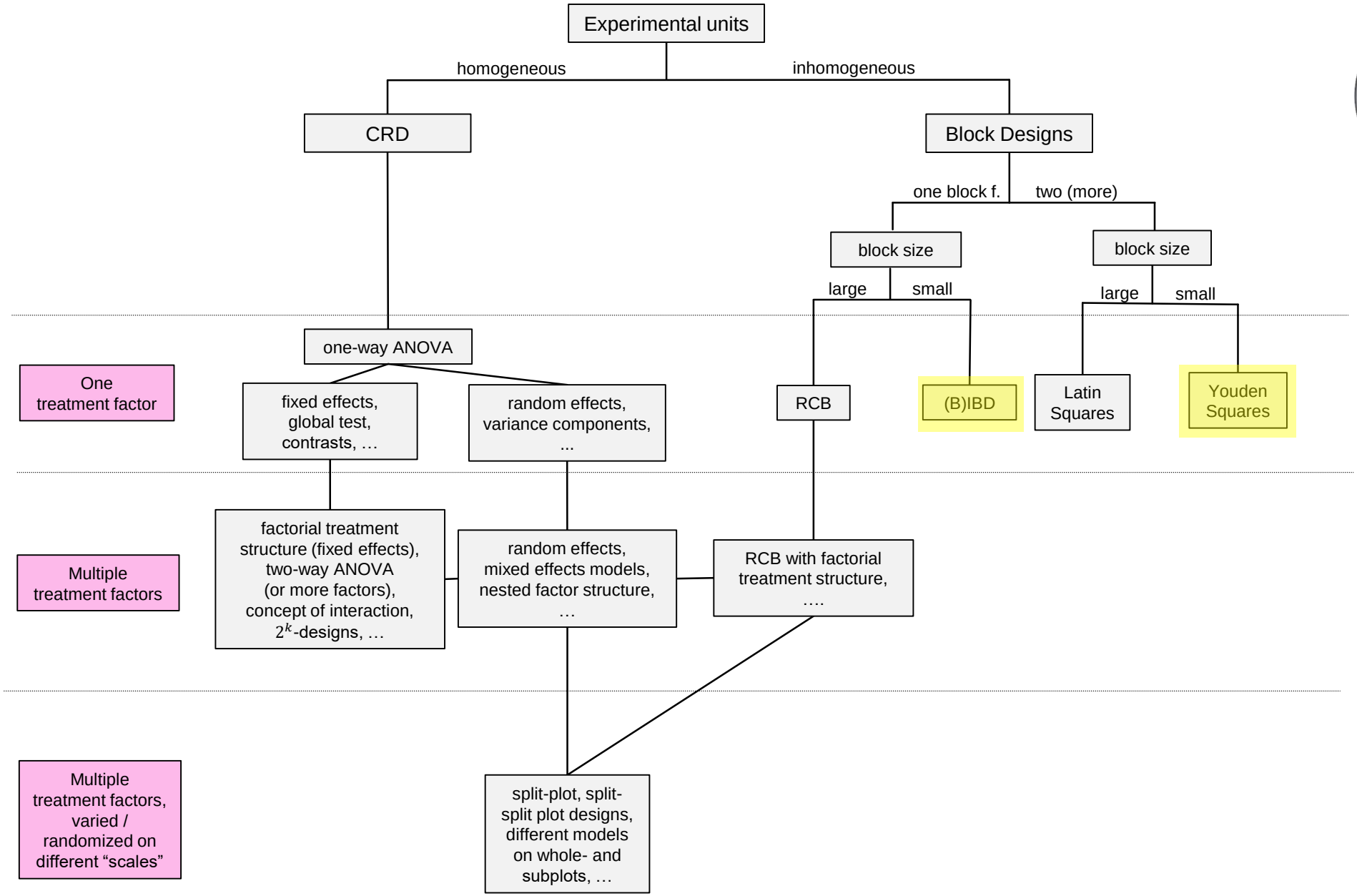
11

Incomplete Block Designs



Similar to Lawson (2015)

Incomplete Block Designs

- Up to now we only considered **complete** block designs.
- This means we see **all** treatments in **each** block.
- In some situations this is **not** possible because
  - (physical) block size too small
  - too expensive
  - not advisable (think of raters having to rate 7 or more champagne brands each)
- Remember the eye drop example? What if we wanted to test **3** different eye drop types?
- It is still a good idea to **block** on subjects, but obviously it is **not** possible to have **complete** blocks in this example!

Example: Eye Drops (Oehlert, 2010)

- Suppose we have 3 subjects getting the treatments A, B, C:

Subject 1	Subject 2	Subject 3
A	A	B
B	C	C

- This is a so-called **incomplete** block design.
- If we want to estimate the difference between A and B, we can use:
  - Subject 1: The estimate has variance $2\sigma^2$.
  - Combine subject 2 and subject 3:

$$A - B = (A - C) - (B - C)$$

  This difference of differences has variance $2\sigma^2 + 2\sigma^2 = 4\sigma^2$.
- In a **complete** block design, we could estimate the difference in **each** block with the **same precision**.

Incomplete Block Designs

- We have to be careful about which treatment pairs we put together in the **same** block.
- We call a design **disconnected** if we can build two groups of treatments such that it **never** happens that we see members of both groups together in the **same block**.

1	2	3	4	5	6
A	A	B	D	D	E
B	C	C	E	F	F

- In a **disconnected** design, it is **not** possible to estimate all treatment differences!

- If the design is **not** disconnected, we call it **connected**.

Balanced Incomplete Block Designs (BIBDs)

- We call an incomplete block design **balanced** (**BIBD**) if all treatments **pairs** occur together in the same block **equally often** (we denote this number by λ).
- What is the benefit of the “balancedness” property?
- The **precision** (variance) of the estimated treatment differences $\alpha_i - \alpha_j$ is the **same** no matter what combination of i and j we are considering.
- This means that we can estimate all treatment differences with the **same accuracy**.
- Let us first give an overview of the different numbers involved in such a problem.

Balanced Incomplete Block Designs (BIBDs)

- We use the following notation:
  - g: number of treatments
  - b: number of blocks
  - k: number of units per block with $k < g$
  - r: number of replicates per treatment
  - N: total number of units

- In the eye drop example we had
  - $g = 3$ treatments (the different eye drops: A, B, C)
  - $b = 3$ blocks (the 3 subjects)
  - $k = 2$ units per block (the 2 eyes per subject)
  - $r = 2$ replicates per treatment
  - $N = 6$

- Of course it must hold that $N = b \cdot k = g \cdot r$.

Unreduced BIBDs

- We can always find a BIBD for every setting of $k < g$.
- How? Simply use **all** possible combinations.
- The number of combinations is $\binom{g}{k}$ (= binomial coefficient: $\frac{g!}{k!(g-k)!}$).
- E.g., for $g = 7$ and $k = 3$ we have $\binom{7}{3} = 35$.
- In R, have a look at the functions `choose` and `combn`.
- We call such a design an **unreduced** balanced incomplete block design.
- In practice, it is often **not** possible to have so many blocks.
- The big question: What number of blocks is “doable”?

Balanced Incomplete Block Designs (BIBDs)

- A treatment occurs in r blocks.
- There are $k-1$ other “available units” in each of these blocks which makes a total of $r \cdot (k-1)$ “available units”.
- The remaining $g-1$ treatments must be divided **evenly** among them, otherwise the design is **not** balanced.
- Hence, $\frac{r \cdot (k-1)}{g-1}$ **must** be a **whole number** $(= \lambda)$ for a BIBD to exist.
- This condition is only **necessary**, **not sufficient**.
- This means: Even if the condition is fulfilled, it might be the case that you **cannot** find a BIBD!

Example: Champagne (Roth, 2013)

- 14 raters, 7 champagne types, every rater rated 3 of them.

1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	1	2	2	1	3	1	1	3	3	1	1	4	2
6	3	6	4	2	5	4	2	4	5	4	5	5	3
7	6	7	5	3	7	7	5	6	7	7	6	6	4

- This is a BIBD. We see every treatment pair combination exactly **twice** in the same block.
- In more detail, we have
  - $g = 7$ treatments
  - $b = 14$ blocks
  - $k = 3$ units per block
  - $r = 6$ replicates per treatment.
- Hence, $\lambda = \frac{r \cdot (k-1)}{g-1} = \frac{6 \cdot 2}{6} = 2$.

BIBD: Finding a Design

- First, make sure that the necessary condition is fulfilled.
- Old way: Check Appendix C.2 of the book with a list of BIBDs.
- Use R, e.g. function `find.BIB` in package `crossdes` or function `(b)ibd` in package `ibd` (among many others).
- See R-File for an example.

(B)IBD: Randomization

- How can we randomize a given (B)IBD?
- Randomize blocks to the groups of treatment letters.
- Within each block: Randomize assignment of treatment letters to physical units.
- Randomize assignment of treatment letters to actual treatments.
- How can we analyze an incomplete block design?

(B)IBD: Analysis

- The **model** for a (balanced) incomplete block design is the standard model, i.e.

$$Y_{ij} = \mu + \alpha_i + \beta_j + \varepsilon_{ij}$$

effect of **treatment**

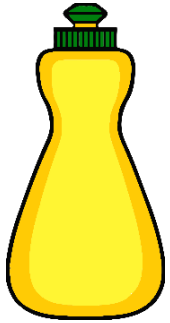
effect of **block**

- However, as we don't observe all treatment × block combinations, the "usual" estimates are **not** working and we need the computer to find the least squares estimates (which is no problem).
- We are using Type III sum of squares to test treatment effects **adjusted for block effects**.
- In other words: We analyze the treatment effects while **controlling for the block effects**.

Intra- and Interblock Analysis

- This is a so called **intrablock analysis of** the (B)IBD.
- It is also possible to recover some information by comparing different blocks.
- This would be called an **interblock analysis**.
- Information from both approaches can be suitably combined.
- This looks complicated in the book, but it is nothing else than the analysis when treating the block factor as **random**.
- We will **not** discuss this any further here.

Example: Dish Detergent (Oehlert, 2010, Ex. 14.2)



- Want to compare **9 different dishwashing solutions**.

Treatment	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*J*
Base detergent	*I*	*I*	*I*	*I*	*II*	*II*	*II*	*II*	control
Additive	3	2	1	0	3	2	1	0	control

- Available resources:
  - 3 washing basins
  - 1 operator for each basin (= 3 operators)
- The 3 operators wash at the same speed during each session, but the **speed might vary from session to session**.
- Response: **Number of plates washed** when foam disappears.

Example: Dish Detergent (Oehlert, 2010, Ex. 14.2)

- If we have 12 sessions, we can find a BIBD.
- The design was as follows:

1	2	3	4	5	6	7	8	9	10	11	12
A	*D*	*G*	*A*	*B*	*C*	*A*	*B*	*C*	*A*	*B*	*C*
B	*E*	*H*	*D*	*E*	*F*	*E*	*F*	*D*	*F*	*D*	*E*
C	*F*	*J*	*G*	*H*	*J*	*J*	*G*	*H*	*H*	*J*	*G*

- Analysis in R:

```
> fit <- aov(dishes ~ session + detergent, data = dish)
> drop1(fit, test = "F")
Single term deletions

Model:
dishes ~ session + detergent
          Df Sum of Sq     RSS     AIC  F value    Pr(>F)
<none>                  13.19   3.841
session   11     10.06   23.25   2.260   1.1103    0.4127
detergent  8   1086.81 1100.00 147.104 164.8539 6.809e-14 ***
```

Example: Dish Detergent (Oehlert, 2010, Ex. 14.2)

- If we call `summary.lm`, we get

```
> summary.lm(fit)

Call:
aov(formula = dishes ~ session + detergent, data = dis

Residuals:
    Min      1Q  Median      3Q     Max
-1.1482 -0.5556  0.1111  0.4630  1.0000

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  18.7037     0.6766  27.643 6.21e-15 ***
session10     1.4074     0.8194   1.718 0.105170
session11     0.6296     0.8194   0.768 0.453458
session12     0.8519     0.8194   1.040 0.313998
session2      1.1111     0.8559   1.298 0.212612
session3      0.4444     0.8559   0.519 0.610667
session4      0.9259     0.8194   1.130 0.275148
session5      1.1481     0.8194   1.401 0.180266
session6      2.1481     0.8194   2.622 0.018513 *
session7      1.8519     0.8194   2.260 0.038127 *
session8      0.6296     0.8194   0.768 0.453458
session9      1.4074     0.8194   1.718 0.105170
detergent2   -2.5556     0.7412  -3.448 0.003309 **
detergent3   -6.5556     0.7412  -8.844 1.47e-07 ***
detergent4  -13.2222     0.7412 -17.839 5.54e-12 ***
detergent5    5.5556     0.7412   7.495 1.28e-06 ***
detergent6    3.2222     0.7412   4.347 0.000499 ***
detergent7    1.3333     0.7412   1.799 0.090928 .
detergent8   -0.5556     0.7412  -0.750 0.464416
detergent9    9.7778     0.7412  13.192 5.16e-10 ***
```

Here we used `contr.treatment`. The coefficients are therefore comparisons to the **reference treatment** (= detergent 1). Note that the standard error is the same for all effects which is a property of the balanced design.

Partially Balanced Incomplete Block Designs

- It might very well be the case that we are in a situation where there is **no** BIBD available.
- In that case we could use a **partially balanced incomplete block design**, where some treatment pairs occur together more often than other pairs.
- Example (Kuehl, 2000, Display 9.3)

Block 1	*Block 2*	*Block 3*
1	2	3
4	5	6
2	3	1
5	6	4

- $(1,4), (2,5), (3,6)$ are observed **twice**, remaining pairs only **once** together in the same block.
- The analysis is the same as for a BIBD!

Row-Column Incomplete Block Designs

- As we have seen with RCBs, we are sometimes facing the situation where we have **more than one** block factor (remember Latin Squares?).
- Latin Squares are often impractical due to their very strict constraint on the design.
- A **row-column incomplete block design** is a design where we block on rows **and** columns and one or both of them are **incomplete blocks**.

Example: Car Tires (Kuehl, 2000)

- Suppose we want to evaluate 7 treatments instead of 4.
- Assume that we have 7 cars and the following design:

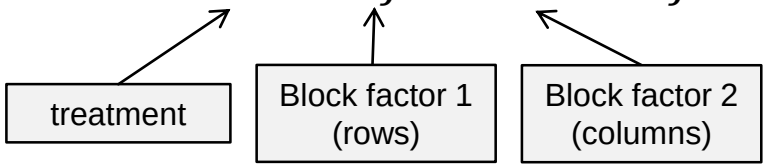
Tire position							
	3	4	5	6	7	1	2
	5	6	7	1	2	3	4
	6	7	1	2	3	4	5
	7	1	2	3	4	5	6

- The tire positions are **complete blocks**, the columns form a **BIBD**. This is a so called **row-orthogonal design**.

Youden Squares

- A **Youden Square** is **rectangular** (!) such that
  - columns (rows) form a **BIBD**
  - rows (columns): each treatment appears **equally often** in each row (column)
- Hence, columns form a **BIBD**, rows an **RCB**.
- The model is as before:

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



- Analysis in R “as usual”, just make sure to use `drop1` to ensure that the correct sum of squares is being used.

Example: Lithium in Blood (Oehlert, 2010, Ex. 14.5)

- Study was performed to measure **blood concentration of lithium** 12 hours after administering lithium carbonite using
  - A: 300mg capsule
  - B: 250mg capsule
  - C: 450mg time delay capsule
  - D: 300mg solution
- 12 subjects, each will be measured and treated **twice**, one week apart:

Week	1	2	3	4	5	6	7	8	9	10	11	12
1	A	D	C	B	D	D	B	B	C	A	A	C
2	B	C	A	C	A	B	A	D	D	D	C	B

- **Response**: Serum lithium level.

Example: Lithium in Blood (Oehlert, 2010, Ex. 14.5)

- We block on **both** rows (weeks) and columns (subjects).
- Every treatment appears 3 times in each week.
- The columns form a BIBD.
- Analysis in R:

```
> fit <- aov(hour.12 ~ subject + period + treatment, data = lith)
> drop1(fit, test = "F")
Single term deletions

Model:
hour.12 ~ subject + period + treatment
          Df Sum of Sq      RSS     AIC F value Pr(>F)
<none>                 0.016203 -143.22
subject   11  0.029946 0.046149 -140.09  1.3442 0.3449
period     1  0.031974 0.048177 -119.06 15.7871 0.0041 **
treatment  3  0.005603 0.021806 -142.09  0.9222 0.4728
```

- Unfortunately, we **cannot** detect any treatment effect here.

Summary

- Balancedness properties etc. ensure that we are performing the experiment as **efficient as possible**.
- If a design is **not** balanced anymore, we lose efficiency but **we can typically still analyze the data**.
- Exceptions are (e.g.) cases where a disconnected design has been used and the focus was on comparing all treatments.

- Package overview: https://cran.r-project.org/web/views/ExperimentalDesign.html