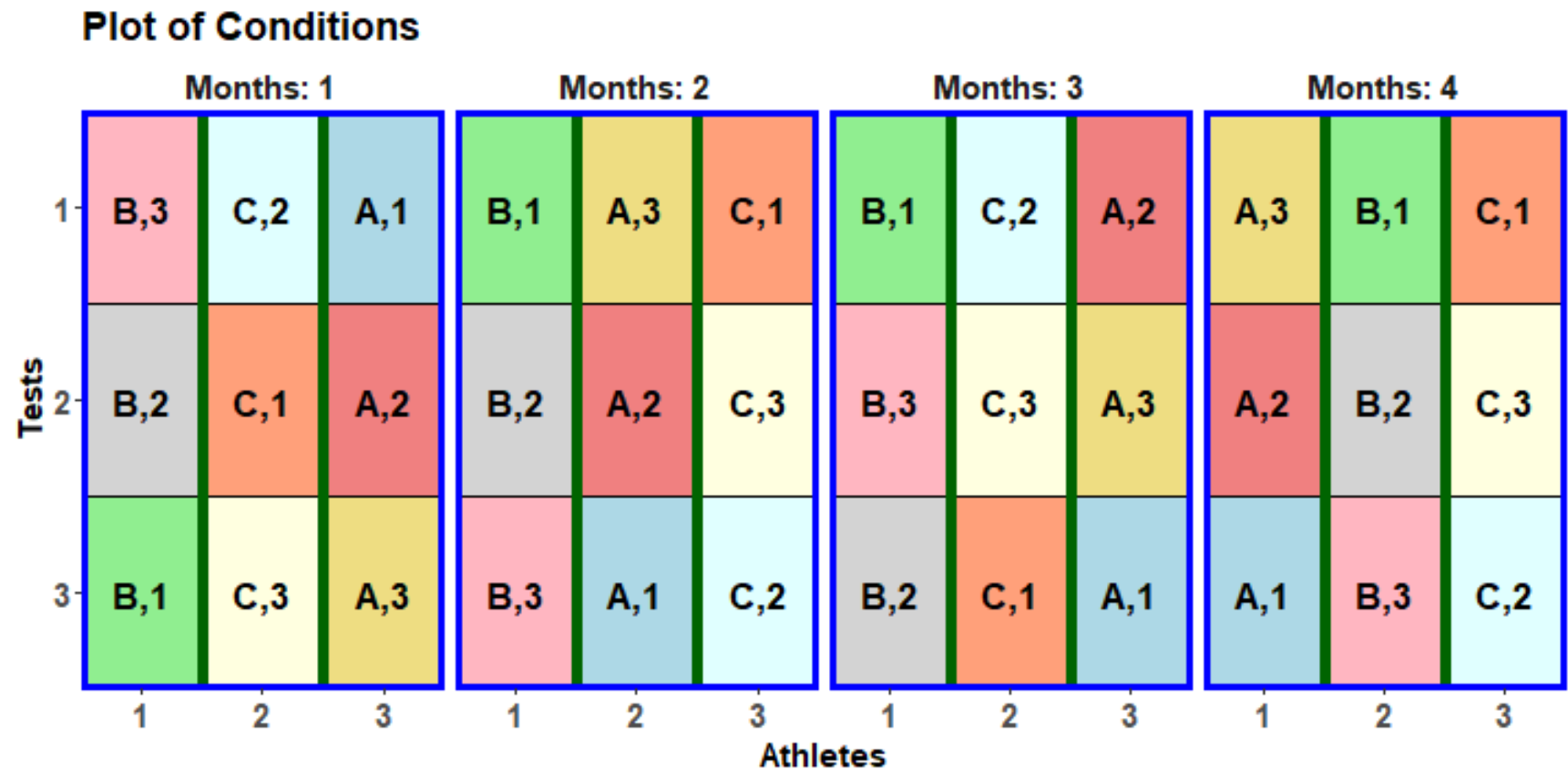
```
> split.sys <- cbind(fac.gen(list(Months = 4, Athletes = 3, Tests = 3)),
+ fac.gen(list(Intensities = LETTERS[1:3], Surfaces = 3),
+ times = 4))
> split.lay <- designRandomize(allocated = split.sys[c("Intensities", "Surfaces")],
+ recipient = split.sys[c("Months", "Athletes", "Tests")],
+ nested.recipients = list(Athletes = "Months",
+ Tests = c("Months", "Athletes")),
+ seed = 2598)
```

Order of levels in **allocated** has to be consistent with **recipient** in order to get a split-unit design.

Note nesting of **Athletes** and **Tests**, dictating that the permutation of **Athletes** will be within **Months** and that of **Tests** will be within **Athletes**.



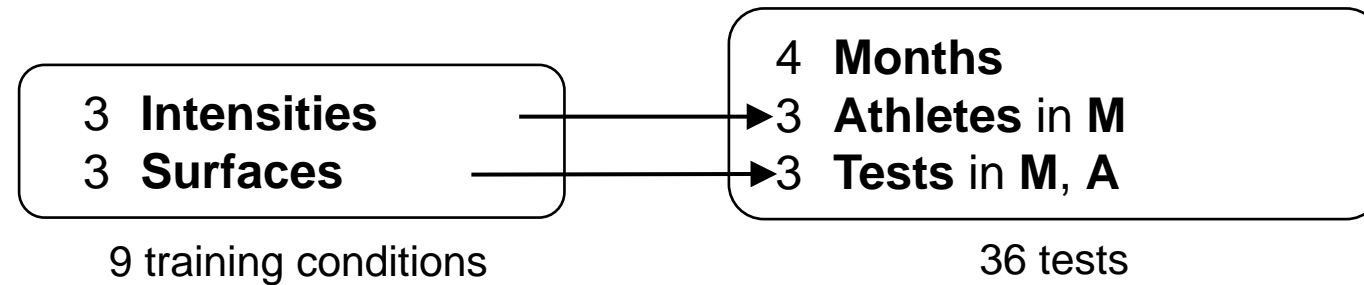
`split.lay`



- Intensities (A, B, C) line up with Athletes and
- Surfaces (1,2,3) differ between Tests,
- both in a random order.

■ With what will Intensities and Surfaces be confounded?

# Working out the confounding



## ■ Terms (omitting Mean):

- Allocated: {Intensities, Surfaces, Intensities:Surfaces}
- Recipient: {Months, Months:Athletes, Months:Athletes:Tests}

## ■ Sources projectors:

- Allocated:  $\{\mathbf{Q}_I, \mathbf{Q}_S, \mathbf{Q}_{I\#S}\}$
- Recipient:  $\{\mathbf{P}_M, \mathbf{P}_{A[M]}, \mathbf{P}_{T[M:A]}\}$

## ■ For each $\mathbf{P}$ , calculate the $\mathbf{PQP}$ products for each of the three $\mathbf{Q}$ s and get their eigenvalues.

- What values do you expect for the eigenvalues of  $\mathbf{P}_{A[M]} \mathbf{Q}_I \mathbf{P}_{A[M]}$ ,  $\mathbf{P}_{A[M]} \mathbf{Q}_S \mathbf{P}_{A[M]}$  and  $\mathbf{P}_{A[M]} \mathbf{Q}_{I\#S} \mathbf{P}_{A[M]}$ ? (Hint: they are either 0 or 1.)

# Using designAnatomy to summarize the confounding

```
> split.canon <- designAnatomy(formulae = list(test = ~ Months/Athletes/Tests,
+ cond = ~ Intensities*Surfaces),
+ data = split.lay)
> summary(split.canon, which.criteria="none")
```

Two formulae, with nesting and crossing corresponding to the factor allocation, and a `data.frame`.

Primary division is according to tests sources.

Summary table of the decomposition for tests & cond

| Source.test            | df1 | Source.cond          | df2 |
|------------------------|-----|----------------------|-----|
| Months                 | 3   |                      |     |
| Athletes[Months]       | 8   | Intensities          | 2   |
|                        |     | Residual             | 6   |
| Tests[Months:Athletes] | 24  | Surfaces             | 2   |
|                        |     | Intensities#Surfaces | 4   |
|                        |     | Residual             | 18  |

Intensities is confounded with Athletes[Months].

Surfaces and Intensities#Surfaces is confounded with Tests[Months:Athletes].

# Prior allocation model

Summary table of the decomposition for test & cond

| Source.test            | df1 | Source.cond          | df2 |
|------------------------|-----|----------------------|-----|
| Months                 | 3   |                      |     |
| Athletes[Months]       | 8   | Intensities          | 2   |
|                        |     | Residual             | 6   |
| Tests[Months:Athletes] | 24  | Surfaces             | 2   |
|                        |     | Intensities#Surfaces | 4   |
|                        |     | Residual             | 18  |

- Probably the same as the initial allocation model, but with Months assumed fixed:
  - Months + Intensities + Surfaces + Intensities:Surfaces | Months:Athletes + Months:Athletes:Tests

# 4. Summary of constructing orthogonal designs

- Straightforward, once the design to use has been chosen.
  - Form a systematic version of the design.
  - Randomize the systematic design, using some randomizing function(s).
  - Check the design.
- This can all be done with **dae** functions:
  - **designRandomize** is a general randomizing function when **recipient** factors form a poset block structure.
    - the levels of all factor combinations, given the nesting relationships, must be equally replicated.
    - e.g. the number of observations (i) per block and (ii) for each Blocks:Plots combination in an RCBD must be equal for (i) all blocks and (ii) for all Blocks:Plots combinations.
  - **designAnatomy** can be used to check the properties of any design, irrespective of the nonorthogonality and the number of tiers.
    - Mistakes will result in nonorthogonality.
    - Slow when the number of observations is large (several hundreds).
- For this, it is necessary to:
  - Divide factors based on allocation of factors (as well as fixed/random).
  - Identify the crossing and nesting, which depends not only on the innate relationships, but also the model needed to describe the anticipated variation.

## 5. Software

(Software and materials at <https://tinyurl.com/BrienWorkshop>)

- R (3.6.x preferable)
- Rstudio (optional)
- Packages:
  - **dae** (Version 3.1-16 or later from CRAN or <http://chris.brien.name/rpackages>)
  - **od** (Version 2.0.0 from <http://mmade.org>)
- **dae**: functions useful in the **d**esign and **a**nova of **e**xperiments (84 functions).
- **od**: generates **o**ptimal experimental **d**esigns for comparative experiments under a general linear mixed model.

# dae: Functions to be used in this workshop

## ii. Factor manipulation functions

- **fac.gen**: generate all combinations of several factors.
- **fac.recode**: recodes the levels and values of a factor.
- **fac.combine**: combines several factors into one.

## iii. Design functions

- **designLatinSqrSys**: Generate a systematic plan for a Latin square design.
- **designRandomize**: Takes a systematic design and randomizes it according to the nesting (and crossing) relationships between the recipient (unit) factors for the randomization.
- **designGGPlot**: A graphical representation of an experimental design using labels stored in a **data.frame** using **ggplot2**.
- **designAnatomy**: Given the layout for a design, produces a **pcanon** object containing its anatomy that shows the confounding and aliasing inherent in the design; obtained via the canonical analysis of the designs projectors.
- **summary.pcanon**: Summarizes the anatomy of a design, being the decomposition of the sample space based on its canonical analysis, as produced by **designAnatomy**. The table produced includes the degrees of freedom and summary statistics of the canonical efficiency factors.
- **efficiencies.pcanon**: Extracts the canonical efficiency factors from a **pcanon.object** produced by **designAnatomy**.

# od

- **od**: generates optimal designs for comparative experiments under a general linear mixed model:
  - based on an *anticipated* mixed model and values for its variance parameters;
  - obtains a design that minimizes the average variance of pairwise differences (AVPD).
- **Functions that will be used:**
  - **od**: Generates optimal designs for comparative experiments under a general linear mixed model.
  - **od.options**: Sets or displays various options that affect the behaviour of **od**.



# Practical session for *Orthogonal experimental design in R*

1. Using `dae` to obtain randomized layouts for orthogonal designs.
  - i. RCBD and LSD
  - ii. Split-unit design for an Oat experiment.
  - iii. Split unit design for an pasture experiment.
2. Except for the last example, you have only to follow the script that has been given.
3. There are some questions for you to answer about each design (answers are in the solutions).

# References

- Box, G. E. P., Hunter, J. S., & Hunter, W. G. (2005). *Statistics for experimenters: design, innovation, and discovery* (2nd ed.).
- Brien, C. J. (2017). Multiphase experiments in practice: A look back. *Australian & New Zealand Journal of Statistics*, **59**, 327-352.
- Brien, C. J. (2019). *dae: functions useful in the design and ANOVA of experiments*. R package version 3.1-16. URL <http://cran.at.r-project.org/package=dae>.
- Brien, C. J., & Demétrio, C. G. B. (2009). Formulating Mixed Models for Experiments, Including Longitudinal Experiments. *Journal of Agricultural, Biological, and Environmental Statistics*, **14**, 253-280.
- Brien, C. J., Harch, B. D., Correll, R. L., & Bailey, R. A. (2011). Multiphase experiments with at least one later laboratory phase. I. Orthogonal designs. *Journal of Agricultural, Biological, and Environmental Statistics*, **16**, 422-450.
- Cochran, W. G., & Cox, G. M. (1957). *Experimental Designs* (2nd ed.). New York: Wiley.
- Kiefer, J. (1959). Optimum Experimental Designs. *Journal of the Royal Statistical Society, Series B (Methodological)*, **21**, 272-319.
- Peeling, P., Dawson, B., Goodman, C., Landers, G., Wiegerinck, E. T., Swinkels, D. W., & Trinder, D. (2009). Training Surface and Intensity: Inflammation, Hemolysis, and Hepcidin Expression. *Medicine & Science in Sports & Exercise*, **41**, 1138-1145.
- Wilkinson, G. N., & Rogers, C. E. (1973). Symbolic description of factorial models for analysis of variance. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, **22**, 392-399.