Basic Blocking

Gary W. Oehlert

School of Statistics
University of Minnesota

October 16, 2014

Variance Reduction Design

We now begin a new phase of the course where we move away from completely randomized designs.

Power increases and margin of error decreases if N is larger or if σ^2 is smaller.

It costs resources to make N bigger.

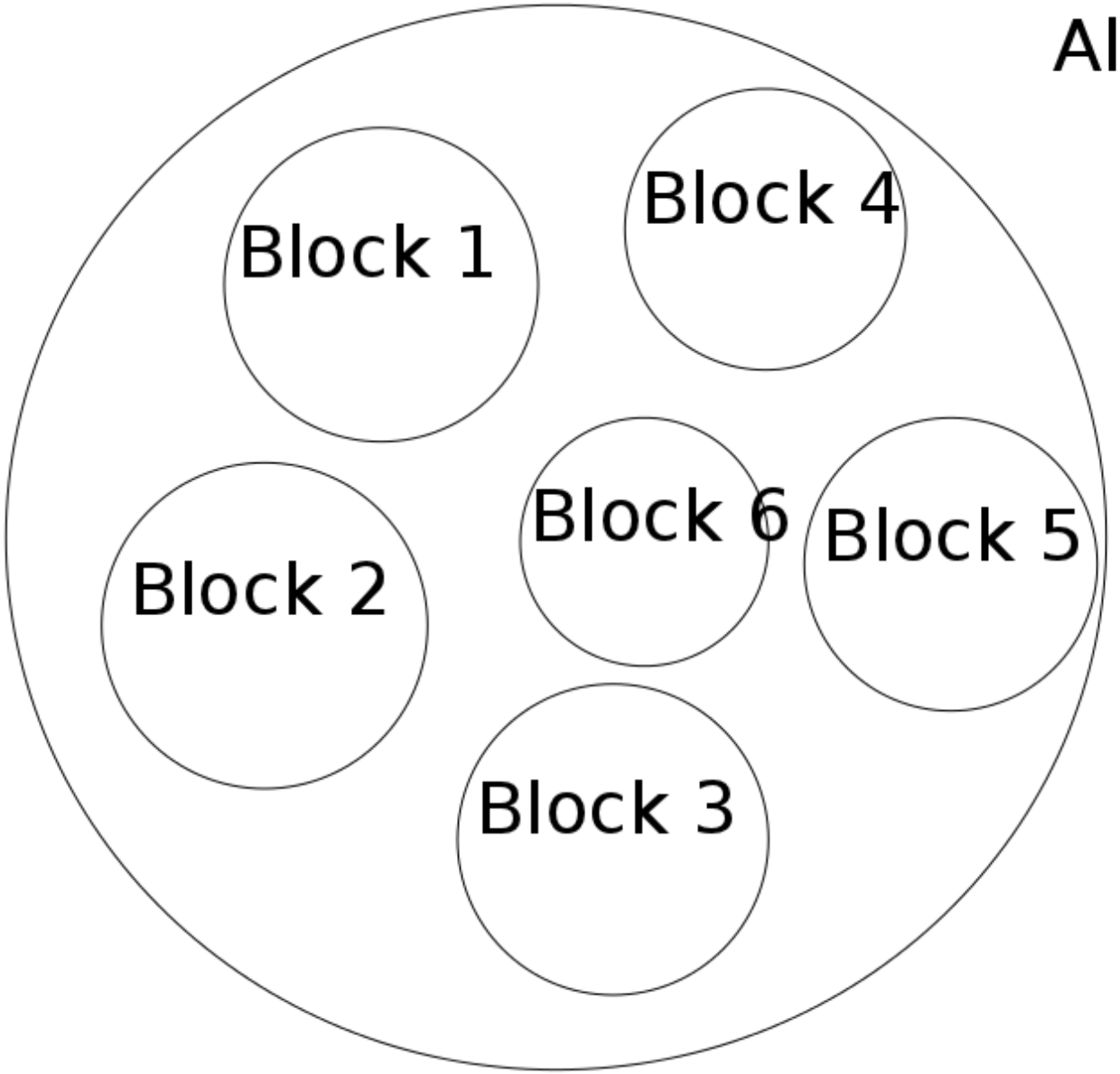
This part of the course is about making σ^2 smaller.

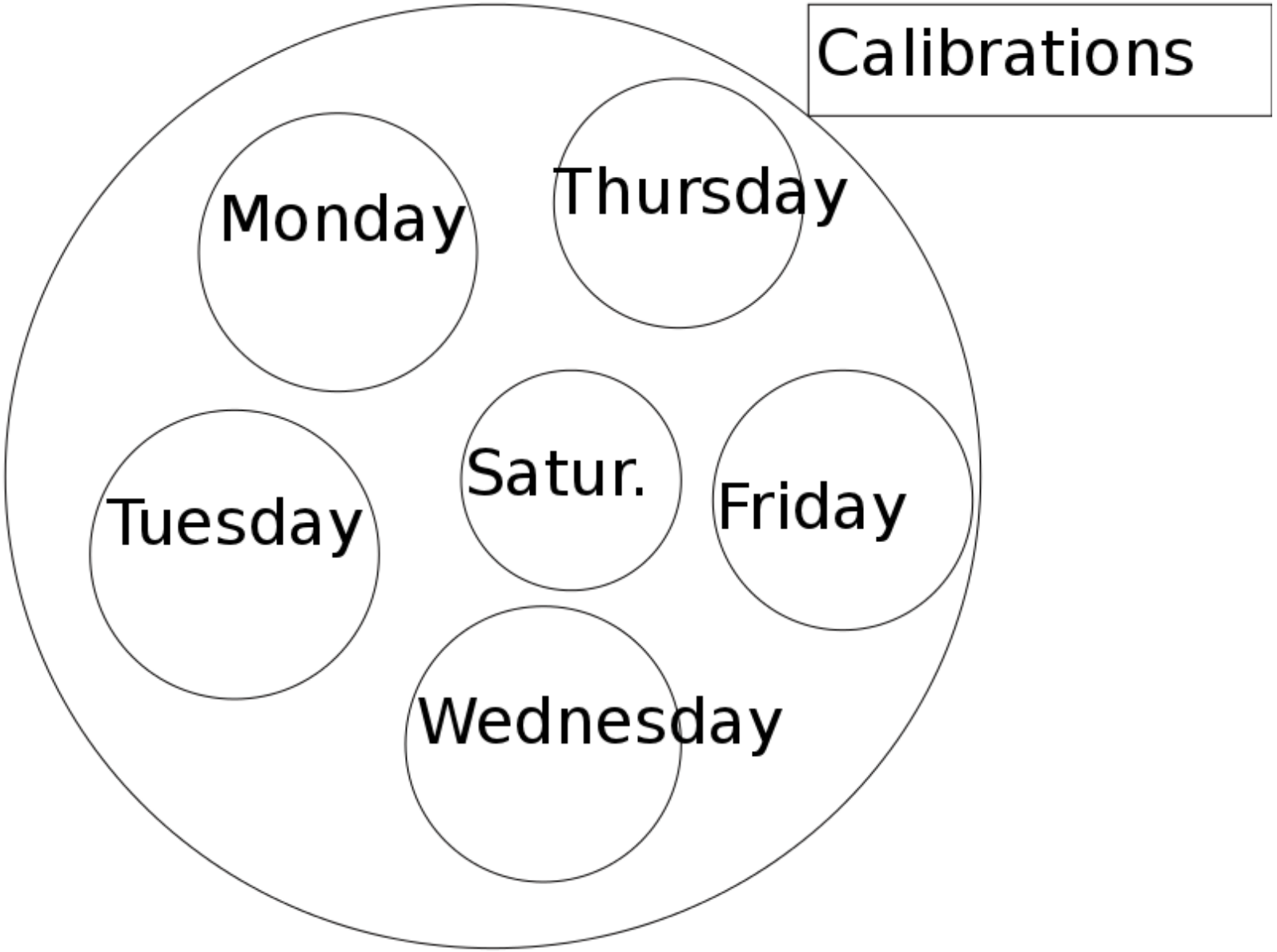
Blocking

The principal tool in variance reduction design is blocking.

A block is a homogeneous subset of units. Prior to running the experiment, we know these units are similar in some way that we expect will make them likely to have similar responses.

Similar soil, similar instrument calibration, similar batch of raw material, similar operators, similar genetics, similar environmental conditions, similar socio-economic background, similar something or other.


All units
Block 1
Block 4
Block 2
Block 6
Block 5
Block 3


Calibrations
Monday
Thursday
Tuesday
Satur.
Friday
Wednesday

Blocking will be some form of repeating the experiment (or part of it) separately and independently in each block.

This restricts the randomization of treatments to units.

With apologies to Woodward, Bernstein, and Deep Throat:

Follow the randomization!

Different designs correspond to different randomizations, and examining the randomization can allow you to discern the design.

Randomized Complete Block design

The RCB is the progenitor of all block designs.

We have:

- g treatments
- g units per block
- r blocks
- $rg = N$ total units

Within each block, randomly and independently assign the g treatments to the g units.

It's like r single-replication CRDs glued together.

Notes:

- This is a complete block design because every treatment occurs in every block.
- The treatments could have factorial structure.
- Consider blocking when you can identify a source of variability prior to experimentation.
- Blocking is done at the time of randomization; it is not imposed later.
- The randomization in an experiment could identify it as RCB.

Model:

$$y_{ij} = \mu = \alpha_i + \beta_j + \epsilon_{ij}$$

$$i = 1, \ldots, g; \quad j = 1, \ldots, r$$

We think that units in some blocks might respond high, units in others might respond low, but within a block units are more homogeneous (less variable) than randomly chosen units from the universe of units.

For a two-factor treatment design, we would use the model:

$$y_{ijk} = \mu = \alpha_i + \beta_j + \alpha\beta_{ij} + \gamma_k + \epsilon_{ijk}$$

$$i = 1, \ldots, a; \quad j = 1, \ldots, b; \quad k = 1, \ldots, r$$

The model assumes that treatments have the same effect in every block, i.e., treatments and blocks are additive.

Assuming additivity does not make additivity true; transformation of the response can sometimes improve additivity.

Because there is only a single observation for each treatment in each block, we cannot distinguish between random error and any potential interaction between treatments and blocks.

From a practical perspective, it doesn't matter much whether we think of blocks as fixed or random.

From a theoretical perspective, blocks are probably random in most situations.

Why does RCB work? Here are a couple points of view.

Make comparisons within blocks (thus small variance), and then combine across blocks.

Treatment totals all contain the same block totals, so block effects cancel out when comparing treatment totals (and similarly for treatment averages).

Block to block variability is still in the totality of variability in the data, but we contrive to make it disappear when comparing treatments.

Do not test blocks!

Note that blocks are in the nature of the units. There ought to be big differences between blocks.

We did not assign blocks to units. Blocks were not randomly assigned, so there is no randomization test for blocks.

The software does not know any better than to test blocks, but now you do!

For unbalanced data, always look at treatments adjusted for blocks (i.e., blocks are always in the base model for any treatment factor). For balanced data, blocks and treatments are orthogonal.

Relative efficiency

How well did blocking work? Should we use blocking in our next similar experiment?

"Testing" the block effect is not what matters. What matters is how large the error variance would have been if we had not blocked.

Relative efficiency answers the following question: using the same universe of units, by what factor would we need to increase our sample size to get the same power in a CRD that we would achieve using the RCB?

This is mostly an issue of how the error variance changes, but there is also a minor effect due to the fact that fitting blocks uses up degrees of freedom for error.

(In CRD, bigger σ^2 hurts, but larger df_{error} helps; usually the first factor dominates.)

We first estimate what error variance would have been if we had used a CRD instead of RCB, then we make a minor df adjustment.

We estimate σ^2_{RCB} by MSE or residual variance in the RCB analysis.

We estimate σ^2_{CRD} via

$$\begin{aligned}\widehat{\sigma}^2_{CRD} &= \frac{df_{blocks}MS_{blocks} + [df_{Trt} + df_{error}]MS_E}{df_{blocks} + df_{Trt} + df_{error}} \\ &= \frac{(r-1)MS_{blocks} + [(g-1) + (r-1)(g-1)]MS_E}{(r-1) + (g-1) + (r-1)(g-1)}\end{aligned}$$

This is an average of MSE and MSBlock weighted by df, but we use df error plus df treatments as the weight for MSE. Typically this estimate is less than the MSE you would get if you just left blocks out of the model.

The df adjustment is less obvious. Let $\nu_{RCB} = (r-1)(g-1)$ be the error df in the RCB analysis. Let $\nu_{CRD} = rg - g$ be the error df if you had not blocked.

The estimated relative efficiency of RCB to CRD is

$$E_{RCB:CRD} = \frac{\nu_{RCB}+1}{\nu_{RCB}+3} \frac{\nu_{CRD}+3}{\nu_{CRD}+1} \frac{\widehat{\sigma}^2_{CRD}}{\widehat{\sigma}^2_{RCB}}$$

If this ratio is 1.7, then you would need 1.7 times as many units in a CRD to achieve the same power as an RCB.

Latin Squares

An RCB is an effective way to block on one source of extraneous variation; what if you have two sources of extraneous variation?

Light and drainage in garden flower trials; gender and blood pressure in cardiac trials; driver and environmental conditions in MPG trials

Think back to why RCB designs work; we want to get that cancellation of block effects to happen simultaneously for two blocking factors.

The Latin Square design is the classical design for blocking on two sources of variation.

There are g^2 units visualized as a square. Those units in the same row are all in the same block based on the first extraneous source of variation. Those units in the same column are all in the same block based on the second extraneous source of variation.

The g treatments are randomized so that each treatment occurs once in each row and once in each column.

Treatments are represented by Latin letters, thus Latin Squares.

A	B	C	D
B	A	D	C
C	D	A	B
D	C	B	A

If you ignore columns, a Latin Square is an RCB in rows. If you ignore rows, a Latin Square is an RCB in columns.

Randomization is often done more like this. Take a square from a table of squares (back of the book). Randomly permute the rows and the columns. Randomly assign treatments to the letters. Not as random as the "randomize subject to" description, but generally good enough and a lot simpler.

Model:

$$y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_k + \epsilon_{ijk}$$

i, j, and k all run from 1 to g.

Note: we only observe g^2 of the g^3 i, j, k combinations.

We are again assuming additivity in a major way, and we might need to transform to achieve additivity.

For unbalanced data, treatments adjusted for all blocking factors.

What if we need more data to achieve acceptable power?

In addition, if you think of rows and columns as fixed effects, we have $(g-1)(g-2)$ degrees of freedom for error. That might not be very many.

Latin Squares are often replicated, i.e., we use more than one square with the same set of treatments.

However, we need to consider how the replication is done.

Suppose that we have r squares to study g treatments. All squares will have row blocks and column blocks. The issue is whether the squares have the same row blocks or different row blocks; similarly for columns.

Example: three squares ($r = 3$) for $g = 4$ treatments; 48 total units
Treatments: gasoline additives
Response: particulate emissions
Row blocks: drivers
Column blocks: cars

Option 1: every square uses different drivers and different cars.

$$y_{ijk\ell} = \mu + \alpha_i + \beta_j + \gamma_{k(j)} + \delta_{\ell(j)} + \epsilon_{ijk\ell}$$

This is mean plus treatment plus square plus car-nested-in-square plus driver-nested-in-square plus error.

There are (g-1) df for treatments, (r-1) df for squares, r(g-1) df for cars within square, r(g-1) df for drivers within square.

Drivers 1–4 \ Cars 1–4				
A	A	B	C	D
B	B	A	D	C
C	C	D	A	B
D	D	C	B	A

Drivers 5–8 \ Cars 5–8				
A	A	B	C	D
B	B	C	D	A
C	C	D	A	B
D	D	A	B	C

Drivers 9–12 \ Cars 9–12				
D	D	C	A	B
A	A	B	D	C
C	C	D	B	A
B	B	A	C	D

Option 2: every square uses the same cars but different drivers.

$$y_{ijk\ell} = \mu + \alpha_i + \beta_j + \gamma_k + \delta_{\ell(j)} + \epsilon_{ijk\ell}$$

This is mean plus treatment plus square plus car plus driver-nested-in-square plus error.

There are (g-1) df for treatments, (r-1) df for squares, (g-1) df for cars, r(g-1) df for drivers within square.

	Cars 1–4			
Drivers 1–4	A	B	C	D
	B	A	D	C
	C	D	A	B
	D	C	B	A
Drivers 5–8	A	B	C	D
	B	C	D	A
	C	D	A	B
	D	A	B	C
Drivers 9–12	D	C	A	B
	A	B	D	C
	C	D	B	A
	B	A	C	D

Option 3: every square uses the same drivers but different cars.

$$y_{ijk\ell} = \mu + \alpha_i + \beta_j + \gamma_{k(j)} + \delta_\ell + \epsilon_{ijk\ell}$$

This is mean plus treatment plus square plus car nested in square plus driver plus error.

There are (g-1) df for treatments, (r-1) df for squares, r(g-1) df for cars within square, (g-1) df for drivers.

Drivers 1–4	Cars 1–4			
	A	B	C	D
	B	A	D	C
	C	D	A	B
	D	C	B	A

Cars 5–8			
A	B	C	D
B	C	D	A
C	D	A	B
D	A	B	C

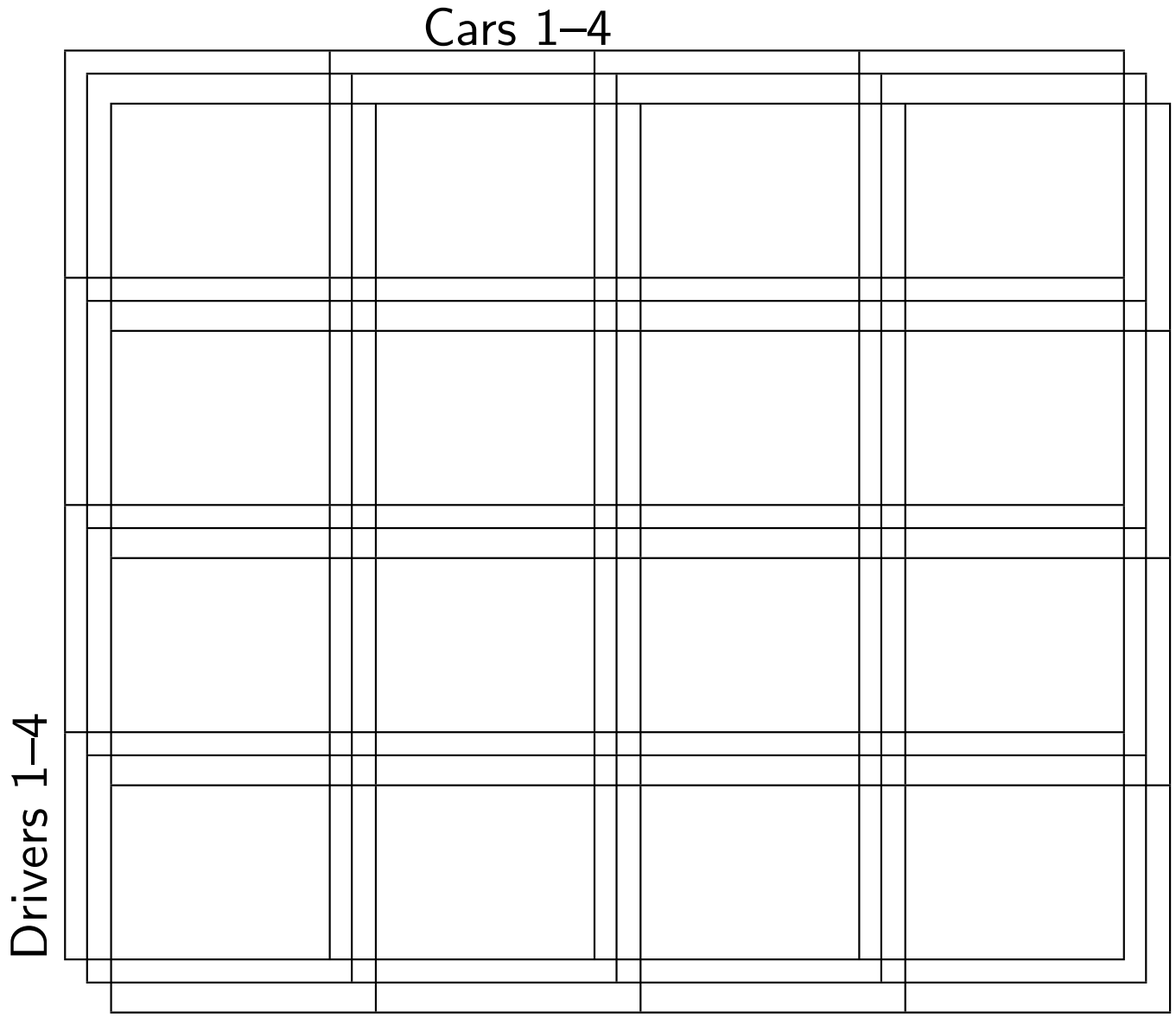
Cars 9–12			
D	C	A	B
A	B	D	C
C	D	B	A
B	A	C	D

Option 4: every square uses the same drivers and the same cars.

$$y_{ijk\ell} = \mu + \alpha_i + \beta_j + \gamma_k + \delta_\ell + \epsilon_{ijk\ell}$$

This is mean plus treatment plus square plus car plus driver plus error.

There are (g-1) df for treatments, (r-1) df for squares, (g-1) df for cars, (g-1) df for drivers.


Cars 1–4
Drivers 1–4

A very common example is the cross over design.

In a cross over, one of the blocking factors is time period, and the other blocking factor is subject. Each subject has each treatment, but some get one treatment first, others have another treatment first, and so on.

To replicate these designs, we generally get a new set of subjects, but the period effects are assumed to be the same for all squares.

We can also compute the relative efficiency of a Latin Square relative to an RCB should we consider not using one of the blocking factors. For example, if we consider not using rows, then

We estimate σ^2_{LS} by MSE in the Latin Square analysis.

We estimate σ^2_{RCB} via

$$\widehat{\sigma}^2_{RCB} = \frac{df_{rows} MS_{rows} + [df_{Trt} + df_{error}] MS_E}{df_{rows} + df_{Trt} + df_{error}}$$

Let ν_{LS} be the error df in the LS analysis. Let ν_{RCB} be the error df if you had not blocked on rows.

The estimated relative efficiency of LS to RCB is

$$E_{LS:CRD} = \frac{\nu_{LS}+1}{\nu_{LS}+3}\,\frac{\nu_{RCB}+3}{\nu_{RCB}+1}\,\frac{\widehat{\sigma}^2_{RCB}}{\widehat{\sigma}^2_{LS}}$$

If this ratio is 1.7, then you would need 1.7 times as many units if you had run a RCB instead of an LS.

Generalizations

There are many possible generalizations of these blocking designs.

The Generalized Randomized Complete Block design is analogous to an RCB except each block has 2g or 3g etc. units and each treatment is assigned to 2 or 3 etc. units within each block.

In this case, the standard approach is to model blocks as random and include a random block by treatment interaction term.

The carry over design, or design balanced for residual effects, is a Latin Square where, in addition to the usual requirements, we also have that each treatment follows each other treatment exactly once.

This is useful when one of the blocking factors is time period, and the effect of one treatment could carry over into the next time period.

For example, a toxic drug might not only suppress the response in the period where it is given, it could also suppress the response in the following period.

The model contains an additional factor with g+1 levels "follows treatment 1" up to "follows treatment g" and the final level of "used first."

If you have three blocking factors, then you can use a Graeco-Latin square.

Latin letters are treatments, Greek letters indicate third blocking factor. Each treatment occurs once in each row, once in each column, and once with each Greek letter.

A α	B γ	C δ	D β
B β	A δ	D γ	C α
C γ	D α	A β	B δ
D δ	C β	B α	A γ

No 6 by 6 GL square.

Model has additive treatment and (three) blocking factors.

Incomplete Blocks

Complete block designs like RCB and LS are set up with every treatment occurring in every block.

Sometimes, there are only k units in a block, and $k < g$. Then we must use an incomplete block design.

Three different eye drops to study relief from irritation. There is large subject to subject variability, so block on subject, but only two eyes per subject.

Six different processes for extracting avocado oil. There is large fruit to fruit variability, so block on fruit, but each fruit is only large enough to test four processes.

Incomplete block designs are inherently less efficient than complete block designs on a per unit basis with equal variances. Example:

A B	A C	C B

versus

A B C	A B C

In the complete block, unit 1 - unit 2 and unit 4 - unit 5 both estimate A - B and both have variance $2\sigma^2_{comp}$.

In the incomplete block, unit 1 - unit 2 and unit 3 - unit 4 + unit 5 - unit 6 both estimate A - B and have variances $2\sigma^2_{incomp}$ and $4\sigma^2_{incomp}$

We prefer complete blocks if $\sigma^2_{comp} = \sigma^2_{incomp}$, but often $\sigma^2_{comp} > \sigma^2_{incomp}$, and that can be where incomplete blocks are preferred.

For example, suppose fruit to fruit variance in oil concentration is 10, but quarter to quarter within a fruit variance is 1.

The relative efficiency of a BIBD with six treatments in blocks of size four is .9.

Without blocking, variance of a pairwise difference in means is $10(\frac{1}{n} + \frac{1}{n}) = \frac{20}{n}$.

With a BIBD, variance of a pairwise difference in means is $1(\frac{1}{.9n} + \frac{1}{.9n}) = \frac{2.22}{n}$.

Here the reduced variance achievable with the BIBD overcomes the loss due to relative efficiency of the BIBD to the RCB.

Balanced Incomplete Block design

The basic prototype of all incomplete block designs is the BIBD. Here we have:

- g treatments
- b blocks
- k units per block
- each treatment used r times
- bk total units
- bk = rg

In addition, each pair of treatments occurs together in the same number of blocks. $\lambda = r(k-1)/(g-1)$.

Both have g=3, k=2, r=2, b=3, but left side is BIBD and right side is not.

Notes:

- If $\lambda = r(k-1)/(g-1)$ is not a whole number, then no BIBD for that set of parameters.
- Treatments could have factorial structure.
- Also balanced in the sense that variance of $\hat{\alpha}_i - \hat{\alpha}_j$ does not depend on i,j.
- A BIBD always exists for any $g > k$ pair; simply take all combinations. There are tables for smaller values of b and r.
- Randomize by randomly assigning the treatments to the treatment "numbers," and then randomly assigning treatment numbers to units within their blocks.

Model:

$$y_{ij} = \mu + \alpha_i + \beta_j + \epsilon_{ij}$$

For RCB, it didn't really matter if blocks were fixed or random. For BIBD, it does matter.

If we assume blocks are fixed, we get the intrablock analysis. All estimates are based on differences from within blocks.

If we assume blocks are random, then there is some information about the treatments in block totals; this leads to the interblock recovery analysis.

Interblock recovery provides slightly more precise estimates in cases where it is appropriate (i.e., when blocks are random), but large block variance relative to units-within-blocks variance means the improvement is often negligible.

Interblock recovery used to be a complicated process (old school), but with lmer(), interblock recovery is not much extra effort.

Intrablock is just treatments adjusted for fixed blocks; let R do the work.

If you could do RCB with same variance as BIBD, then

$$E_{BIBD:RCB} = \frac{g(k-1)}{(g-1)k}$$

is the relative efficiency. The effective sample size is rE

$$Var(\sum_i w_i \hat{\alpha}_i) = \sigma^2 \sum_i \frac{w_i^2}{rE}$$

$$SS_{Trt} = \sum_i rE\hat{\alpha}_i^2$$

Interblock recovery is a combination of the intrablock estimates with estimates based on regressing block totals on the treatments appearing in blocks—the raw interblock estimates.

The error variance in this regression is a combination of σ^2_{incomp} plus the block to block variance (which is often much bigger than σ^2_{incomp}). Raw interblock estimates have greater variability than intrablock estimates, often much greater.

Interblock recovery combines these two, but the combination is usually pretty close to the intrablock estimates.

If you assume random blocks and use lmer, you get the interblock recovery analysis.

Youden Squares

These are always amusing, because Youden squares are not square.

Consider a situation with two blocking factors, but one of the factors can only have blocks of size $g-1$ instead of g.

Take a Latin square and delete one row (or column) obtaining a g by g-1 arrangement.

This is a Youden square. It is RCB for one blocking factor and BIBD for the other blocking factor.

A	B	C	D
B	A	D	C
C	D	A	B

Intrablock analysis is just treatments after both blocking factors.

You can do interblock recovery if the short (incomplete) blocks are random.

Other incomplete block designs

There are many other kinds of incomplete block designs (including only designs in text):

- Partially balanced incomplete block designs
- Cyclic designs
- Lattice designs
- Alpha designs

Most of these are motivated by trying to get good properties from a smaller design than a BIBD. E.g., the smallest BIBD with g=12 and k=7 has 132 blocks.

Most of these have N=gr=bk but relax the equal pair occurrence requirement of BIBD in some way.

PBIBD was an early competitor. Treatments are in associate classes, e.g., 2 classes.

Pairs of treatments that are first associates occur together λ_1 times; second associates occur together λ_2 times. All treatments have the same number of first associates and second associates.

In addition, pick a pair of ith associates and let ρ^i_{jk} be the number of treatments that are jth associates of one member of the pair and kth associates of the other member of the pair. This number cannot depend on the original pair chosen.

Evidently, generating or verifying a PBIBD is a bit fiddly, and some of the later designs are competitors based on ease of construction.

From a data analysis perspective, we still do intrablock as treatments adjusted for fixed block effects, and we can get interblock recovery if we have random block effects.

The difference is that there are "simple" formulae allowing hand calculations for BIBD, and the formulae get progressively less simple or non-existent as we move to more complex designs. In the end, it's all R anyway.