Design and Analysis of Experiments

Lecture notes

Part III

Christoph Kopp

Bern University of Applied Sciences
School of Agricultural, Forest and Food Sciences HAFL

December 10, 2018

Contents

8 Classical analysis (ANOVA tables)

8.1 Completely randomized designs

In Chapter 2.2, we showed how to use `aov` to produce an analysis of variance table for the analysis of a completely randomized design:

```
> ins.aov <- aov(sqrt(count) ~ spray, data = InsectSprays)
> summary(ins.aov)

#             Df Sum Sq Mean Sq F value Pr(>F)
# spray        5   88.4   17.69    44.8 <2e-16 ***
# Residuals   66   26.1    0.39
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Remember the *between sum of squares*

$$\text{SSB} = \sum_{j=1}^{k} \sum_{i=1}^{n_j} (\bar{Y}_j - \bar{Y})^2$$

and the *within sum of squares*

$$\text{SSW} = \sum_{j=1}^{k} \sum_{i=1}^{n_j} (Y_{ji} - \bar{Y}_j)^2 \,,$$

where $\bar{Y}_j$ denotes the mean of the n_j observations from treatment j and $\bar{Y}$ is the overall mean. By dividing each sum of squares by its degrees of freedom, the mean sums of squares are obtained, whose ratio is the F test statistic. (So far, this is only a brief reminder.) By decomposing the between sum of squares, we obtain similar tables for factorial designs, as shown in Chapter 7.

8.2 Complete blocked designs

For designs with plot structure, one has to be careful to tell the `aov` function about the plot structure, otherwise the wrong F tests are produced. This is simple to do for complete block designs. Let us illustrate this with the proper analysis of the insect sprays data, which takes the block effects into account.

```
> InsectSprays$block <- factor(rep(rep(1:6, each=2), times = 6))
> summary(aov(sqrt(count) ~ spray + Error(block), data = InsectSprays))
```

```
#
# Error: block
#           Df Sum Sq Mean Sq F value Pr(>F)
# Residuals  5   5.55    1.11
#
# Error: Within
#           Df Sum Sq Mean Sq F value Pr(>F)
# spray      5   88.4   17.69    52.6 <2e-16 ***
# Residuals 61   20.5    0.34
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The `Error` term is used to tell `aov` to add a *random* block effect. As a result, no test for the significance of the block effect is performed, as this is usually not of interest. The corresponding sums of squares (formulae not discussed) are calculated and the F test for the significance of the (fixed) spray effect takes the blocks into account and uses the proper degrees of freedom ($b-1$ for the blocks and $k-1$ for the treatments, supposing we have b blocks and k treatments).
The results obtained with `lme` are the same (compare the F statistic and the degrees of freedom with `aov`):

```
> library(nlme)
> ins.lme <- lme(sqrt(count) ~ spray, random = ~ 1 | block,
+                data = InsectSprays)
> anova(ins.lme, type = "marginal")

#             numDF denDF F-value p-value
# (Intercept)     1    61 364.780  <.0001
# spray           5    61  52.621  <.0001
```

The numerator degrees of freedom correspond to the number of parameters we test. In case we have k treatments, the numerator degrees of freedom are $k-1$. There is some controversy in the statistical literature regarding the denominator degrees of freedom. We follow the convention used by the authors of the `nlme` package, which we will explain in the context of split plots.
In general, you should only use `aov` for balanced designs and use the more general mixed models approach (e. g. `lme`) otherwise.

9 Incomplete block designs

This section is based on Dean, Voss, and Draguljić 2017, Ch. 11.

9.1 General design issues

If you want to test five treatments per goat in Example 5.1, you cannot use a complete block design because each goat has only four legs.

Example 9.1. *In a laboratory experiment, the day is to be used as blocking factor. A complete block design is not possible because we have six treatments each of which takes roughly six hours to be run.*

In this chapter, we focus on the situation where blocks sizes are smaller than the number of treatments, such that complete block designs are not realizable. For consistency with the literature, we use a slightly different notation in this chapter than in the rest of the script:

- v: number of treatments (originally, "varieties")
- b: number of blocks
- r: number of replications per treatment
- k: number of plots per block
- λ: number of concurrences (see below)

Treatments may or may not be structured. For example, it is possible to have a factorial treatment structure in an incomplete block design.
We focus on designs in which each treatment is observed the same number of times r in the whole experiment. The number of times that treatment i is observed in block j is denoted by n_{ij}. Statistically, it makes sense to observe as many different treatments as possible in each block, so that we will use *binary* designs, in which n_{ij} is either one or zero for all i, j (since $k < v$).
For general purposes (if no treatment comparison is more important than the others), often designs in which all pairs of treatments occur in the same or almost the same number of blocks together have good properties.
We refer to Dean, Voss, and Draguljić 2017, Ch. 11 for randomization issues and for the important result that all contrasts are estimable in a design if and only if the design is connected (cf. Session 3 of the learn team coaching). The same reference should be consulted for furter information on group divisible designs and cyclic designs. Here, we focus on balanced incomplete block designs.

9.2 Balanced incomplete block designs

9.2.1 Definition and construction

Consider a design with v treatments, each of which is observed r times. Hence, we must have vr experimental units in total. These experimental units are divided into b blocks, each of which has the same number of $k < v$ plots (*uniformity*). We further impose that

- the design is binary (as defined above) and
- each pair of treatments appears together in exactly λ blocks.

Each design which satisfies these conditions is called a *balanced incomplete block design* (BIBD), and λ is called the number of *concurrences*.
The following three conditions are necessary for the existence of a BIBD:

1. $vr = bk$, because all experimental units are used.
2. $r(k-1) = \lambda(v-1)$, because each treatment is applied in r blocks and in each of these blocks, there are $k-1$ other treatments. Since each treatment is observed exactly λ times together with each of the $v-1$ other treatments, the condition follows. Note that λ must be an integer.
3. $b \geq v$. (Fisher's inequality, proof omitted here.)

Unfortunately, these conditions are not sufficient for the existence of a BIBD. We do not discuss this further, but refer to the `ibd` package, which can produce BIB designs with the `bibd` function used as follows:

```
> bibd(v, b, r, k, lambda, ntrial, pbar = FALSE)
```

`ntrial` denotes the number of trials (set to 1). We produce a design with $v = 7$ treatments, $b = 7$ blocks, $r = 3$ replications of each treatment, block size $k = 3$ and concurrence $\lambda = 1$ (we directly extract the design):

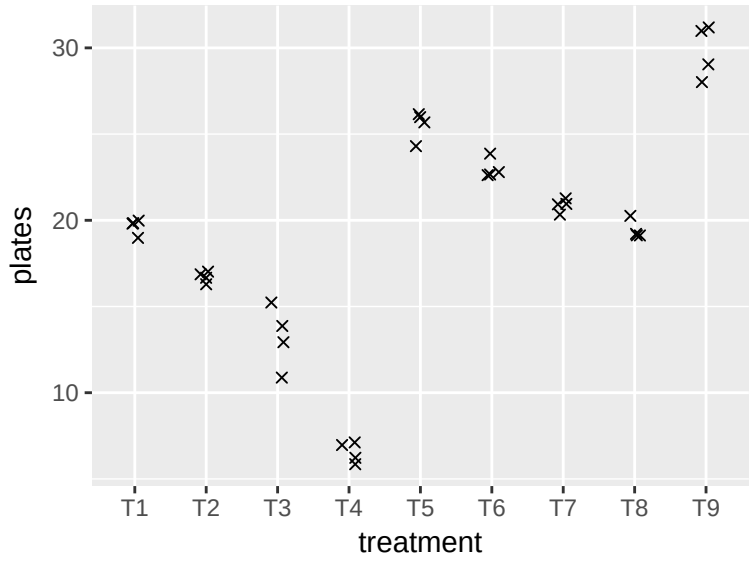
```
> library(ibd)
> set.seed(1) # to always get the same design
> bibd(7, 7, 3, 3, 1)$design

#         [,1] [,2] [,3]
# Block-1    5    6    7
# Block-2    1    4    5
# Block-3    1    3    7
# Block-4    3    4    6
# Block-5    2    4    7
# Block-6    1    2    6
# Block-7    2    3    5
```

As in Section 1.4, randomizing block and treatments yields the plan of the experiment.

9.2.2 The detergent data

This example is taken from Dean, Voss, and Draguljić 2017, Ch. 11. Three base detergents and an additive were studied. Detergents I and II were observed with 3, 2, 1 or zero parts of the additive, which defines treatments 1 up to 4 and 5 up to 8. Detergent III was the standard detergent and only observed without additive. It plays the role of the control treatment and is called treatment 9 below. For the experiment, three sinks were available, and three dishwashers washed (equally dirty) plates at a common rate. It was then counted how many plates could be washed before the detergent disappeared. A block consists of three observations, one from each sink. A BIBD with $v = 9$ treatments, block size $k = 3$, $r = 4$ observations per treatment and thus 12 blocks and concurrence 1 was used.

```
> detergent <- read.table("detergent.csv", header = TRUE, sep = ",")
> head(detergent)

#   block treatment plates base additive
# 1   B01        T3     13    1        1
# 2   B01        T8     20    2        0
# 3   B01        T4      7    1        0
# 4   B02        T4      6    1        0
# 5   B02        T9     29    3        0
# 6   B02        T2     17    1        2

> addmargins(with(detergent, table(treatment, block)))

#           block
# treatment B01 B02 B03 B04 B05 B06 B07 B08 B09 B10 B11 B12 Sum
#       T1    0   0   0   1   0   0   0   1   1   0   0   1   4
#       T2    0   1   0   0   1   0   0   0   0   1   0   1   4
#       T3    1   0   1   0   0   0   0   0   0   0   1   1   4
#       T4    1   1   0   0   0   1   0   1   0   0   0   0   4
#       T5    0   0   0   1   0   1   0   0   0   1   1   0   4
#       T6    0   0   1   0   1   1   0   0   1   0   0   0   4
#       T7    0   0   0   0   1   0   1   1   0   0   1   0   4
#       T8    1   0   0   0   0   0   1   0   1   1   0   0   4
#       T9    0   1   1   1   0   0   1   0   0   0   0   0   4
#       Sum   3   3   3   3   3   3   3   3   3   3   3   3  36

> ggplot(detergent, aes(x = treatment, y = plates)) +
+     geom_jitter(shape = 4, width = 0.1)
```



Points were jittered a bit to avoid overplotting. The plot above shows the *unadjusted* raw data: block effects are not accounted for.

9.2.3 Fixed effects model – intrablock analysis

We start with the more simple fixed effects model discussed in Chapter 5.3, according to which the measurement Y_{ji} of treatment j in block i is given by

$$Y_{ji} = \mu + \beta_i + \vartheta_j + \varepsilon_{ji}$$

for all (j, i) in the design (i. e. that $n_{ji} > 0$), where μ is a constant, β_j denotes the effect of treatment j, ϑ_i is the effect of block i and ε_{ji} are the random error terms. It is assumed that $\varepsilon_{ji} \sim \mathcal{N}(0, \sigma^2)$ independently. This model assumes that that blocks have *fixed* effects.

The model also assumes that there is no interaction of blocks and treatments. Since in an incomplete design, this is very hard/impossible to test, incomplete designs should only be used if there are very good reasons to assume that there is no substantial interaction between blocks and treatments; if there were such an interaction, a proper estimation of treatment effects would not be possible.

Because not all treatments are observed in all blocks, it is necessary to adjust for block effects when comparing treatments. Suppose that in the dishwasher experiment, the conditions were particularly favorable (for all treatments) exactly in those blocks in which treatment 1 occurs (i. e. blocks 4, 8, 9, and 12). Now suppose that you want to compare treatments 1 and 2. If you do this by comparing the sample means of treatments 1 and 2, this is very unfair, because treatment 1 was in the favorable blocks each time, whereas treatment 2 was in a favorable block exactly one time ($\lambda = 1$). A direct comparison of the sample means would be biased in favor of treatment 1. This problem does not occur in complete block designs (as long as there is no interaction of blocks and treatments), because there, the block effects apply to all treatments since the

design is complete. As long as the design is connected, software can handle the required adjustments, cf. Dean, Voss, and Draguljić 2017, Ch. 11.4.
In this situation (with fixed effects for the blocks), we can actually make good use of sequential F tests. In case we want to ask "*is there a treatment effect after adjusting for any block effect?*" we could run the analysis as below. Due to the asymmetric role of the two factors, the order of the variables in the model formula now matters. To answer the question above, the treatment effect has to be written *after* the block effect in the model formula.

```
> detergent.lm <- lm(plates ~ block + treatment, data = detergent)
> anova(detergent.lm)

# Analysis of Variance Table
#
# Response: plates
#           Df Sum Sq Mean Sq F value   Pr(>F)
# block     11  412.7   37.52   45.53 6.03e-10 ***
# treatment  8 1086.8  135.85  164.85 6.81e-14 ***
# Residuals 16   13.2    0.82
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> ## library(car)
> ## Anova(detergent.lm, type = 2) ## If you want Type II sums of squares
```

This analysis strategy is sometimes called the *intrablock analysis* because we base the analysis on differences from within the blocks. The treatment effect, adjusted for blocks, is highly significant. Now we could start investigating treatment contrasts or pairwise comparisons using `emmeans` for example, cf. Dean, Voss, and Draguljić 2017, Ch. 11.9.2.

Adjusting for block effects

It is very sensible to plot the data after adjusting for the block effects, since a plot of the raw data may be misleading due to confounding of block and treatment effects. We define

$$Y_{ji}^* = Y_{ji} - (\hat{\vartheta}_i - \hat{\bar{\vartheta}})$$

as the observation Y_{ji} adjusted for the block effect, where $\hat{\vartheta}_i$ is the least squares estimate of the effect of block i and $\hat{\bar{\vartheta}}$ denotes the least squares estimate of the mean block effect.

```
> theta.hat <- c(0, coef(detergent.lm)[2:12]) ## 0 for block 1 (reference)
> theta.avg <- mean(theta.hat)
```

```
> detergent$block.num <- as.numeric(detergent$block) ## for indexing
> detergent$adj <- theta.hat[detergent$block.num] - theta.avg ## note use of []
> detergent$plates.adj <- detergent$plates - detergent$adj
> head(detergent)

#   block treatment plates base additive block.num       adj plates.adj
# 1   B01        T3     13    1        1         1  0.361111   12.63889
# 2   B01        T8     20    2        0         1  0.361111   19.63889
# 3   B01        T4      7    1        0         1  0.361111    6.63889
# 4   B02        T4      6    1        0         2 -0.416667    6.41667
# 5   B02        T9     29    3        0         2 -0.416667   29.41667
# 6   B02        T2     17    1        2         2 -0.416667   17.41667
```

In the code above, we explicitly calculate the adjustment `adj` for each observation. This is done to clearly show what happens. We do not show the plot of the adjusted values, because they are very similar to the original values (treatment effects dominate block effects for this data set).

9.2.4 Mixed effects model – interblock analysis

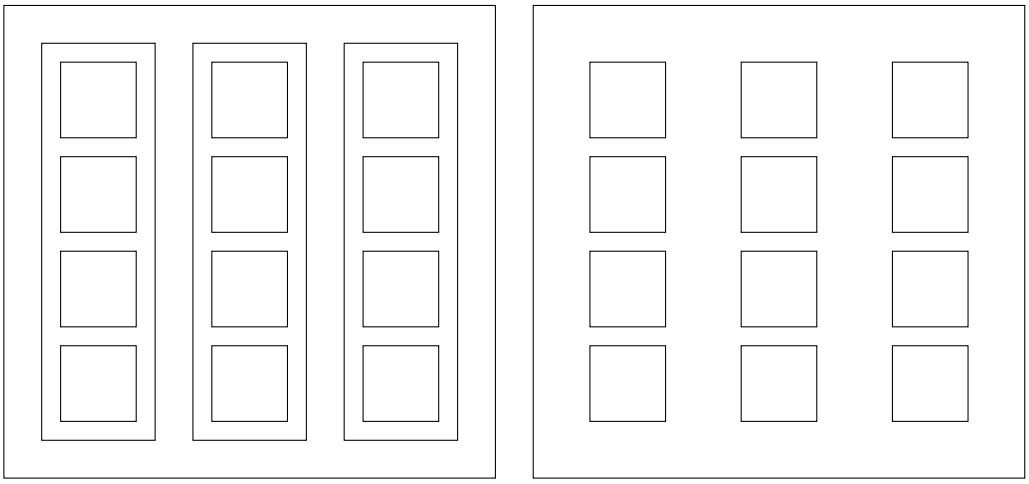
As discussed previously, in case the blocks were randomly chosen from a larger population of blocks, the interest is usually not in estimating the effect of the particular blocks sampled for this study, but more in the overall contribution of the blocking factor to the overall variability. Accordingly, in such a situation, the blocks effects should be modelled as random effects, as this provides more precise effect estimates. The corresponding analysis is simply a mixed effects model (with random block effect), and is sometimes called the interblock analysis.

```
> detergent.lme <- lme(plates ~ treatment, random = ~1 | block,
+                      data = detergent)
> anova(detergent.lme)

#             numDF denDF  F-value p-value
# (Intercept)     1    16 13942.63  <.0001
# treatment       8    16   220.57  <.0001
```

The significant treatment effect is confirmed with the mixed effects model.

10 Split-plot designs

Further discussion is found in Bailey 2008, Ch. 8.3, Dean, Voss, and Draguljić 2017, Ch. 19, Oehlert 2010, Ch. 16 (with many extensions such as split-split plots), Pinheiro and Bates 2000, Ch. 1.6 or Venables and Ripley 2002, Ch. 10.2.
The oats example (Ex. 1.10) is the classical split-plot example and will be analyzed below. As a reminder, we have six fields (large blocks), each of which is divided into strips (small blocks). In each strip, one variety is planted. Each strip is then divided into four plots, each of which receives one of four levels of Nitrogen fertilizer. In other words, variety is the main plot factor and Nitrogen is the subplot factor. Compare the layout of the oats split plot (at the level of one field) with a completely randomized design:



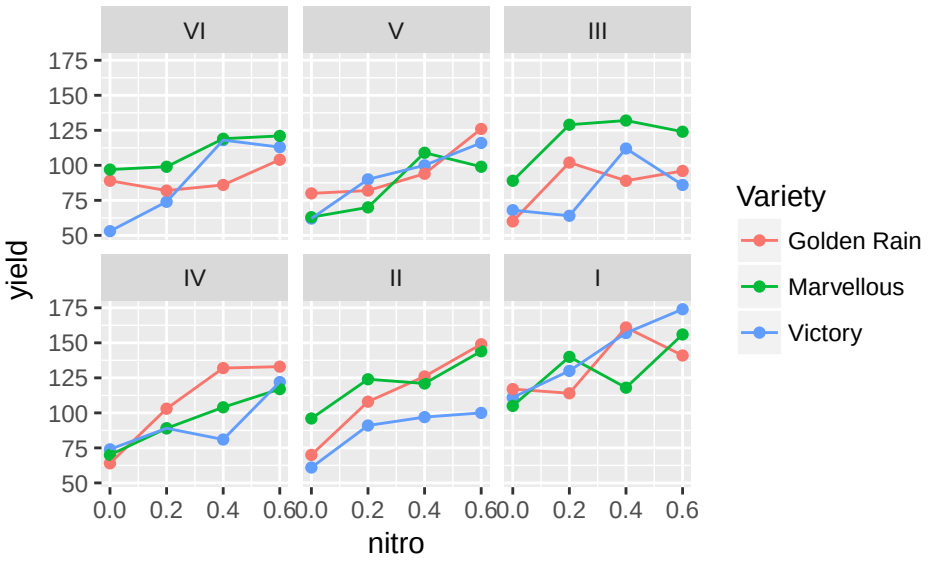
While these two designs are entirely different regarding the randomization and, in consequence, the proper statistical analysis, they will produce exactly the same data structure. In other words, by looking at the data set in software, you cannot tell the difference between the two designs.
A classic split-plot design has two different nested sizes of blocks: large blocks which contain small blocks which contain plots. The main reason for using a split-plot design is that the levels of one of the factors are harder to change. This factor then is the main plot factor. (The cost of using a split-plot design is that statements about the main plot factor are less precise than statments about the subplot factor.) The principle of split-plot designs may be further iterated and the split-plots may be split yet again. This leads to split-split-plot designs and so on.

10.1 Visualization of the raw data

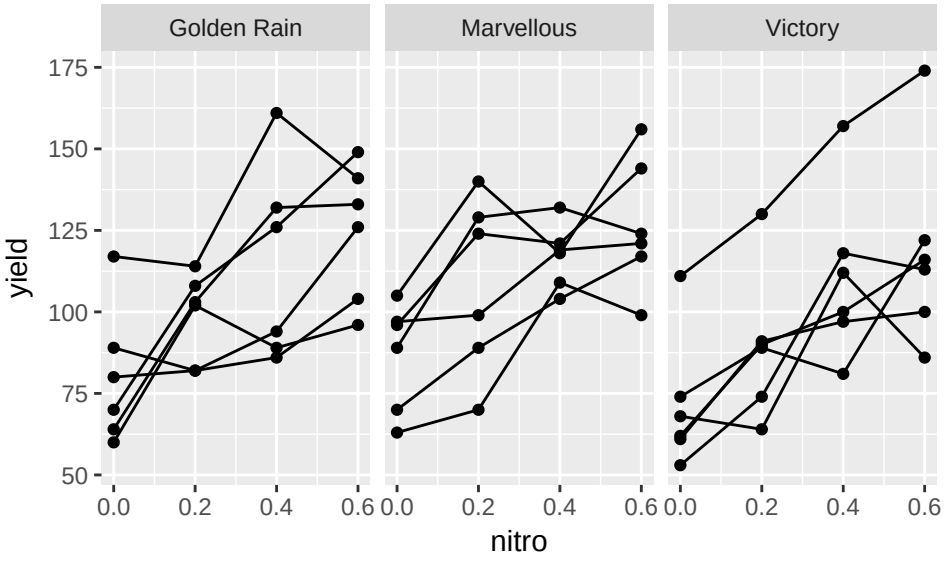
Let us compare two ways to visualize the `Oats` raw data. We can either facet by blocks or by varieties. (The block arrangement is unusual because `ggplot2` does not understand roman numerals – can you override this?).

To guide the eye, it is possible to connect the plots from the same group with a line. We use color here; in scientific journals, using color is often expensive, so different symbols (`shape` aesthetic in `ggplot2`) tend to be used instead of colors.

```
> ggplot(Oats, aes(x = nitro, y = yield, col = Variety)) +
+     geom_point() +
+     geom_line() +
+     facet_wrap(~Block)
```




```
> ggplot(Oats, aes(x = nitro, y = yield, group = Block)) +
+     geom_point() +
+     geom_line() +
+     facet_wrap(~Variety)
```



Faceting by blocks is closer to the experimental design in the sense that each facet contains a block and the values from the same block (field) stay in the same facet. On the other hand, the main interest is in the effect of the varieties, not the blocks, and this may be easier to compare when faceting by variety (omitting colors for blocks).

10.2 Basic statistical model

Classically, split-plot designs are analyzed with nested random effects. We show the analysis for the `Oats` data from `nlme` (you could also use `oats` from `MASS`). In our first model, we ignore the fact that the nitrogen levels have a natural order for simplicity (we will indicate how to do better below). In case we use the (default) treatment contrasts for both fixed effects, their first level is the respective reference level. The model equation is

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_{ij} + b_k + b_{i(k)} + \varepsilon_{ijk}$$

where Y is the yield (in quarter lb) and

- α_i is the main effect of variety i,
- β_j is the main effect of nitrogen level j,
- γ_{ij} is the interaction effect of variety i and nitrogen level j,
- $b_k \sim \mathcal{N}(0, \sigma^2_{\text{block}})$ is the block random intercept for block k,
- $b_{i(k)} \sim \mathcal{N}(0, \sigma^2_{\text{W}})$ is the whole plot i random intercept *within* block k,
- $\varepsilon_{ijk} \sim \mathcal{N}(0, \sigma^2)$ is the (sub-plot) error term.

We assume that the random effets and error terms are independent. To fit this model with the factorial treatment structure and the nested random effects:

```
> library(nlme)
> Oats$nitroF <- factor(Oats$nitro)
> Oats.lme <- lme(yield ~ Variety * nitroF, data = Oats,
+                 random = ~ 1 | Block / Variety)
> summary(Oats.lme)

# Linear mixed-effects model fit by REML
#  Data: Oats
#       AIC     BIC   logLik
#   559.029 590.444 -264.514
#
# Random effects:
#  Formula: ~1 | Block
```

```
#         (Intercept)
# StdDev:      14.645
#
#  Formula: ~1 | Variety %in% Block
#         (Intercept) Residual
# StdDev:     10.2986  13.3073
#
# Fixed effects: yield ~ Variety * nitroF
#                                Value Std.Error DF  t-value p-value
# (Intercept)                 80.0000   9.10696 45  8.78449  0.0000
# VarietyMarvellous            6.6667   9.71503 10  0.68622  0.5082
# VarietyVictory              -8.5000   9.71503 10 -0.87493  0.4021
# nitroF0.2                   18.5000   7.68296 45  2.40793  0.0202
# nitroF0.4                   34.6667   7.68296 45  4.51215  0.0000
# nitroF0.6                   44.8333   7.68296 45  5.83543  0.0000
# VarietyMarvellous:nitroF0.2  3.3333  10.86534 45  0.30679  0.7604
# VarietyVictory:nitroF0.2    -0.3333  10.86534 45 -0.03068  0.9757
# VarietyMarvellous:nitroF0.4 -4.1667  10.86534 45 -0.38348  0.7032
# VarietyVictory:nitroF0.4     4.6667  10.86534 45  0.42950  0.6696
# VarietyMarvellous:nitroF0.6 -4.6667  10.86534 45 -0.42950  0.6696
# VarietyVictory:nitroF0.6     2.1667  10.86534 45  0.19941  0.8428
#
# Standardized Within-Group Residuals:
#        Min         Q1        Med         Q3        Max
# -1.8130090 -0.5614484  0.0175804  0.6386448  1.5703417
#
# Number of Observations: 72
# Number of Groups:
#              Block Variety %in% Block
#                  6                 18

> ## ranef(Oats.lme) ## look at results!
```

In the `random` argument, the nested random effects are given such that the size of the nested blocks decreases from left to right. We estimate that $\hat{\sigma}_{\text{block}} = 14.65$, $\hat{\sigma}_{\text{W}} = 10.30$ and $\hat{\sigma} = 13.31$. To test the significance of the fixed effects, we keep using the marginal F test approach.

```
> anova(Oats.lme, type = "marginal")

#                numDF denDF F-value p-value
```

```
# (Intercept)        1    45 77.1673  <.0001
# Variety            2    10  1.2245  0.3344
# nitroF             3    45 13.0227  <.0001
# Variety:nitroF     6    45  0.3028  0.9322
```

The interaction effect is not significant, no evidence is found for a variety-specific nitrogen effect.
Before we reduce the model by removing the interaction terms, look at the degrees of freedom in the ANOVA table above. The numerator degrees of freedom are $m_0 = 1$ for the intercept and equal to the number of estimated terms for the other fixed effects (compare the `summary`). For the denominator degrees of freedom, the following approach is chosen, taking the nesting into account:

1. The block effect is the *level 1* effect, and there are $m_1 = 6$ blocks. There are $p_1 = 0$ fixed effects estimated at level 1. From that,

$$\text{denDF}_1 = m_1 - (m_0 + p_1) = 6 - (1 + 0) = 5\,.$$

2. The main plots (varieties) are the *level 2* effect. There are $m_2 = 18$ main plots, and we estimate $p_2 = 2$ (varieties) parameters at this level, so that

$$\text{denDF}_2 = m_2 - (m_1 + p_2) = 18 - (6 + 2) = 10\,.$$

3. The nitrogen effect and the interaction of nitrogen and the variety are *level 3* (sub-plot) effects. There are 12 subplots per block, so $m_3 = 72$ sub-plots, and we estimate $p_3 = 3 + 6 = 9$ effects at this level, so

$$\text{denDF}_3 = m_3 - (m_2 + p_3) = 72 - (18 + 9) = 45\,.$$

   (The intercept also has denDF_3 denominator degrees of freedom, this corresponds to the error term.)

The general formula for the denominator degrees of freedom at level $i > 0$ is

$$\text{denDF}_i = m_i - (m_{i-1} + p_i).$$

10.3 Model reduction

First of all, we want to remove the non-significant interaction effect. So we refit the model without it:

```
> Oats.lme <- lme(yield ~ Variety + nitroF, data = Oats,
+                 random = ~ 1 | Block / Variety)
> anova(Oats.lme, type = "marginal")

#             numDF denDF F-value p-value
# (Intercept)     1    51 94.5135  <.0001
# Variety         2    10  1.4853  0.2724
# nitroF          3    51 41.0528  <.0001
```

(Note how the degrees of freedom change because we now estimate six effects less at level three.) Now something very interesting happens. The main effect of the variety is not significant (so the situation is the same as before we removed the interaction). But the variety is also a part of the random effect. So, should we remove it or not?
To answer this, think about what the fixed and the random variety effects do (cf. Pinheiro and Bates 2000, Ch. 1.6). The random variety within blocks effect allows for different random intercepts for each of the strips within a block. It essentially models intra-strip correlation in each block (due to fertility, soil, ...). On the other hand, the fixed effect of the variety is not specific to a particular block and models a systematic variety difference across blocks. Based on the F test above, there seems to be no need for such a fixed effect. So we could remove the fixed effect of the variety.

```
> Oats.lme <- lme(yield ~ nitroF, data = Oats,
+                 random = ~ 1 | Block / Variety)
```

10.4 Exploiting order structure

The nitrogen levels have a very special structure, and we did not model it so far. The levels are equidistant, so we could actually treat the levels as numeric. To take advantage of this special structure, we can convert the nitrogen into an *ordered factor*. (This technique is not limited to split plots but is a general approach.) For brevity, we continue with the model that has no variety effect.

```
> Oats$nitroF <- factor(Oats$nitro, ordered = TRUE)
> Oats.lme.o <- lme(yield ~ nitroF, data = Oats,
+                 random = ~ 1 | Block / Variety)
> coef(summary(Oats.lme.o))

#                  Value Std.Error DF   t-value     p-value
# (Intercept) 103.972222   6.64067 51 15.656889 4.03006e-21
# nitroF.L     32.944735   3.00517 51 10.962691 5.12120e-15
```

```
# nitroF.Q     -5.166667   3.00517 51 -1.719260 9.16317e-02
# nitroF.C     -0.447214   3.00517 51 -0.148815 8.82287e-01
```

This has the consequence that now, so-called *orthogonal polynomial* contrasts are used for the nitrogen factor. Briefly, the first contrast (`nitroF.L`) estimates the linear trend, the second contrast (`nitroF.Q`) estimates the quadratic effect orthogonal (independent) of the linear term, and `nitroF.C` estimates the orthogonal cubic trend. In this case, only the linear trend is significant, no curvature is needed.
If we compare this with the fitted values (excluding the random effects) from the model that treats the nitrogen as a normal factor, the linearity seems acceptable (perhaps except for the last level).

```
> df <- data.frame(nitroF = levels(Oats$nitroF))
> df$pred <- predict(Oats.lme, df, level = 0)
> df

#   nitroF     pred
# 1      0  79.3889
# 2    0.2  98.8889
# 3    0.4 114.2222
# 4    0.6 123.3889

> ## coef(summary(Oats.lme)) ## to check
```

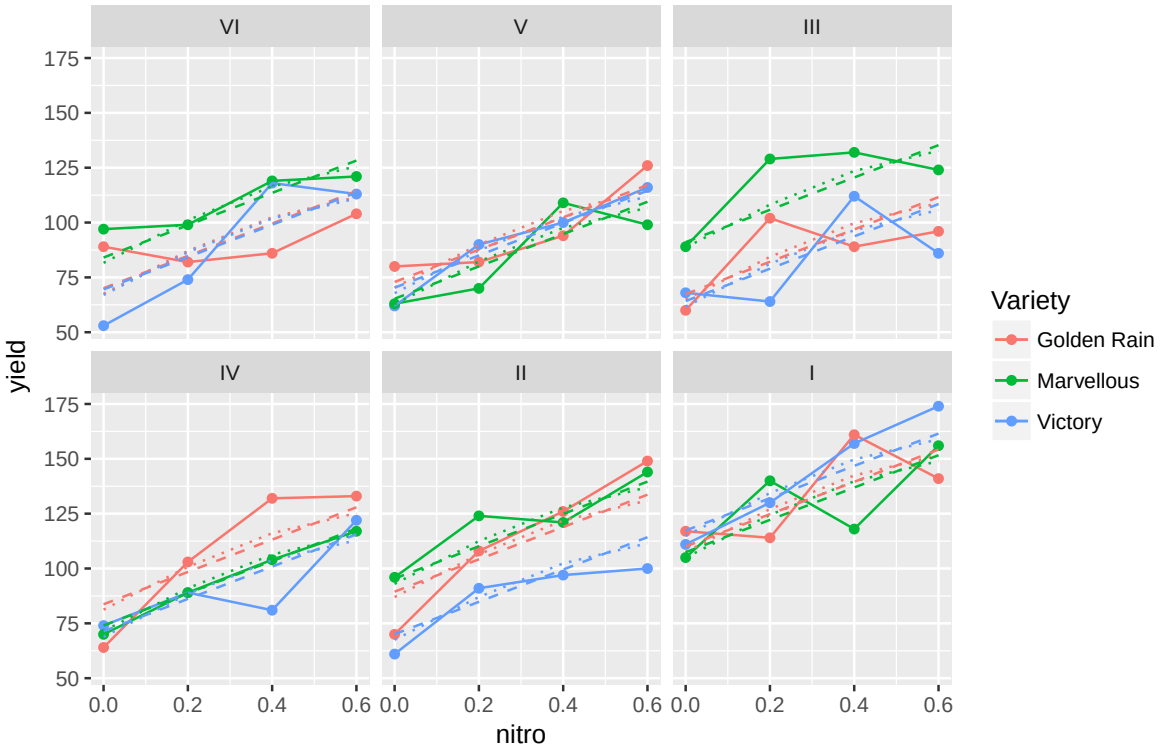
This leads us to our final model, which has a linear regression structure for the fixed effects (compare the next section).

```
> Oats.lme.lin <- lme(yield ~ nitro, data = Oats,
+                     random = ~ 1 | Block / Variety)
> summary(Oats.lme.lin)

# Linear mixed-effects model fit by REML
#  Data: Oats
#       AIC     BIC   logLik
#   603.042 614.284 -296.521
#
# Random effects:
#  Formula: ~1 | Block
#         (Intercept)
# StdDev:      14.506
#
```

```
#  Formula: ~1 | Variety %in% Block
#         (Intercept) Residual
# StdDev:     11.0047   12.867
#
# Fixed effects: yield ~ nitro
#               Value Std.Error DF t-value p-value
# (Intercept) 81.8722   6.94528 53 11.7882       0
# nitro       73.6667   6.78148 53 10.8629       0
#
# Standardized Within-Group Residuals:
#        Min         Q1        Med         Q3        Max
# -1.7438078 -0.6647522  0.0171043  0.5429879  1.8029891
#
# Number of Observations: 72
# Number of Groups:
#              Block Variety %in% Block
#                  6                 18
```

We claim that (for our range of nitrogen values) an increase of the nitrogen by 0.2 units (cwt) raises the yield by $0.2 \cdot 73.66 = 14.73$ units on average for a typical plot. To visualize the two models, we plot the data and the fitted values.

```
> Oats$yield.factor <- predict(Oats.lme)
> Oats$yield.lin <- predict(Oats.lme.lin)
```



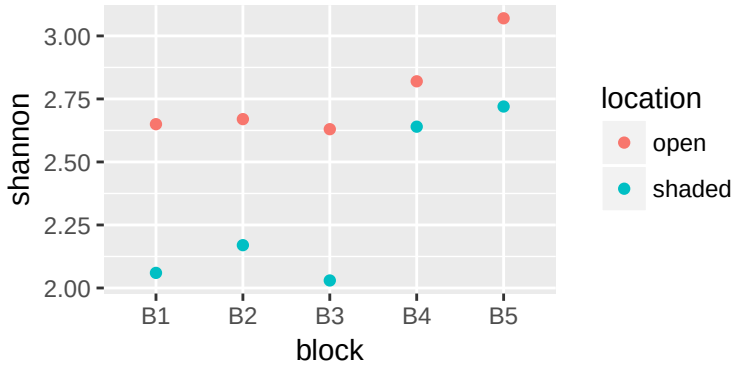
The data are connected by solid lines, the factor model fitted values by dotted lines, and the linear regression model fitted values by dashed lines.

The two models yield very similar fitted values in all blocks. In case a linear effect provides a reasonably accurate description of the nitrogen-yield relation, such a simple regression model is preferred to a model which treats the nitrogen as a factor because in the regression model, less parameters have to be estimated (bias-variance tradeoff). In consequence, the simpler regression model should be preferred here (none of the quadratic or cubic terms of the orthogonal polynomial contrasts for the nitrogen were significant).

11 Using covariate information

In the RCBD Example 5.3, the biodiversity of ten plots was measured with the Shannon index (calculated from the number of observed individual plants and plant species on a plot; higher values mean more diversity). The ten field plots were divided in five blocks depending on the distance to a hedge in the west. In each block, one plot was closer to the forest and one was further away from the forest (variable `location`, acts as treatment here). Strictly speaking, this is not an experiment, because of the missing randomization, but an observational study.

```
> gasel <- read.table("gasel.csv", sep = ",", header = TRUE)
> ggplot(gasel, aes(x = block, y = shannon, col = location)) +
+     geom_point()
```



Bioviversity is higher in open plots. Is the effect significant? Given the small sample size, a Friedman test seems like a reasonable approach:

```
> friedman.test(shannon ~ location | block, data = gasel)
```

The Friedman test finds a significant location effect. To use covariates below, we use a parametric mixed effects model now, although the data set is perhaps too small.

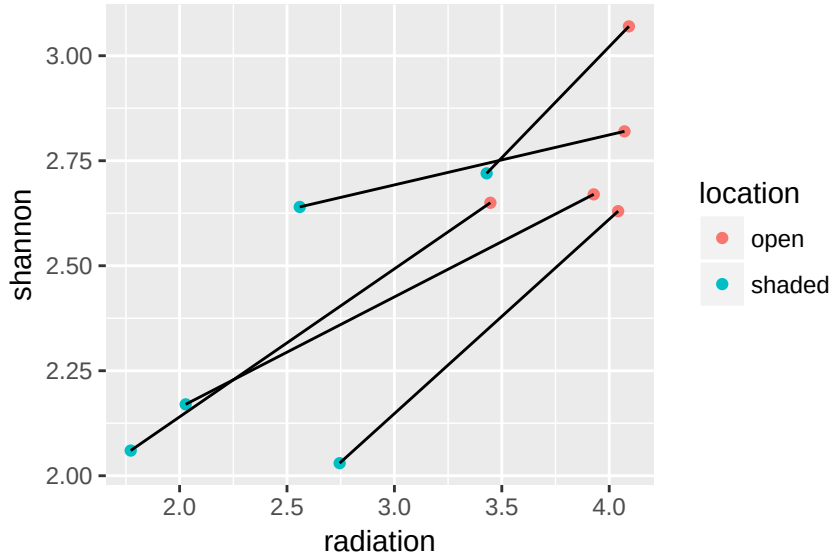
```
> gasel.blk <- lme(shannon ~ location, random = ~ 1 | block,
+     data = gasel)
> coef(summary(gasel.blk))

#                 Value Std.Error DF t-value     p-value
# (Intercept)     2.768 0.1196871  4 23.1270 2.07149e-05
# locationshaded -0.444 0.0797872  4 -5.5648 5.10736e-03
```

The mixed effects model also finds a significant location effect, and it furthermore yields an effect estimate. For a typical block, the Shannon index of the shaded plots is 2.77. On

average, the Shannon index is 0.44 units lower in the shaded plots than in the open plots.

During the experiment, GPS coordinates were taken for each plot and then used in a GIS (using a digial surface model based on LIDAR data) to calculate the total amount of solar radiation that each plot receives over the vegetation period up to the experiment date (March 03, 2018 – July 04, 2018) (`radiation`, Wh/m^2).
The question now is whether the radiation data can help us understand the Shannon index better, because the radiation data might be more useful than just the blocks and the treatment.

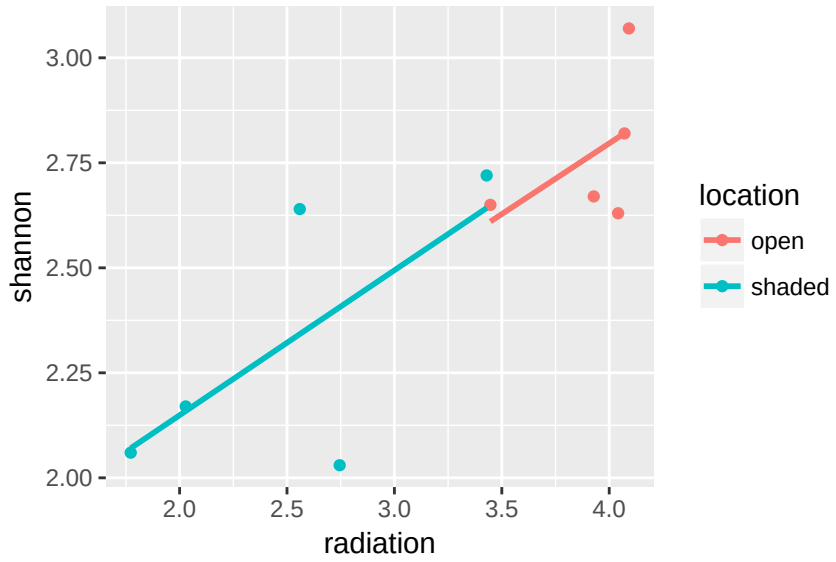
```
> gasel$radiation <- gasel$radiation / 100000
> ggplot(gasel, aes(x = radiation, y = shannon)) +
+     geom_point(aes(col = location)) +
+     geom_line(aes(group=block))
```



Here is a visualization where the two plots from each block are connected by a line. A clear effect of radiation is seen in each block. Let us now show how to model this effect (see also the D3 lecture notes, Section 5). We clearly overfit this small data set, but we just want to show the principle. The most general approach is to allow a different linear effect of radiation for the open and for the shaded plots.

```
> gasel.lme <- lme(shannon ~ radiation * location, random = ~ 1 | block,
+                  data = gasel)
> coef(summary(gasel.lme))

#                               Value Std.Error DF   t-value  p-value
# (Intercept)                1.9511178  1.590948  4  1.226387 0.287309
```

```
# radiation                 0.2086094  0.405372  2  0.514612 0.658050
# locationshaded           -0.2660250  1.458294  2 -0.182422 0.872068
# radiation:locationshaded  0.0462399  0.365392  2  0.126549 0.910873

> ggplot(gasel, aes(x = radiation, y = shannon, col = location)) +
+     geom_point() +
+     geom_smooth(method = "lm", se = FALSE)
```



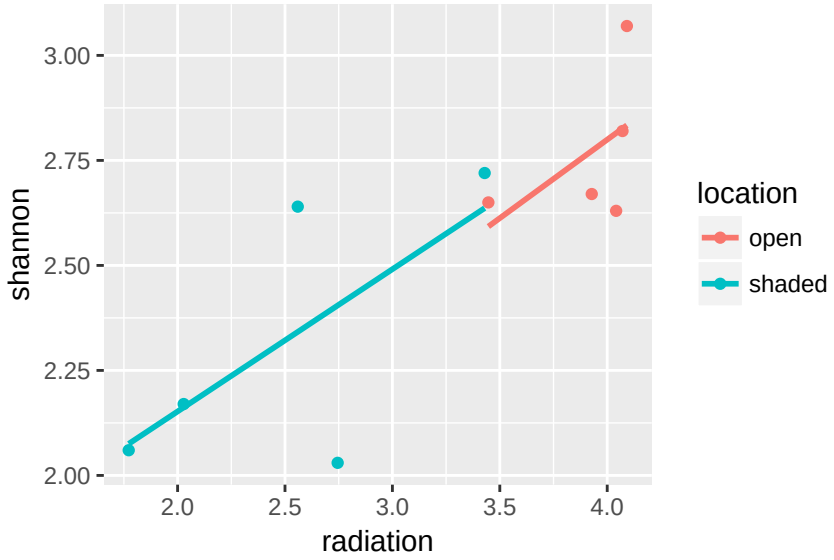
We estimate that for a typical block, on average

$$\text{shannon} = \begin{cases} 1.95 + 0.21 \cdot \text{radiation} & \text{for } \texttt{open} \text{ locations}, \\ (1.95 - 0.27) + (0.21 + 0.046) \cdot \text{radiation} & \text{for } \texttt{shaded} \text{ locations}. \end{cases}$$

The shaded locations have a slope which is 0.046 units higher than the open locations, but this slope difference is not significant. Therefore, we may want to reduce the model, so that only one common slope is estimated (but different intercepts remain allowed):

```
> gasel.lme <- lme(shannon ~ radiation + location, random = ~ 1 | block,
+                  data = gasel)
> coef(summary(gasel.lme))

#                    Value Std.Error DF   t-value   p-value
# (Intercept)     1.818033  0.618109  4  2.941281 0.0423351
# radiation       0.242595  0.155727  3  1.557828 0.2171526
# locationshaded -0.102220  0.236605  3 -0.432029 0.6948894

> ggplot(gasel, aes(x = radiation, y = shannon, col = location)) +
+     geom_point() +
```

```
+     geom_smooth(aes(y = predict(gasel.lme, gasel)), method = "lm",
+                 se = FALSE)
```

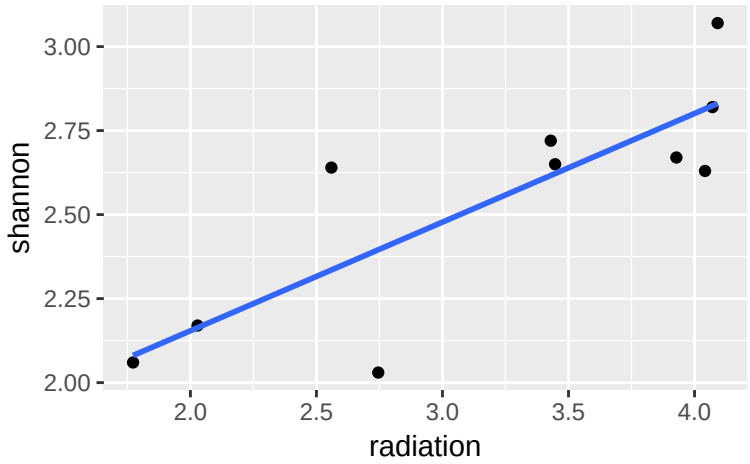


We estimate that for a typical block, on average

$$\text{shannon} = \begin{cases} 1.81 + 0.24 \cdot \text{radiation} & \text{for } \texttt{open} \text{ locations}, \\ (1.81 - 0.10) + 0.24 \cdot \text{radiation} & \text{for } \texttt{shaded} \text{ locations}. \end{cases}$$

The shaded locations have an intercept which is 0.10 units lower than the open locations, but this difference is not significant. Because the data set is so small and we fit so many parameters, the radiation effect is also not significant yet. We reduce the model, so that only one common regression line is used.

```
> gasel.lme <- lme(shannon ~ radiation, random = ~ 1 | block,
+                  data = gasel)
> fixef(gasel.lme)

# (Intercept)   radiation
#    1.562511    0.306247

> ggplot(gasel, aes(x = radiation, y = shannon)) +
+     geom_point() +
+     geom_smooth(aes(y = predict(gasel.lme, gasel)), method = "lm",
+                 se = FALSE)
```



We estimate that the Shannon index of observation i in block j is given by

$$\text{shannon}_{ji} = 1.56 + 0.31 \cdot \text{radiation} + b_j + \varepsilon_{ji}\,,$$

where $b_j \sim \mathcal{N}(0, 0.16^2)$ and $\varepsilon_{ji} \sim \mathcal{N}(0, 0.14^2)$. For this data set, the covariate was so important that it completely eliminated the treatment information from the model. If covariates are thought to be relevant for the experiment, they should be measured and used in the analysis, because this allows a more precise estimation of the treatment effect. With more data, it is also possible to use methods from multiple linear regression, such as regression splines to account for nonlinear covariate effects.