Design and Analysis of Experiments

Lecture notes

Christoph Kopp

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

March 15, 2019

Preface

These lecture notes accompany the D2 module, which is the second statistics module in the MSLS curriculum. The D1 module focuses on data management and visualisation. I will use parts of the D1 module without further comments.
These lecture notes are accompanied by short further notes, namely:

- *Tools from Probability Theory*
  containing basic probabilistic tools and distributions used throughout the course,
- *Concepts from Statistical Inference*
  where we conceptually show how to quantify the uncertainty induced by making statements about a population, based on a random sample from it

These are also used in the D3 module. The course Moodle site details how these documents are related to preparatory work for the lectures, and I assume that you have read them and done the preparatory work.

In this module, we assume that we are in a position where we can plan and carry out an experiment. Two perspectives will often be important: the scientist and the statistician. The scientist has a clear research question that she wants to answer. She has some resources available to collect data and mostly comes up with an initial proposal for a designed experiment.
We will adopt the perspective of the statistician in these notes: she tries to help the scientist with the design and the analysis of the data and translates statistical results back to the framework of the scientist to answer the research questions. Planning data analysis before data collection helps her design a good experiment that has a good chance of providing an answer to the research question while being efficient in the use of resources.

I profited enormously from scripts by Michael Vock, Michael Mayer, Sabine Güsewell and Beat Huber-Eicher to compile these lecture notes. I would also like to thank the numerous students which commented and improved earlier versions.

Contents

1 Definitions and first examples 6

1.1 Basic definitions 6

1.2 Treatment structure 7

1.3 Plot structure 7

1.4 The design 8

2 Completely randomized designs without treatment structure 10

2.1 Introduction 10

2.2 Parametric models: one-way ANOVA 15

2.3 Pairwise treatment comparisons 21

3 Completely randomized designs with control 25

3.1 Comparing all treatments with the control 25

3.2 More about contrasts 27

3.3 Specific comparisons 29

4 Model diagnostics and alternative analysis methods 32

4.1 The normality assumption 32

4.2 The homoskedasticity assumption 35

4.3 Consequences of model assumptions not holding 37

4.4 Nonparametric methods 39

4.5 Robust methods 40

4.6 Permutation methods 41

4.7 Exercises 43

5 Blocking 44

5.1 Types of blocks 44

5.2 Complete block designs 45

5.3 Fixed block effect models 46

5.4 Random block effect models 47

6 Fitting and interpreting mixed effects models 48

6.1 Data structure 48

6.2 Fitting the model 48

6.3 Extracting model results 50

6.4 Model diagnostics 53

6.5 Confidence intervals and a hypothesis test 54

6.6 Treatment comparisons 55

6.7 Ignoring blocks 58

6.8 The intraclass correlation coefficient 58

6.9 The coefficient of determination 59
6.10 An example: the `immer` data and the Friedman test 59
6.11 Interactions of block effects and treatment effects 63

7 Designs with factorial treatment structure 67
7.1 Introduction 67
7.2 Visualization 68
7.3 A model for factorial two-way ANOVA with replications 70
7.4 F tests 74
7.5 Tests on individual coefficients 76
7.6 Model diagnostics and related topics 77
7.7 Fitted values and confidence intervals 81
7.8 Post hoc tests 82
7.9 Unbalanced designs 83
7.10 Accounting for blocks and for heteroskedasticity 84

8 Classical analysis (ANOVA tables) 88
8.1 Completely randomized designs 88
8.2 Complete blocked designs 88

9 Incomplete block designs 90
9.1 General design issues 90
9.2 Balanced incomplete block designs 91

10 Split-plot designs 96
10.1 Visualization of the raw data 96
10.2 Basic statistical model 98
10.3 Model reduction 100
10.4 Exploiting order structure 101

11 Using covariate information 105

12 Extensions 110
12.1 Row-column designs 110
12.2 Confounded factorial experiments 112
12.3 Fractional factorial designs 113
12.4 Alternative analysis approaches 114

13 Power and sample size calculations 115
13.1 Monte Carlo approach 116
13.2 Analytical results 118
13.3 Efficiency 119

14 Planning experiments 120

14.1 The protocol 120

References 123

A Data set overview 124

1 Definitions and first examples

This section borrows heavily from Bailey 2008, Ch. 1.

better experiment design

cause ~ effect relationship

optimal disign

1.1 Basic definitions

An *experimental unit* is the smallest unit to which a treatment can be applied. A *treatment* describes everything that was applied to the experimental unit.
An *observational unit* is the smallest unit on which a response will be measured.

What are the treatments, the experimental and the observational units in the following examples?

Example 1.1. *Diarrhea can be a serious health problem for piglets. The two-week weight gain of piglets on two different diets is measured. Sixteen pens are available, each of which contains five piglets from the same mother. Each diet is given to all animals in eight randomly selected pens, piglets eat ad lib. Piglets are weighed individually at birth and after two weeks, and their weight gain is calculated.*

Example 1.2. *To study the effect of varieties and location on the expression of a potato disease, seven varieties of potatoes were planted in five different locations. In each location, 28 plots were available, and all seven varieties were planted four times at each location. After harvest, the total number of infected tubers out of 300 randomly sampled tubers was determined for each plot.*

The examples highlight the type of relation between experimental and observational units that we will encounter. Either

- experimental and observational units coincide, or
- each experimental unit contains several observational units.

In Example 1.1, the treatments are the two diets. They are applied to pens, so the experimental units are the pens (not the piglets!), while the observational unit are the piglets.

In Example 1.2, the treatments are the *combinations* of location and variety, so there are 35 treatments. The treatments are applied to the plots, so the experimental units are plots. The observational units could have been the individual tubers, but here, only the total number of infected tubers per plot was analyzed, so that we have plots as observational units.

The design of experiments has its roots in agronomy, which is why observational units are often called *plots*.

1.2 Treatment structure

The simplest experiments have no treatment structure. This means that no treatment is in any way special compared to the others.

Example 1.3. *We study the foam production rate of five different brands of soap.*

In many experiments, the treatments have some special relations. In Example 1.3, one soap could be the industry standard to which we compare four new soaps.

Many studies contain several treatments plus a *control* (do-nothing "treatment"). Omission of the control treatment can completely invalidate the conclusions of the entire experiment.

Example 1.4. *To compare the effect of the homeopathic and standard medical treatment of tooth ache, a placebo treatment has to be included.*

Not all experiments need a control treatment.

Example 1.5. *In the treatment of an aggressive pest, it may not be a realistic option to apply no countermeasures. In that case, the control treatment could be the standard treatment. The same principle applies to clinical studies of severe diseases. It is not ethically defendable to give a patient no treatment if a standard treatment is available.*

Example 1.6. *In a study of the effect of MDMA (Ecstasy) on the treatment of patients with PTSD, the "control" group received a so-called* active placebo, *i. e. a low dose of MDMA. The reason was that patients would otherwise notice that the were in the placebo group.*

Example 1.7. *Often, all combinations of two different treatment variables are studied, these are called* factorial *treatment structures, e. g. Example 1.2. The same idea extends to several factors (then often with fewer levels per factor, as applied in chemistry). These designs may or may not have one or several added control treatments.*

Example 1.8. *Patients in a clinical study may be assigned placebo (Dose 0) or one of three increasingly higher doses of a substance (50, 100 or 150 mg/kg bodyweight). The dose is then called an* ordinal factor. *This setup is typical in toxicity studies.*

Treatment structure does not involve the plots. Similary, we can study the structure of plots while ignoring treatments.

1.3 Plot structure

The simplest experiments have no plot structure. This means that no plot is in any way special compared to the others.

link the treatment to the plot

Example 1.9. *To study the solidity of two new cements for fixing teeth, wisdom teeth are collected from patients who need their wisdom teeth extracted and are willing to donate them for research. One tooth per person is used and assigned the cement at random. Then, the tooth is treated and its breaking strength is measured.*

In many experiments, the plots have some special relations. The most important case is that experimental units contain observational units, see Example 1.1. Let us study this a bit closer.

Due to the the high variability between patients, the treatments in Example 1.9 should only be compared *within* patients. So, a pair of wisdom teeth could be collected from patients who need several wisdom teeth extracted. Each treatment is then randomly applied to one of the two teeth from the same patient. Each patient is a *block* and the plots (teeth) are structured in blocks. The pens in Example 1.1 also serve as blocks.

Blocks are very important in experimental design. They may also be further nested.

Example 1.10. *You have six fields for your experiment on the effect of three varieties and four levels of Nitrogen fertilizer on the yield of oats. Inside each field, you plant three varieties in strips (because this is technically easier). Inside each strip, you have four plots to each of which you apply one of four levels of Nitrogen fertilizer. This is an example of bigger blocks (fields) containing smaller blocks (strips) containing the plots. This particular type of plot structure is a* split-plot *design.*

Remark 1.1. In the context of split plot experiments, special terminology is often used. The bigger blocks are just called blocks, and the smaller blocks are called *whole/main plots*, while what we call plots (smallest units) are called *subplots*. With these terms, the oats experiment consists of six blocks each having three main plots, each of which is split into four sub-plots. The varieties are assigned to the main plots and the nitrogen treatments to sub-plots. A compact way to express this is to say that varieties are the main plot factor and nitrogen is the sub-plot factor.

Split-plot designs are often used due to practical reasons because one factor is harder to change; this is then the factor assigned to the main plots. They are very popular in industry. We will discuss the statistical consequences for the precision of the estimates due to such a nested structure.

1.4 The design

Let us now link treatments and plots. Each plot receives exactly one treatment.

> The *design* is the allocation of treatments to plots.

It is useful to distinguish the design (which has abstract plots that may be numbered from 1 to n, and abstract treatments $T_1, \dots T_k$) from the actual *plan* or *layout*. The layout concerns the real plots and is obtained by randomizing the design.

Example 1.11. *For a small plant experiment, six pots are available to compare the effect of three different fertilizers on the yield. Each treatment (A, B or C) is to be applied to two pots. The design is given in the first two columns of the table below.*

To obtain the plan, we randomize the design by choosing a random permutation of the numbers 1 to 6. To do this in `R`*, you can simply type* `sample(1:6)`*. The results may change every time you run the code. I did this and obtained the vector* $(1, 5, 6, 3, 2, 4)$*. This defines the mathematical permutation*

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 5 & 6 & 3 & 2 & 4 \end{pmatrix}$$

which means that the design treatment of plot 1 (T_1) is unchanged (we label it as A for the actual treatment), the design treatment of plot 2 (T_1) is applied to plot 5, and so on. This then gives the randomized design with the actual treatments.

Pot Nr.	*Design Treatment*	*Actual Treatment*
1	T_1	*A*
2	T_1	*C*
3	T_2	*B*
4	T_2	*C*
5	T_3	*A*
6	T_3	*B*

Remark 1.2. We *completely randomized* the design, without any restrictions. Imagine that pots 2, 4 and 6 are closer to the window and get more light than the other three pots. In that situation, we might not want to allow a permutation that puts both A treatments in relative shade. One solution is to restrict the randomization such that additional constraints (e. g. each treatment being in the shade exactly one time) are satisfied. It is important to incorporate relevant constraints in the randomization.

Remark 1.3. Many important research questions cannot be studied by experiments because randomization is not possible. The classical example is that of the effect of smoking on health. It is not feasible to randomly assign people to a smoking group (none, moderate, heavy smoking); instead, one has to rely on so-called *observational data* from a nonrandomized “design”. This immediately makes interpretation more difficult because the lack of randomization implies that smoking behavior could be associated with income, education or other factors which are also known to have health effects. The question then becomes how to properly separate the effect of the factors. In these notes, wo focus on experimental data.

It is crucial to respect plot and treatment structure in the statistical analysis. We show how to do this for a few designs of increasing complexity below, starting with completely randomized designs.

2 Completely randomized designs without treatment structure

方差分析

ANOVA provides a statistical test of whether two or more population means are equal, and therefore generalizes the t-test beyond two means

2.1 Introduction

> Assume that we have k treatments and that treatment j is applied to n_j plots, $j = 1, \ldots, k$. If treatments are assigned to the plots at random without any restrictions, the design is called a *completely randomized design* (CRD).

In a CRD, plots have no structure. Treatments may or may not be structured in a CRD. In this chapter, we also assume unstructured treatments.

In the context of statistical models, the outcome is often called the *dependent variable* and the variable defining the treatments is called the *independent variable*. The terms *dependent* and *independent* are not a property of the variables themselves, they indicate how we intend to use them in the analysis.

In completely randomized designs, observational units were randomly allocated to the treatments, so the only systematic influence on the outcome should be due to the treatment. It is then justified to speak of the *effect* of the treatment on the outcome.

In general, not all plots will show precisely the same response to a treatment.[1] There is always some variablility, such that often one has to be content if a treatment shows some desired effect *on average*.

The first aim of the analysis is usually to investigate whether the mean of the numeric variable differs significantly between the treatments and to quantify the difference. Sometimes also other measures of location such as the median are studied.

2.1.1 R formulae

An R *formula* is an expression of the kind `y ~ x, data = mydata` with two sides separated by the `~` symbol. The *left* hand side contains the *dependent* variable which we are trying to model using the *independent* variables (there may be more than one) on the *right* hand side. Formulae are a way of specifying which variable should play which part in a model. They can also be used for plotting in R.

2.1.2 The insect sprays data

Consider the `InsectSprays` data set contained in R. It contains counts[2] of surviving insects (tobacco hornworm) in agricultural plots treated with different insecticides. The research question is whether the efficacy of the insecticides differs. The dependent

[1] Calling humans "plots" should not be done outside statistics lecture notes!

[2] Although these are count data, we treat them as numeric because they take a large number of different values in the data set. See *generalized linear model* for proper count data models.

variable is the number of surviving insects and the independent variable is the type of insecticide applied. To get a first impression of the data:

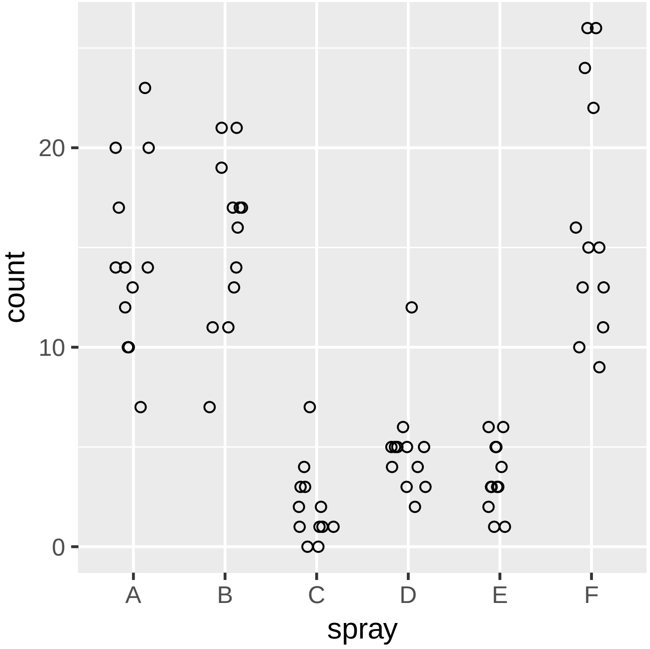
```
> with(InsectSprays, tapply(count, spray, length)) #12 obs. per trt
> with(InsectSprays, tapply(count, spray, summary)) #compare distr.
> with(InsectSprays, tapply(count, spray, sd)) #standard deviations
```

> Before fitting a model, get to know the data better (*know your data!*). Visualization is an essential step to understand the information contained in the data.

2.1.3 Visualization

To visualize the data, box plots or strip plots can be used.[3] Using ggplot2[4], a jittered strip plot can be produced by: what is a jittered....处理由于点重合比较严重的现象

```
> library(ggplot2)
> ggplot(InsectSprays, aes(spray, count)) +
+     geom_point(shape = 1, position = position_jitter(width = 0.2,
+                                                      height = 0))
```

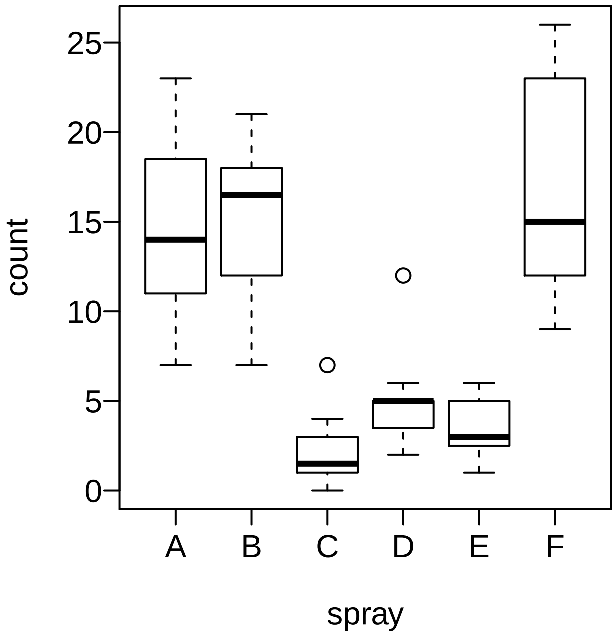


[3] In case less than 15 points per treatment are available, perhaps using a strip plot is better. Here, the `position` argument aids visualization by adding some random jitter in horizontal direction so that points do not overlap completely.

[4] The same can be obtained with the `stripplot` function from the `lattice` package with `stripplot(count ~ spray, data = InsectSprays, jitter = 0.2)`.

Visually, it seems that the amount of surviving insects differs by insecticide. It also looks as if the three sprays C, D and E with the lower values also had a lower variability. While a strip plot plots the raw data, a more coarse view is obtained with box plots of the data (strictly speaking, box-and-whisker plots; the differences do not matter here). The box plots may be obtained with[5]

```
> plot(count ~ spray, data = InsectSprays, las = 1)
```



Look for

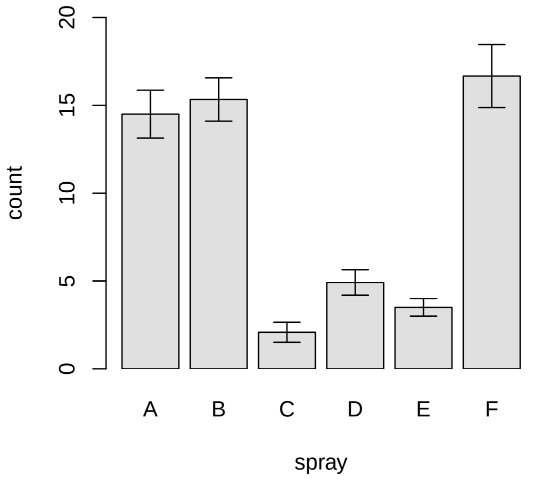
- location differences (compare the location of the bold lines representing the sample medians);
- variability differences (compare the box heights containing the central 50% of the data);
- skewness (look at the whiskers and at the location of the median in the box);
- outliers (look at single points).

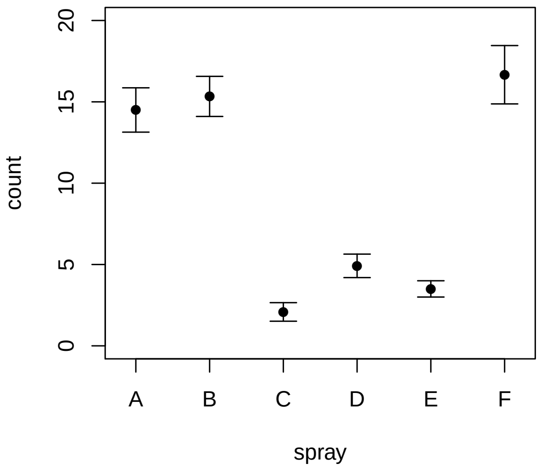
This box plot should not be over-interpreted because the boxes contain only twelve points each. Still the box plot indicates clear location differences, some differences in variability, some skewness and two outliers for treatments C and D.

[5] In `ggplot2`, the default box plots do not have horizontal lines at the whiskers. These have to be added on their own.

A third popular family of graphs in this context plots the treatment means with their *standard errors*.[6] They are implemented for example in the `sciplot` package. Two variants are obtained with:

```
> library(sciplot)
> bargraph.CI(spray, count, col = (gray(0.88)), data = InsectSprays,
+             xlab = "spray", ylab = "count", ylim = c(0,20))
> lineplot.CI(spray, count, type = "p", data = InsectSprays,
+             xlab = "spray", ylab = "count", ylim = c(0,20))
```





Both plots contain the same information: the means ± their standard errors are plotted. Any outliers, asymmetry and so on are glossed over, so the "information per space used" ratio of this plot is not so high. It is popular because the standard error of the mean is directly related to the length of the confidence interval for the mean, see below.[7]

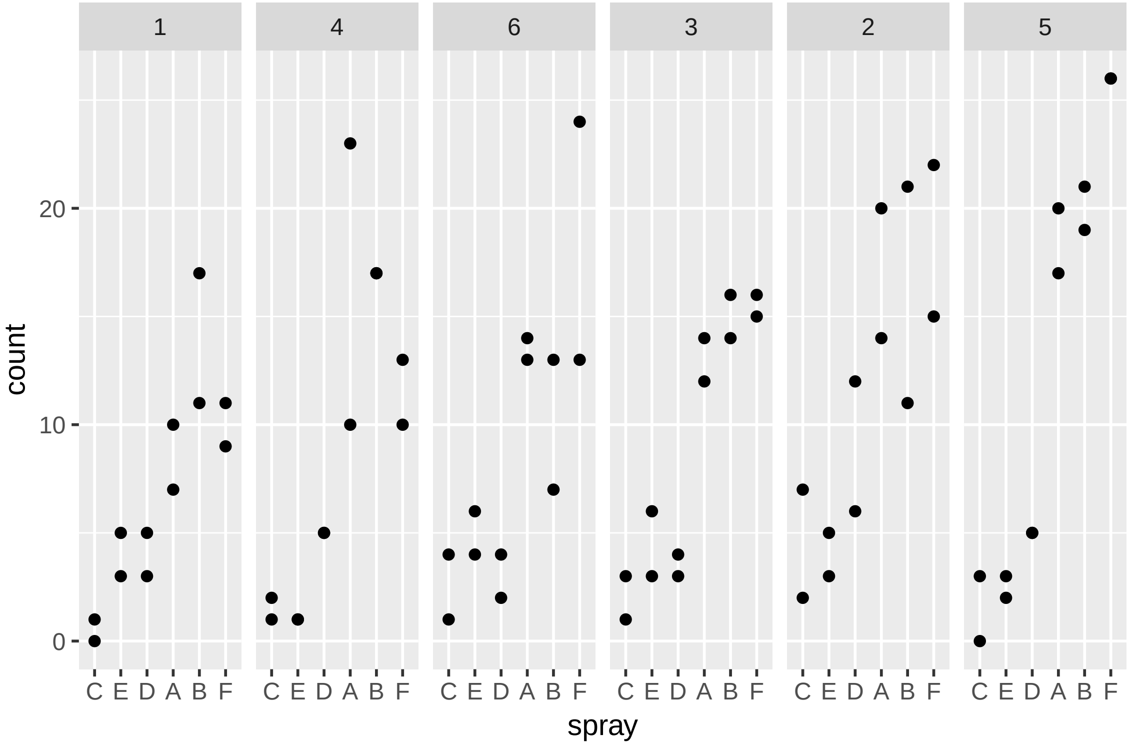
2.1.4 Block structure of the insect sprays data set

The `InsectSprays` data come from a blocked experiment (the `R` help does not mention this, I found out the block structure by reading the original publication.) Here is a plot of the data taking the six blocks into account, additionally ordering the blocks and treatments by count means for better overwiew:

[6] The *standard error* of a statistical estimator is the estimated value of its standard deviation. For a sample of n independent observations with sample standard deviation s, the standard error of the mean is $s/\sqrt{n}$.

[7] With a little work, it is possible to directly plot confidence intervals for the treatment means – can you find out how?

```
> InsectSprays$block <- factor(rep(rep(1:6, each=2), times = 6)) ## design
```



We treat the data as coming from a CRD in order to have a simple example for didactic purposes, but this is statistically not a correct analysis because the following two important questions are ignored:

- Do blocks themselves have an effect?
- Do treatment effects depend on the blocks?

For me, it is not visually clear whether the block effects are important. The second question is difficult to assess with only two observations for each treatment in each block. We will see how to answer these questions when we discuss blocked data.

2.1.5 Notation and independence assumptions

Treatment			
1	2	$\cdots$	k
Y_{11}	Y_{21}	$\cdots$	Y_{k1}
Y_{12}	Y_{22}	$\cdots$	Y_{k2}
$\vdots$	$\vdots$		$\vdots$
Y_{1n_1}	$\vdots$		Y_{kn_k}
	Y_{2n_2}		

The n_j observations of the dependent variable Y from treatment j are denoted by $Y_{j1}, Y_{j2}, \ldots, Y_{jn_j}$ for each $j = 1, \ldots, k$. The treatment index is first, then comes the plot index. The treatment sample sizes $n_1, \ldots, n_k$ need not be equal. We write $n = n_1 + \ldots + n_k$ for the total sample size.

To define a statistical model for the completely randomized design, assumptions on *independence* and *distributional* assumptions are made.

If you are not willing to assume anything, no statistical analysis is possible because the model for the data is then too flexible. In other words, some things need to remain the same, such that you have more than one observation measured under comparable conditions and can start fitting a model.

We assume that observations are *mutually independent*. Remember that by definition, the random variables $Y_1, \dots, Y_n$ are called mutually independent if for any $k \leq n$ and $y_1, \dots, y_k \in \mathbb{R}$, we have that

$$\mathbf{P}(Y_1 \leq y_1, \dots, Y_k \leq y_k) = \mathbf{P}(Y_1 \leq y_1) \cdots \mathbf{P}(Y_k \leq y_k)\,.$$

For a simple example, set $k = 2$. Then, we require that for all (y_1, y_2), the probability of the event that $(Y_1 \leq y_1,\, Y_2 \leq y_2)$ must be the product of the probabilities of $(Y_1 \leq y_1)$ and $(Y_2 \leq y_2)$. In other words, Y_1 is not allowed to have any influence on Y_2, and vice versa.

It is impossible to establish independence with a statistical test, the assumption can only be justified with a good study design which produces observations that can be treated as independent.

We also assume that all the observations with the same treatment have the same distribution. If those two assumptions are not fulfilled (e. g. because of blocks), we need different models to account for it. The parametric and nonparametric methods discussed below differ with respect to the assumptions on the distribution of the outcome variable.

2.2 Parametric models: one-way ANOVA

2.2.1 Distributional assumptions

Classical one-way ANOVA assumes that the measurements for each treatment j follow a normal distribution with mean μ_j and variance σ^2, symbolized by

$$Y_{ji} \sim \mathcal{N}(\mu_j, \sigma^2)$$

for all treatments $j = 1, \dots, k$ and all plots $i = 1, \dots, n_j$. The population treatment means $\mu_1, \dots, \mu_k \in \mathbb{R}$ and the variance $\sigma^2 > 0$ are assumed to be fixed, unknown numbers. The variance is assumed to be the same for each treatment.

The assumption of equal variances is made for several reasons. If simplifies the statistical analysis and means that only $k + 1$ parameters have to be estimated. However, it is by no means always satisfied for real data. We come back to this important point in Section 4.2.2.

The mathematical expectation (theoretical average) of Y_{ji} is μ_j. One-way ANOVA models the expectation of the outcome variable, conditional on the treatment. This is often stated as modeling the *conditional expectation* of Y. The equality of the variances means that the *shape* of the distributions remains precisely the same in for all treatments. By choosing different means, one can only shift the *location*. 什么意思?

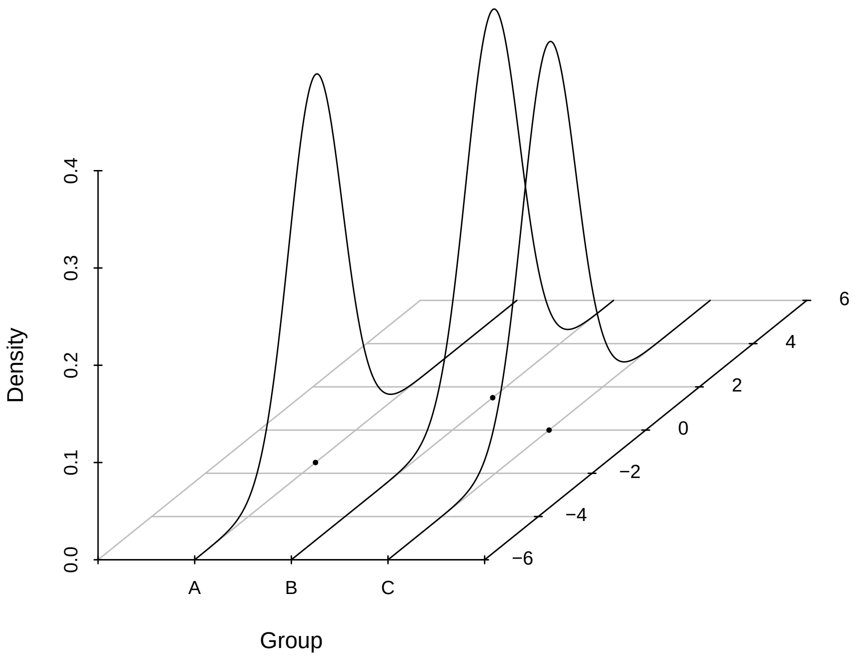


Figure 1: Visualizing the classical ANOVA assumptions

The model is visualized in Figure 1 by plotting the density functions of three treatments, $\mathcal{N}(-\frac{3}{2}, 1)$, $\mathcal{N}(\frac{3}{2}, 1)$ and $\mathcal{N}(0, 1)$. Expectations are indicated with black dots.

2.2.2 The ANOVA table and the overall F test

We test the null hypothesis that all theoretical treatment means are the same,

$$H_0 : \mu_1 = \ldots = \mu_k = \mu$$

against the alternative that at least two theoretical treatment means differ.

Under the null hypothesis, μ should be estimated as the mean of all observations, while under the alternative the means μ_j for each treatment should be estimated by using *observations from that treatment only:*

$$\hat{\mu}_j = \bar{Y}_j = \frac{1}{n_j}\sum_{i=1}^{n_j} Y_{ji} \quad \text{for all } j\,.$$

With $\bar{Y}$ denoting the grand mean,

$$(Y_{ji} - \bar{Y}) = (Y_{ji} - \bar{Y}_j) + (\bar{Y}_j - \bar{Y})\,.$$

Squaring the above equation, summing over all observations and a little algebra shows that

$$\underbrace{\sum_{j=1}^{k}\sum_{i=1}^{n_j}(Y_{ji} - \bar{Y})^2}_{\text{total sum of squares SST}} = \underbrace{\sum_{j=1}^{k}\sum_{i=1}^{n_j}(Y_{ji} - \bar{Y}_j)^2}_{\text{within sum of squares SSW}} + \underbrace{\sum_{j=1}^{k}\sum_{i=1}^{n_j}(\bar{Y}_j - \bar{Y})^2}_{\text{between sum of squares SSB}} \quad .$$

The total sum of squares SST is the sum of the within sum of squares SSW and the between sum of squares SSB, in symbols

$$\text{SST} = \text{SSW} + \text{SSB}\,. \tag{1}$$

There are two (only theoretically possible) extreme cases. It could be that

- SSB $= 0$ (i. e. that SST $=$ SSW). This happens only if all *sample* treatment means are equal to the overall mean. This should never happen for a real data set, *even if the null hypothesis is true* (that is, if the *theoretical* means were all the same).
- SSW $= 0$ (i. e. that SST $=$ SSB). This is the case if within every treatment, all the values are equal to the respective treatment mean.

Real data will always be somewhere in between these extreme cases. *If the null hypothesis is true, SSB should be small, otherwise it is big.*
Recall from the preparatory work that SST$/(n-1)$ is the sample variance, which is our estimate of the variance σ^2 of the error terms if the data all come from the same treatment. Also recall that SSW$/(n-k)$ is our estimate of σ^2 if the data come from separate treatments.
To quantify this and to summarize the situation, one traditionally builds an *ANOVA table.* Calling the treatment variable `factor` (with k levels), the table looks as follows:

这个表
比较重要

	Df	Sum Sq.	Mean Sq.	F value	p value
`factor`	$k-1$	SSB	MSB $=$ SSB$/(k-1)$	MSB/MSW	$\mathbf{P}(F_{k-1,n-k} > F)$
Residuals	$n-k$	SSW	MSW $=$ SSW$/(n-k)$		

The first column (Df) is called the *degrees of freedom*. It is used to keep track of the number of treatments and the overall sample size. The next two columns are called the

F 值：MSB/MSW

sums of squares and the *mean sums of squares*. Note that MSB quantifies the variability between treatment means. Next comes the F *value*. Because $F = \text{MSB}/\text{MSW}$, big values of F are evidence against the null hypothesis of equal treatment means.
In the last column we write $\mathbf{P}(F_{k-1,n-k} > F)$ to denote the probability of a random variable with an $F_{k-1,n-k}$ distribution exceeding the observed value F. The bigger the observed F value, the smaller this probability.
To obtain the ANOVA table for the insect counts example with R, use:

```
> ins.lm <- lm(count ~ spray, data = InsectSprays)
> anova(ins.lm)

# Analysis of Variance Table
#
# Response: count
#           Df Sum Sq Mean Sq F value    Pr(>F)
# spray      5 2668.8  533.77  34.702 < 2.2e-16 ***
# Residuals 66 1015.2   15.38
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

两种方法可以获得anova表
1: lm, anova()
2: aov
summary()

First the `lm` command is used to fit a `l`inear `m`odel.[8] Next `anova` is used to produce the ANOVA table based on the model. The function `aov` may also be used for this task. We will encounter it again later and discuss it there; for the moment, it produces the same results.

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

If the alternative is true, MSB is big and F will have big values. How big does F have to be in order to reject the null hypothesis with a given significance level?
The *overall* F *test* provides the answer. If the model assumptions hold, then under the null hypothesis, the F test statistic has an F distribution. The F distribution has to keep track of both the number of treatments and the number of observations, that is why it has two degrees of freedom which must be given in the correct order. In general, we write $F \sim F_{\nu_1,\nu_2}$. In one-way ANOVA, the F statistic has an $F_{k-1,n-k}$ distribution. For our data, this means that F has an $F_{5,66}$ distribution. The p value is the probability of an $F_{k-1,n-k}$ distribution being at least as extreme as in our sample.[9]

[8] As will be seen later, ANOVA is a special linear model. Calculating F only involves computing means and sums of squares and can easily be done without special software. For the p value, you need the distribution function of F.

[9] To obtain it, use $\mathbf{P}(F_{5,66} > x) = 1 - \mathbf{P}(F_{5,66} \leq x)$, in R: `1 - pf(34.702, 5, 66)`.

If the p value of the F test is smaller than the significance level, reject the null hypothesis and conclude that not all the treatments have the same theoretical means.

For our data, F is so big and p so small that in the ANOVA table, only the statement `<2.2e-16` is given for p, this means $p < 2.2 \times 10^{-16}$. The null hypothesis is thus rejected and we conclude that there are significant differences between the theoretical mean number of surviving insects for some of the treatments.

Before trying to find out *which* treatments differ from each other, remember *that the p value is only valid if the model assumptions hold.*

Assumption checking (also called *model diagnostics*) is a set of techniques to detect violations of assumptions that could possibly invalidate our p values. One difficulty in applied work is to judge which amount of not fulfilling the assumptions is still tolerable. We discuss this in Section 4.

2.2.3 Residuals, RMSE, and the R^2

The ANOVA model can be written as $Y_{ji} = \mu_j + \varepsilon_{ji}$, where μ_j is the mean for treatment j and the *error* terms ε_{ji} have a $\mathcal{N}(0, \sigma^2)$ distribution. We can not directly observe μ_j or ε_{ji}.

The *residuals* are defined as

$$e_{ji} = Y_{ji} - \hat{Y}_{ji}$$

where $\hat{Y}_{ji}$ is the fitted value for observation Y_{ji}.

In ANOVA, the fitted value $\hat{Y}_{ji}$ is the treatment mean $\bar{Y}_j$ (estimating μ_j) for any observation from treatment j. The residuals may thus be interpreted as "estimates" of the *errors* ε_{ji}.[10] The way to memorize this is

$$\text{residuals} = \text{observed values} - \text{fitted values}\,.$$

From the residuals, we derive two important quantities. First of all, rewrite SSW as follows (because $\bar{Y}_j = \hat{Y}_{ji}$):

$$\text{SSW} = \sum_{j=1}^{k} \sum_{i=1}^{n_j} (Y_{ji} - \hat{Y}_{ji})^2 = \sum_{j,i} e_{ji}^2\,.$$

As can be shown by calculation or geometric arguments, MSW $=$ SSW$/(n-k)$ is an unbiased estimate of the variance σ^2. From it, we define the *root mean square error* (RMSE) as RMSE $= \sqrt{\text{MSW}}$.

[10] Residuals are accessed with `resid(mod)` where `mod` denotes a model object.

> The *root mean square error* (RMSE) estimates the standard deviation σ of our observations.

For the insect sprays, RMSE $= \sqrt{15.38} = 3.922$. The summary function computes this, here is how to quickly get it:

```
> summary(ins.lm)$sigma

# [1] 3.921902
```

在R分析中，sigma就是RMSE

Divide (1) by SST, then

$$1 - \frac{\text{SSW}}{\text{SST}} = R^2 \,.$$

> The *coefficient of determination* R^2 is the proportion of the total variance of the dependent variable which can be "explained" by the model.

For our model of the insect sprays, $R^2 = 0.7244$. The numeric value may be obtained with

```
> summary(ins.lm)$r.squared
```

The closer to one this proportion is, the higher the contribution of our factor to "explaining" the variance. What constitutes sufficiently high values of R^2 is not a mathematical question, it depends on the phenomenon under study.
In the three bottom lines of the output of `summary(ins.lm)`, you find the root mean square error (called "residual standard error" here), the coefficient of determination (called "multiple R^2" here) and also the F statistic with its degrees of freedom and the p value.

Technical remark

The use of the `lm` command produces so-called *least-squares* estimators of the treatment means, which are chosen such that the sum of the squared residuals SSW becomes as small as possible. These estimators have attractive theoretical properties if the equal variance assumption is fulfilled.

2.2.4 Estimating and testing group means

One minor drawback of using `lm` in our setting is that its default settings (explained below) may not be what we want here. Because we have no treatment structure, we should not prefer any treatment in the parametrization. One solution is to directly estimate group means (*cell means model*):

```
> ins.lm.noint <- lm(count ~ spray - 1 , data = InsectSprays)
> summary(ins.lm.noint)$coefficients
```

t value是什么?

```
#          Estimate Std. Error   t value     Pr(>|t|)
# sprayA 14.500000   1.132156 12.807428 1.470512e-19
# sprayB 15.333333   1.132156 13.543487 1.001994e-20
# sprayC  2.083333   1.132156  1.840148 7.024334e-02
# sprayD  4.916667   1.132156  4.342749 4.953047e-05
# sprayE  3.500000   1.132156  3.091448 2.916794e-03
# sprayF 16.666667   1.132156 14.721181 1.573471e-22
```

删除间距intercept

with the term `- 1` being `R` formula syntax to remove the intercept (you may also write `+ 0` instead of `- 1`). The model `summary` contains in each line the information from a two-sided t test to test whether the respective group mean is zero.
In more detail, the `Estimate` is the estimated group mean, if we divide it by its estimated standard deviation, called the *standard error*, we obtain the `t value`, which has a t distribution unter the null hypothesis that the true coefficient is zero. A two-sided p value is given in the rightmost column.

2.3 Pairwise treatment comparisons

In general, if the model assumptions are met, a model for the whole population makes more precise predictions than a model for subgroups; because we can use more data to estimate σ^2, its estimate is more precise than with pairwise treatment comparisons.
For that reason, a *top-down approach* is often used in statistics. First one estimates a "big" model and then looks at subquestions. In our case, the overall F test is performed first and pairwise treatment comparisons after that.

2.3.1 The multiple testing problem

If you pairwisely compare k treatments, you conduct $m = \binom{k}{2} = \frac{k(k-1)}{2}$ hypothesis tests. If m hypotesis tests are performed, each at level α, then the probability of committing at least one Type I error in all the tests together is not bounded by $1-(1-\alpha) = \alpha$, but by a bigger number. In case the tests are independent, this probability is $1-(1-\alpha)^m$. With six treatments (as in the insect sprays data), there are $\binom{6}{2} = 15$ pairwise comparisons and $1 - 0.95^{15} = 0.54$.

> Suppose you perform a family of hypothesis tests. The probability of committing at least one Type I error *in all the tests together* is called the *familywise error rate*. The increase of the familywise error rate with the number of tests is called the *multiple testing* problem.

2.3.2 Directly adjusting a set of p values

One solution of the multiple testing problem is to adjust the significance levels in the individual tests. The idea is to be more strict in the individual tests so that the familywise error rate is lowered as well. This is implemented in the `p.adjust` function. It features several adjustment methods, the `holm` method is a good general method. This stepwise procedure is guaranteed to control the familywise error rate, but is less conservative than e. g. the Bonferroni method.

2.3.3 Pairwise t and Wilcoxon tests

R has convenience functions to conduct pairwise comparisons, adjust the p values and report results. We show how to apply pairwise t-tests (numeric results not shown) and pairwise Wilcoxon rank sum tests to the insect sprays data set.[11]

```
> ## with(InsectSprays, pairwise.t.test(count, spray, "holm"))
> with(InsectSprays, pairwise.wilcox.test(count, spray, "holm"))

#
#  Pairwise comparisons using Wilcoxon rank sum test
#
# data:  count and spray
#
#   A       B       C       D       E
# B 1.00000 -       -       -       -
# C 0.00051 0.00051 -       -       -
# D 0.00062 0.00062 0.01591 -       -
# E 0.00051 0.00051 0.26287 0.69778 -
# F 1.00000 1.00000 0.00051 0.00062 0.00051
#
# P value adjustment method: holm
# Warning message:
# In wilcox.test.default(xi, xj, paired = paired, ...) :
#   cannot compute exact p-value with ties
```

For this example, the results of pairwise t tests (not shown here) and pairwise Wilcoxon tests are quite different for some categories, a consequence of violating test assumptions.

> If one doubts the normality assumption, it is safer to use the pairwise Wilcoxon tests or another nonparametric or robust procedure.

[11] We have not forgotten that the equal variance assumption was not satisfied and in real analysis would consider using a different approach.

The matrix gives the adjusted p values so that the familywise error rate is 0.05. For example, the comparison of sprays C and D gives a Holm-corrected p value of 0.016.

The Wilcoxon test function gave warnings because it detected ties (different observations having the same value of the dependent variable) in the data, which affects the calculation of p values. One solution is to perform an *exact* Wilcoxon rank sum test, which accounts for the pattern of the ties. This is implemented in the `coin` and `exactRankTests` packages. A full solution is available on the course web site.

2.3.4 Tukey's honest significant difference (HSD)

We now show a method which is a bit more refined than just running all pairwise comparisons. Tukey's HSD relies on normality, so only use it when normality and also roughly similar variances are acceptable assertions. Also, the method is sensitive to outliers and conservative if not all treatments have the same number of observations, so perhaps use another method in that case.
Consider the following two methods to obtain it:

```
> TukeyHSD(ins.aov, "spray")
> library(agricolae)
> HSD.test(ins.lm, "spray", group = TRUE, console = TRUE)
```

The first method uses the `TukeyHSD` function and needs an `aov` object (not an `lm` object). The output consists of confidence intervals for all pairwise differences and a column with adjusted p values, `p adj`.

```
           diff        lwr       upr     p adj
B-A   0.8333333  -3.866075  5.532742 0.9951810
C-A -12.4166667 -17.116075 -7.717258 0.0000000
>snip<
E-D  -1.4166667  -6.116075  3.282742 0.9488669
F-D  11.7500000   7.050591 16.449409 0.0000000
F-E  13.1666667   8.467258 17.866075 0.0000000
```

Only the differences with `p adj` $< \alpha$ are significant.

The second method uses the `agricolae` function `HSD.test`. Its output contains:

```
Treatments with the same letter are not significantly different.

  count groups
F 16.67      a
B 15.33      a
```

```
A 14.50      a
D  4.92      b
E  3.50      b
C  2.08      b
```

The `Treatments` that have the same `groups` letter are not statistically different (they are in a group of comparable treatments); treatments that do not share a letter are statistically different.

This way of communicating results is called a *compact letter display* (CLD) and often used in agronomy, where the letters are added to a bar plot above the bars (one bar for each treatment). Sometimes treatments are in more than one group and have more than one letter. CLDs may be used after any groupwise comparison procedure.

3 Completely randomized designs with control

Let us now suppose that one among the treatments is special, we call it the *control* treatment; an example is the standard treatment in clinical studies.

3.1 Comparing all treatments with the control

Assume that treatment 1 is the control treatment and the aim is to compare all the other treatments to it (but not among themselves). If we denote by μ_j the mean of treatment j, then the aim is to estimate $\mu_j - \mu_1$ and to test whether $\mu_j - \mu_1 = 0$ for $j = 2, \ldots, k$.
There are several ways to accomplish this in `R`. The simplest way is to redefine the control treatment as the *reference level* of the factor in question, to fit the model with `lm` and to look at its `summary`.

The following *treatment contrasts* are the parametrization chosen by `lm` by default.

Technically, `R` uses what it calls *contrasts* to encode factor levels. Behind the scenes, `R` runs a *multiple linear regression model* to which it adds suitably coded auxiliary variables. The details of this coding are important to interpret the results correctly. Recall that we are in a setting with k treatments. We introduce the following auxiliary variables:

- $X_1 = 1$ if the treatment is T_1, $X_1 = 0$ otherwise;
- $X_2 = 1$ if the treatment is T_2, $X_2 = 0$ otherwise;
- $\ldots$
- $X_k = 1$ if the treatment is T_k, $X_k = 0$ otherwise.

Clearly, one of these variables is redundant, since if for example $X_2 = \ldots = X_k = 0$, we know that $X_1 = 1$. What `lm` does by default is to omit variable X_1 from the regression model, so that the model equation becomes

$$\mathrm{E}(Y \mid X_2 = x_2, \ldots, X_k = x_k) = \alpha + \beta_2 x_2 + \ldots + \beta_k x_k \,.$$

If an observation obtained treatment T_1, we have $x_2 = 0, \ldots x_k = 0$. If it obtained any other treatment T_j, then x_j and only x_j is equal to one. In other words,

$$\begin{aligned}\mu_1 &= \mathrm{E}(Y \mid X = T_1) = \alpha \,,\\ \mu_j &= \mathrm{E}(Y \mid X = T_j) = \alpha + \beta_j \,, \quad j = 2, \ldots, k \,.\end{aligned}$$

We find that $\alpha = \mu_1$, while $\beta_j = \mu_j - \mu_1$ for $j = 2, \ldots, k$. In other words, *if we use treatment contrasts*, then the intercept estimates the mean of the reference treatment, while all the other coefficients estimate differences of the respective treatment to the mean of the reference treatment.
It is possible to look at the contrasts that R uses for a particular variable:

```
> contrasts(InsectSprays$spray)

#   B C D E F
# A 0 0 0 0 0
# B 1 0 0 0 0
# C 0 1 0 0 0
# D 0 0 1 0 0
# E 0 0 0 1 0
# F 0 0 0 0 1
```

Each row of the contrast matrix corresponds to one treatment and tells us how the treatment is represented by the auxiliary variables.
R chooses (alphabetically by default) one reference level whose estimate you find in the first line below labeled as `(Intercept)`.[12]

```
> summary(ins.lm)$coefficients

#              Estimate Std. Error t value  Pr(>|t|)
# (Intercept)  14.5000      1.132 12.8074 1.471e-19
# sprayB        0.8333      1.601  0.5205 6.045e-01
# sprayC      -12.4167      1.601 -7.7550 7.267e-11
# sprayD       -9.5833      1.601 -5.9854 9.817e-08
# sprayE      -11.0000      1.601 -6.8702 2.754e-09
# sprayF        2.1667      1.601  1.3532 1.806e-01
```

The estimated mean number of insects surviving a treatment with the reference level (Spray A) is $\hat{\mu}_A = 14.5$. *All the other coefficients (the numbers in the* ***`Estimate`*** *column) are differences $\hat{\mu}_j - \hat{\mu}_A$ to the mean of the reference level, A.*

The estimated coefficient for Spray B is 0.8333, which means that its mean number of surviving insects is found as $14.5 + 0.8333 = 15.3333$.

[12] The reference level is the one which does not appear with a name in the `Coefficients` block. If you know that the levels of `spray` are A through F, it is clear that A is the reference level. To choose the B spray as control treatment, we would use `InsectSprays$spray.b <- relevel(InsectSprays$spray, ref = "B")`, then fit the model using the factor `spray.b`. 改变对照组为B

> For levels which are not the reference level, *positive coefficients imply that the dependent variable has a higher mean than the reference treatment*, while negative coefficients imply that the mean is lower than in the reference treatment.

For example, Spray C has a mean of $14.5 - 12.4167 = 2.0833$ surviving insects. To control, look at the sample treatment means again:

```
> with(InsectSprays, tapply(count, spray, mean))

#      A      B      C      D      E      F
# 14.500 15.333  2.083  4.917  3.500 16.667
```

The other columns of the `summary` output contain the standard errors of the estimated coefficients and a t test of the null hypothesis that the corresponding coefficient is zero versus the two-sided alternative. As a result of the reference level parametrization, the interpretations of the t tests above are different. For the reference level, it is tested whether $\mu_{\text{reference}} = 0$, i. e. whether its mean is zero or not. For all other coefficients, it is tested whether $\mu_j - \mu_{\text{reference}} = 0$, i. e. whether the difference of the respective level to the reference level is significant.

3.2 More about contrasts

`R` supports two types of factors: unordered factors (with nominal levels) and ordered factors (with ordinal levels). We can see what contrasts are used for each of these two types as follows

```
> options("contrasts")

# $contrasts
#         unordered           ordered
# "contr.treatment"      "contr.poly"
```

We see that `R` uses treatment contrasts for unordered factors and polynomial contrasts for ordered factors by default.

3.2.1 Sum contrasts

每个处理的平均值和总体平均值（grand mean）的差别

Often, it is not so interesting to ask whether the group means are zero or not. In the unstructured treatment setting, it sometimes makes sense to ask whether the treatments differ from the *grand mean*, the average of all the treatment means. To achieve this, we can switch to using *sum contrasts*. There are at least three ways of increasing persistence in which you can do this:

1. Pass the constrasts only to the `lm` call.
   E. g. `lm(count ~ spray, data = InsectSprays, contrasts = list(spray = "contr.sum"))`

2. Permanently define contrasts for a particular factor.
   E. g. `contrasts(InsectSprays$spray) <- "contr.sum"`

3. Change the `contrast` option globally.
   E. g. `options(contrasts = c("contr.sum", "contr.poly"))`

Depending on your setting, each of these may make sense. In any case, do not forget to change back if you used the third option.

```
> ins.lm.sum <- lm(count ~ spray, data = InsectSprays,
+                          contrasts = list(spray = "contr.sum"))
> summary(ins.lm.sum)$coefficients

#              Estimate Std. Error t value  Pr(>|t|)
# (Intercept)     9.500     0.4622  20.554 2.161e-30
# spray1          5.000     1.0335   4.838 8.224e-06
# spray2          5.833     1.0335   5.644 3.778e-07
# spray3         -7.417     1.0335  -7.176 7.867e-10
# spray4         -4.583     1.0335  -4.435 3.572e-05
# spray5         -6.000     1.0335  -5.805 2.004e-07
```

To understand the output, it helps to look at the *contrast matrix*

```
> InsectSprays$spray.sum <- InsectSprays$spray
> contrasts(InsectSprays$spray.sum) <- "contr.sum"
> contrasts(InsectSprays$spray.sum)

#   [,1] [,2] [,3] [,4] [,5]
# A    1    0    0    0    0
# B    0    1    0    0    0
# C    0    0    1    0    0
# D    0    0    0    1    0
# E    0    0    0    0    1
# F   -1   -1   -1   -1   -1
```

Because we changed the contrasts, the interpretation of the coefficients changes. The `Intercept` is now an estimate of the overall mean (over all the treatments). On average (over all the treatments), 9.5 insects survive. With Spray A, 5.0 more insects survive than the grand mean, etc. Spray E kills 6 more insects than the grand mean. Spray F is the reference and accordingly suppressed by `R` here (but see below). Other types of contrasts often used for ordinal factors are *Helmert* contrasts and *orthogonal polynomial* contrasts. We omit these here.

3.3 Specific comparisons

Sometimes we want to test specific hypotheses, such as in a dose-response study, where one only compares each dose to the next higher dose. We can either have only one such hypothesis or several at the same time, such that the multiple testing aspect should not be forgotten about.

Reducing the number of comparisons is attractive because it makes the p value correction less strict, such that there is a gain in power.

Suppose you have six treatments and a) compare them all pairwisely, b) compare them in one specific order. Then you have $\binom{6}{2} = 15$ comparisons in a) but only $6 - 1 = 5$ in situation b). As a result, the multiple testing p value correction in a) is stricter than in b), and it is more difficult to obtain a significant result in a).
Some popular variants of specific comparisons are implemented in the `multcomp` library. Four methods are often used:

Method	Description
`Dunnett`	Compare all levels to a reference level
`GrandMean`	Compare all levels to the overall mean
`Sequen`	Test a specific sequence
`Tukey`	Tukey's HSD

The default of `multcomp` is to adjust for multiple testing.

```
> library(multcomp)
> summary(glht(ins.lm, mcp(spray = "Dunnett")))

#
#   Simultaneous Tests for General Linear Hypotheses
#
# Multiple Comparisons of Means: Dunnett Contrasts
#
#
# Fit: lm(formula = count ~ spray, data = InsectSprays)
#
# Linear Hypotheses:
```

```
#             Estimate Std. Error t value Pr(>|t|)
# B - A == 0    0.833      1.601    0.52     0.98
# C - A == 0  -12.417      1.601   -7.76   <1e-04 ***
# D - A == 0   -9.583      1.601   -5.99   <1e-04 ***
# E - A == 0  -11.000      1.601   -6.87   <1e-04 ***
# F - A == 0    2.167      1.601    1.35     0.53
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# (Adjusted p values reported -- single-step method)
```

Compared to the **summary** in Section 3.1, the estimates and their standard errors are the same, but due to the adjustment for the multiple testing, the p values obtained now are higher. It is possible to disable the p value correction.

To test one or several specific hypotheses, the easiest way is to give a contrast matrix in which each row encodes one hypothesis. For example, we could ask whether the difference of treatments B and F to treatment A is the same, $\mu_B - \mu_A = \mu_F - \mu_A$. This amounts to fitting the model with treatment contrasts and leaving A as the reference level for the spray. Then, we simply need to test $\beta_B - \beta_F = 0$. We rewrite this such that each parameter gets a weight:

$$0\alpha + 1\beta_B + 0\beta_C + 0\beta_D + 0\beta_E - 1\beta_F = 0$$

We extract the weights and pass them to **glht**:

```
> K <- matrix(c(0, 1, 0, 0, 0, -1), nrow = 1)
> summary(glht(ins.lm, linfct = K))

#
#   Simultaneous Tests for General Linear Hypotheses
#
# Fit: lm(formula = count ~ spray, data = InsectSprays)
#
# Linear Hypotheses:
#         Estimate Std. Error t value Pr(>|t|)
# 1 == 0    -1.33       1.60   -0.83     0.41
# (Adjusted p values reported -- single-step method)
```

The difference is higher by 1.33 units for Spray F, but this difference is not significant. To finish, we show how to use **glht** to test for each treatment whether its mean is equal to the grand mean, *while adjusting for multiple testing* (which we did not do above with the sum contrasts).

```
> summary(glht(ins.lm, mcp(spray = "GrandMean")))

#
#   Simultaneous Tests for General Linear Hypotheses
#
# Multiple Comparisons of Means: GrandMean Contrasts
#
#
# Fit: lm(formula = count ~ spray, data = InsectSprays)
#
# Linear Hypotheses:
#          Estimate Std. Error t value Pr(>|t|)
# C 1 == 0     5.00       1.03    4.84  < 1e-04 ***
# C 2 == 0     5.83       1.03    5.64  < 1e-04 ***
# C 3 == 0    -7.42       1.03   -7.18  < 1e-04 ***
# C 4 == 0    -4.58       1.03   -4.43  0.00021 ***
# C 5 == 0    -6.00       1.03   -5.81  < 1e-04 ***
# C 6 == 0     7.17       1.03    6.93  < 1e-04 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# (Adjusted p values reported -- single-step method)
```

All the differences to the grand mean are significant (which is not surprising, given that we have three good and three bad treatments). The familywise error rate of this procedure is 5%.

4 Model diagnostics and alternative analysis methods

4.1 The normality assumption

4.1.1 Graphical normality checking

A first, graphical approach to assess normality uses *normal quantile-quantile (QQ) plots*. The idea is to visually judge whether the residuals come from a normal distribution by comparing the quantiles of the residual distribution with the quantiles of a standard normal distribution.

More precisely, the plot is defined as follows in `R` (for $n > 10$): sort the n residuals such that $e_{(1)} \leq e_{(2)} \leq \ldots \leq e_{(n)}$. Define the points $x_i = \Phi^{-1}((i - 0.5)/n)$ for $i = 1, \ldots, n$ where Φ^{-1} is the quantile function of the standard normal distribution. Now plot the points (x_i, e_i) for $i = 1, \ldots, n$. (The correction by -0.5 improves results a little.)
Nice QQ plots are produced with

```
> library(car)
> qqPlot(resid(ins.lm))
```

The output is given in Figure 2.[13]

> If the normality assumption holds, the points in the normal QQ plot should lie "closely" around the straight line given in the normal QQ plot. I recommend not to use normal QQ plots for less than around 30 observations.

The dashed lines in the `qqPlot` output serve as an indication of what "closely" means. They are (pointwise) 95% confidence intervals for the respective points. Not too many points should fall outside the dashed lines.
Judging a QQ plot takes some experience. Here, the points in the left tail seem to systematically lie below the line. This indicates that the left tail (small values) of the residual distribution is a bit heavier than the left tail of a normal distribution. The values in the right tail of the QQ plot are a bit above the line. This indicates that the right tail of our residuals (big values) is a bit too heavy. In summary, the negative residuals are a bit too big in absolute value and the positive residuals are also too big. In other words, the residual distribution is more spread out than it should be.
Based on my personal experience, this amount of non-normality is of middle severity (especially the right tail) and we should definitely keep an eye on the too heavy tails. To substitute my experience, run the command

[13] A slightly different variant based on studentized residuals may be obtained with `qqPlot(ins.lm)`.

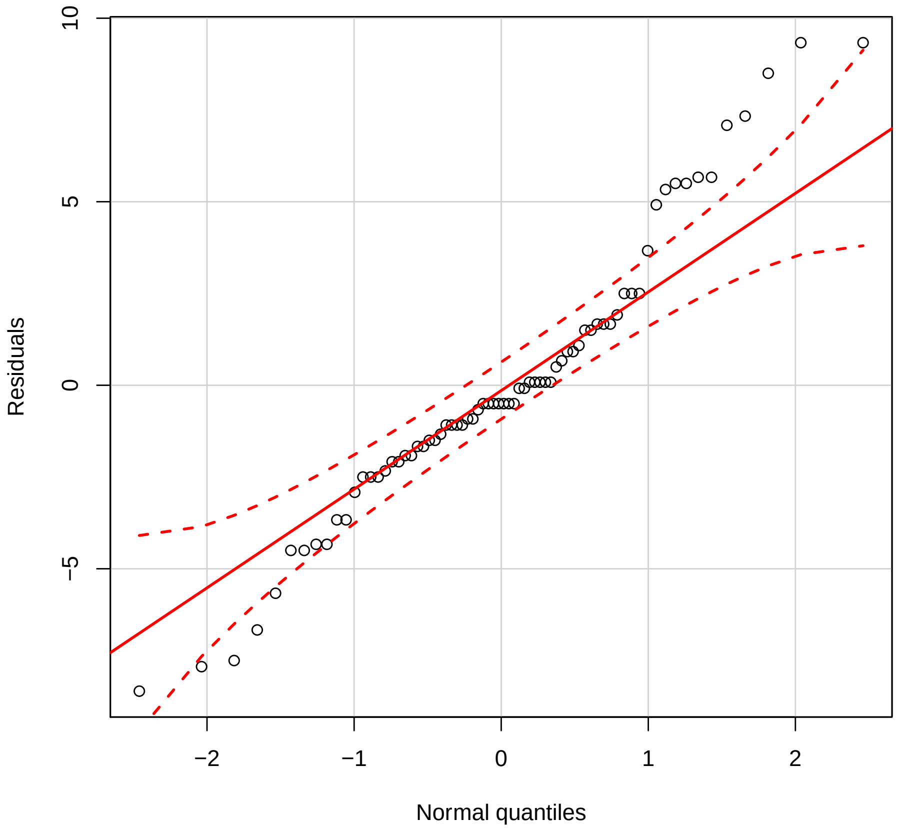


Figure 2: The QQ plot of the insect sprays ANOVA residuals

```
> qqPlot(rnorm(72, mean = mean(resid(ins.lm)), sd = sd(resid(ins.lm))))
```

maybe twenty times or so. Every time a QQ plot of 72 *simulated* normally distributed values (with mean and standard deviation as for the residuals) is produced.
This gives you an idea of what the QQ plot *should* look like and what constitutes usual fluctuations. Most of the plots should be better behaved than the plot for the insect sprays. Every now and then, some points will fall outside the dashed lines by chance, mostly in the tails of the distribution.

4.1.2 Testing for normality

Because judging QQ plots is somewhat subjective, we need a more objective and formal tool to assess normality. There are a number of different statistical tests for normality. We use the *Shapiro-Wilk* test and reject the null hypothesis of normality if $p < \alpha$. Performing the test

```
> shapiro.test(resid(ins.lm))

#
#  Shapiro-Wilk normality test
#
# data:  resid(ins.lm)
# W = 0.96, p-value = 0.02
```

yields a p value of 0.022.

> Because $p < 0.05$, we reject the null hypothesis of normality and conclude that the residuals do not come from a normal distribution.

We also know what the problem is: the tails are too heavy.

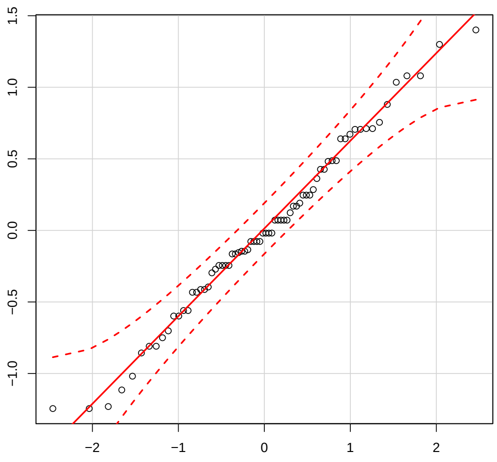
It is very difficult to detect non-normality in small samples (any normality test has low power against many alternatives in small samples), so that some statisticians discourage the use of normality tests altogether. They argue that the danger is that researchers do not find the nonnormality due to the small sample and falsely conclude that they are safe from nonnormality. If one has this in mind and uses the Shapiro-Wilk test critically, it can be a helpful tool.

> For small samples, a nonsignificant Shapiro-Wilk test does not imply that no problems with normality are present.

4.1.3 Accounting for non-normality

We have at least three options now.

- The *transformation approach*:



Try to find a transformation g such that the sample $g(Y)$ produces better behaved residuals. A simple transformation yields the normal QQ plot to the left, see the exercises. The transformation approach is very attractive if a function g such that $g(Y)$ has a nice interpretation can be found. Often tried transformations are power functions, the logarithm, and some more. *Box-Cox* transformations provide a systematic approach.

The idea is to apply ANOVA to the transformed data $g(Y)$ and transform back for interpretation. Any tried transformations should be stated in the final report. Incidentally, Beall 1942 used the insect sprays data set to illustrate a family of transformations.

- Abandon normal distribution based methods, use nonparametric, robust, bootstrap or exact methods instead (see Section 4.4 and following).
- Do not care about the non-normality and proceed with the analysis. In most cases, this is not a good idea, even though ANOVA is robust to some degree to some violations of the model assumptions. In many cases, nothing is lost by choosing a different statistical technique.

4.2 The homoskedasticity assumption

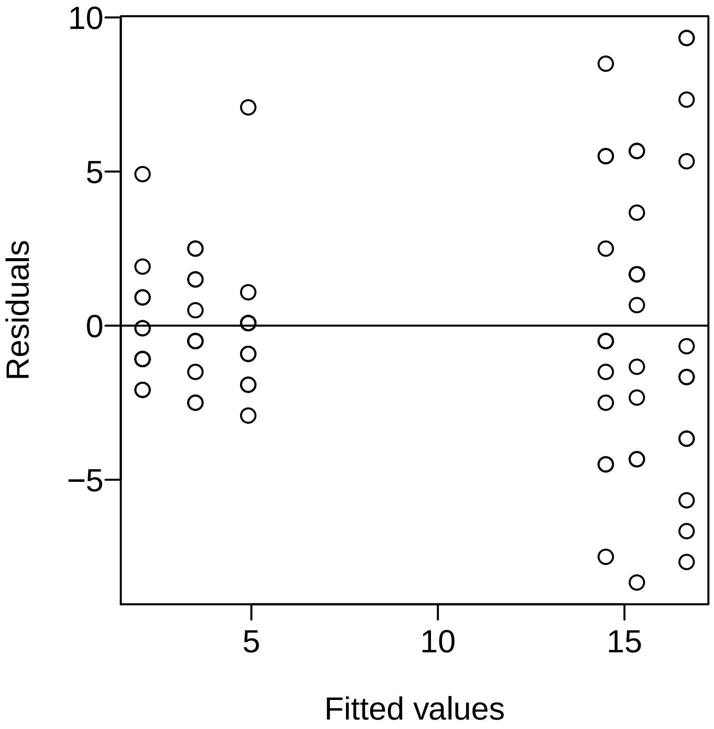
varibility in each treatment is the same

4.2.1 Residual plots

ANOVA assumptions include that the true variance of the dependent variable is the same in each treatment (i. e. that the data are *homoskedastic*). A visual assessment is possible with residual plots. Here, plotting the residuals vs. the fitted values is interesting.[14]

```
> plot(fitted(ins.lm), resid(ins.lm), las = 1,
+      xlab = "Fitted values", ylab = "Residuals")
> abline(h = 0)
```

[14] Sometimes, also (Pearson) standardized residuals are plotted. To get them, supply `type = "pearson"` to `resid`.



spread arund 0

The variance of the residuals seems to increase with the observed value, violating the equal variance assumption. Furthermore, one point with a fitted value of about 5 might be an outlier.

4.2.2 Testing homoskedasticity (equal variances)

As discussed in Venables 1998, if treatments show large differences in variance, this is often more important in practice than differences in means, because a high variance implies a low reliability, which is generally not desirable (e. g. in production processes). Thus it is important to test the equality of variances in the different treatments (and model possible heteroskedasticity).

> To formally test homoskedasticity, we use both the *Bartlett* and the *Levene* test.

```
> bartlett.test(count ~ spray, data = InsectSprays)
> leveneTest(count ~ spray, data = InsectSprays)
```

gives $p = 9.085 \cdot 10^{-5}$ for Bartlett's and $p = 0.004$ for Levene's test, both clearly rejecting the null hypothesis of equal variances.[15]

[15] Bartlett's test is very sensitive to non-normality. (This is not so problematic as we want to test normality anyway.) The residuals do have problems with normality, which explains why Bartlett's test produces a much smaller p value than the Levene test for this data set.

> If any of the two tests rejects variance homogeneity, this indicates a problem that needs to be accounted for.

4.2.3 Accounting for heteroskedasticity

We have three main alternatives.

1. Find a transformation g such that the variances of $g(Y)$ are more homogeneous. Sometimes a good g fixes both the normality and the variance problems.

2. Use robust methods, see Section 4.5.

3. Perform a one-way ANOVA with adjusted degrees of freedom. We show this next.

By calling

```
> oneway.test(count ~ spray, data = InsectSprays)

#
#  One-way analysis of means (not assuming equal variances)
#
# data:  count and spray
# F = 36, num df = 5, denom df = 30, p-value = 8e-12
```

we perform a version of ANOVA *which requires normality but does not need equal variances.* It is a generalization of *Welch's* two-sample t-test to more than two samples. We should not use this approach here because we have problems with normality.

> If your data seem to come from a normal distribution but suffer from unequal variances, using `oneway.test` is sensible.

4.3 Consequences of model assumptions not holding

In classical ANOVA, the variance is assumed to be the same for each treatment. This is often a questionable assumption for real data. Also the assumption of normality is often highly questionable for real data.
One reason why ANOVA is still used as a model even though the data may not come from a normal distribution is as follows. By the Central Limit Theorem, we know that under mild conditions, treatment *means* have an asymptotically normal distribution when they are properly scaled. In many cases, this approximation works already surprisingly well for moderate treatment sizes (in the region of dozens) and clearly nonnormal data.

However, one should be careful not to rely on this too much, especially for small samples, skewed data or in the presence of outliers or heavy tails.
Much statistical research has been done on the consequences of violations of the model assumptions, and it is difficult to give a short summary. An excellent and very accessible discussion is found in Wilcox 2010.

> Violations of the model assumptions can very severely inflate the probability of a Type I error (i. e. that the F test rejects the true null hypothesis of no treatment effects too often on average) and diminish the power (i. e. that the F test too often fails to reject a false null hypothesis).

In essence, everything can go wrong, especially in small samples. On top of that, with a small sample, violations of the normality assumption are easily missed by normality tests because they lack power.

In large samples, the situation is generally better.

> *Significance of a model assumption violation is no measure of its severity.*

For large samples, normality tests often yield a significant result because the power of the tests increases with sample size, so even small deviations from normality lead to a rejection of the normality hypothesis. On the other hand, the sample *means* are often close to normally distributed, *even though the data are not*, by the Central Limit Theorem.[16]
It is hard to make theoretical quantitative statements about this. One remedy is to conduct simulation studies to assess the severity of the impact of some model assumption not holding *in a specific way*.
One consequence of the model assumptions not holding is that the F statistic no longer has an F distribution under the null hypothesis. By comparing it with quantiles of the F distribution, invalid p values are obtained. But if the real distribution of the F statistic is reasonably close to the F distribution, or if the F statistic is very high, this does not matter from a practical point of view.

> In many cases, the best remedy is to choose a test or method that does not require normality, for example bootstrap methods or robust methods.

Some researchers refuse to do this because in case the data *do* come from a normal distribution, some power is lost by using these more stable procedures. Sometimes, a good strategy would be to have a sample which is big enough to reach the required power with a statistical procedure that does not rely on normality.

[16] This leads to the unsatisfactory situation that normality tests start becoming powerful when we do not need them any more (normality of the residuals is no longer that important for large samples by CLT arguments). But beware: the CLT argument (or some other argument, often simply stated as "ANOVA is robust to model violations") is often overused and wrong in such generality.

4.4 Nonparametric methods safe but not as powerful as parametric methods

4.4.1 Assumptions

The main use of nonparametric analyses of the one-way layout is that no normality assumptions are necessary.

With the notation introduced in Section 2.1.5, let us now only assume that:

1. The n random variables $\{Y_{ji}\}$ are mutually independent,

2. all observations $Y_{j1}, Y_{j2}, \cdots, Y_{jn_j}$ from treatment j come from the same continuous distribution F_j, for all $j \leq k$, and that

3. the treatment distribution functions F_j are *shifted* variants of some distribution F, i. e. that for all $t \in \mathbb{R}$
$$F_j(t) = F(t - \tau_j)$$
where τ_j is the shift (*effect*) for treatment j.

Because the treatments may still only differ by shifts, the variances for all the treatments should still be the same. The parametric model is retrieved if one assumes that F is a normal distribution function.

4.4.2 The Kruskal-Wallis test

The Kruskal-Wallis test is used to test the null hypothesis 0假设：

$$H_0 : \tau_1 = \ldots = \tau_k$$

that each of the underlying distributions is the same, against the alternative that at least two treatment effects differ. The Kruskal-Wallis test is in general not a test for equality of medians (because the τ_j need not be medians). It is a rank based method and generalizes the (two-sample) Wilcoxon rank sum test to the setting of k samples. The test idea is to compare the mean rank in each treatment. If the null hypothesis is true, mean ranks should not differ too much between treatments. As all rank-based methods, the Kruskal-Wallis test has problems with ties; an exact version `kruskal_test` is found in `coin`. The standard test is implemented as `kruskal.test` in `R`:

```
> kruskal.test(count ~ spray, data = InsectSprays)

#
#  Kruskal-Wallis rank sum test
#
# data:  count by spray
# Kruskal-Wallis chi-squared = 54.691, df = 5, p-value = 1.511e-10
```

The null hypothesis is clearly rejected for the insect sprays data.
To test which treatments differ from each other, Wilcoxon rank sum tests for pairwise treatment comparisons could now be applied, as explained in Section 2.3.3. It is preferrable to use special post hoc procedures to perform all two-sided pairwise comparisons, namely the method by Dwass, Steel, and Critchlow-Fligner.[17] This method is implemented in the `pSDCFlig()` function in the `NSM3` library.

```
> library(NSM3)
> with(InsectSprays, pSDCFlig(x = count, g = as.numeric(spray), method=NA))
```

We used `as.numeric()` to turn the names of the insecticides into numbers, as `pSDCFlig` requires this. Depending on your computer, these computations could take quite some time.[18] In return, you get p values for all the pairwise treatment comparisons.
There also exist nonparametric tests for special alternatives, such as those belonging to ordered factors. In this case, one could use the *Jonckheere-Terpstra* test or variants of it, see Hollander, Wolfe, and Chicken 2014.

4.5 Robust methods

In case that the treatment variances are not equal, the power of the Kruskal-Wallis test can suffer. The Brunner-Dette-Munk test is less affected by this and is implemented in the `asbio` package as `BDM.test` function. It tests the null hypothesis that the distributions of the numeric variable are the same in each treatment.

```
> library(asbio)
> with(InsectSprays, BDM.test(count, spray))

#
# One way Brunner-Dette-Munk test
#
#        df1       df2        F*    P(F > F*)
#   4.828351 63.19189 44.26642 4.138278e-19
```

Here, this null hypothesis is rejected. This only indicates that some difference between the distributions was found and closer inspection is now necessary.

Another method from robust statistics is implemented as `trim.test` in `asbio`. It is especially useful for data with outliers and tests the equality of trimmed means instead of means. Here is how to use it (with 20% trimming, on each side. This seemingly excessive amount of trimming works quite well for many scenarios):

[17] The situation is similar to Tukeys HSD improving upon pairwise t tests.

[18] Roughly a half hour on my 2009 MacBook Pro, some other software running.

```
> with(InsectSprays, trim.test(count, spray, tr = 0.2))

# $Results
#   df1      df2        F*         P(>F)
# 1   5 18.92297 28.22811 3.745716e-08
```

The null hypothesis is rejected, and we conclude that the trimmed means differ between the insecticides. See Wilcox 2012 for further information.

Although nonparametric and robust methods get little room in this script, they are important for applications, because it is desirable to use statistical methods that do not rely on questionable assumptions.

A number of articles uses simulation (Monte Carlo) methods to study the consequences of non-normality and heteroskedasticity in the one-way setting. See e. g. Cribbie et al. 2007 for an overview. A good overview of robust methods is given in Wilcox 2012, see also the R vignette for robust methods.

4.6 Permutation methods

Consider the following one-way layout with three treatments having two observations:

$$Y_1, Y_2 \,|\, Y_3, Y_4 \,|\, Y_5, Y_6$$

Bars separate the treatments. To test the null hypothesis that the three treatment means are the same, calculate the F statistic. (Permutation methods are also applicable to any other statistics, we just chose the F statistic to have something tangible.)

Now consider the following rearrangement (called a *permutation*) of the data:

$$Y_3, Y_5 \,|\, Y_2, Y_6 \,|\, Y_1, Y_4$$

If the null hypothesis is true, then the theoretical means of the three treatments of this permuted data set are still all the same (under the null hypothesis, all Y_i have the same distribution). This leads to the following procedure, cf. Basso et al. 2009, Ch. 5.2:

1. Compute F for the original data.

2. Generate either all or a large number of randomly chosen permutations $\pi_b Y$ of the data. Call the total number of permutations B.

3. For each permutation $b = 1, \ldots, B$, calculate the F statistic for the permuted data $\pi_b Y$ and call it F_b.

4. Compute the p value
$$p = \frac{\#\{b : F_b \geq F\}}{B}$$

as the proportion of permutation test statistics F_b which are at least as extreme as the F statistic of the observed data.[19]

The reasoning is that if the null hypothesis is false, then the observed data should generate a large value of F, and only a small fraction of the F_b statistics obtained from the permuted data should be at least as extreme by chance.

This approach to hypothesis testing is called a *permutation test*. By simulation studies, it can be shown that permutation tests perform quite well for a large number of settings. They can be very flexibly adapted to a much wider set of test problems than ANOVA; one main disadvantage is that permutation tests only provide p values, but do not directly measure effect sizes.

In `R`, for example the libraries `lmPerm` and `coin` can perform the permutation test for one-way ANOVA; see the help files and vignettes for these two packages for further information.

Before running the analysis, let us think about the number of permutations briefly. There are $72! \approx 6 \cdot 10^{103}$ permutations of the insect sprays data. This astronomical number defies any modern computer. The packages implement different stopping rules to decide what constitutes a sufficiently big number B of randomly chosen permutations. Here, we show the use of `lmp` from `lmPerm`; `lmp` can be used like `lm`:

```
> library(lmPerm)
> anova(lmp(count ~ spray, data = InsectSprays))

# [1] "Settings:  unique SS "
# Analysis of Variance Table
#
# Response: count
#           Df R Sum Sq R Mean Sq Iter  Pr(Prob)
# spray      5   2668.8    533.77 5000 < 2.2e-16 ***
# Residuals 66   1015.2     15.38
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Here, 5000 iterations were deemed sufficient, and the resulting Monte Carlo p value is highly significant. Note that the p value may now be different each time you run the analysis, so it is important to have a sufficient number of iterations such that these fluctuations are negligible for practical purposes. For scientific work, it is compulsory to set and note the random number seed so that it is always possible to reproduce the exact same results. See the help for `set.seed`.

[19] # denotes the number of elements of a set.

4.7 Exercises

1. Do you agree with my judgment of the insect sprays normal QQ plot? Simulate QQ plots and discuss with your neighbors.

2. Find a good transformation g for the insect sprays data so that normality and equal variances tests and the QQ plot show no anomalies.

3. Should we adjust the p values when we do two tests for homogeneous variances in Section 4.2.2 (multiple testing)?

4. Perform a one-way ANOVA of the data set `PlantGrowth` supplied in `R`. The data set contains results from an experiment to compare yields (measured by dried weight of plants) obtained under a control and two different treatment conditions. Check assumptions, make nice plots, etc. Also try nonparametric, robust and permutation methods.

all the p value are trusted only when the assumption is fulfilled

5 Blocking

This section is mostly based on Bailey 2008, Ch. 4 and Dean, Voss, and Draguljić 2017, Ch. 10.

5.1 Types of blocks

To account for a source of variability or correlation that is expected to matter for the experiment, similar plots are combined in *blocks*. But why are the plots similar?

Natural discrete blocks

The plots may naturally be grouped in blocks.

In Example 1.1, piglets from the same mother live in the same pen. Offspring from the same parents tends to be more similar than offspring from different parents, so each mother/pen naturally defines a block.

Example 5.1. *To compare three different treatments of knee cap injuries in goats, 12 goats are used. Three legs per goat are chosen at random, the knee caps are surgically damaged (a small clean cut, under anesthetic) and each knee cap is thereafter carefully treated with one of the treatments. After 52 weeks, the animals are sacrificed and the healing process is measured. The legs are the experimental units, and the goats are the blocks.*
Animal research in Switzerland is very strictly reglemented, and each experiment needs approval by a cantonal committee, which has to judge if the severity of the interventions is justified by the expected knowledge gain. More information is found at the BLV website.

Sometimes, for example in agronomy, the same plot may be used for several experiments over the years. Then it could be a good idea to use the treatments of the previous year as blocking factor in case there are any *carry-over* effects.

Continuous gradients

The plots may show differences which seem to change along a continuous gradient. This is often due to plot inhomogeneity in space or time.

Example 5.2. *Agricultural field plots may cover quite large areas depending on the size of the plots and the experiment. As a result, it is often to be expected that soil properties, shade, etc. and therefore the fertility vary considerably over the whole set of plots. Plots are thus often blocked such that plots close to each other are defined as one block.*
Sometimes, a particular variable such as the average exposition (calculated e. g. with a GIS) of the plot is also used as a covariate *in the statistical analysis.*

If an experiment has to be split over several days, maybe the days can explain a part of the variance. Then, the day of the run might be included as a blocking factor.
If enough data are available, it is also feasible to fit models which explicitly model the effect of time and/or space (time series models, longitudinal data models, mixed models with temporal/spatial components, generalized additive models, spatial statistics, Bayesian hierarchical models, . . .). However, experimental data sets are often on the small side for these methods to work well, so blocking is often preferred.

Trial management blocks

person as block

Trial management may introduce plot inhomogeneity.
Many experiments need more staff than one person. However careful staff is instructed, there may still be some individual freedom and thus extra variance. (For this reason, the same person should grade all the student answers to the same question.) The same idea applies to lab technicians, nurses, etc. One should be careful how to assign the staff to the treatments. *If possible, staff assignment should respect the block structure.*

Example 5.3. *An experiment is used to measure the amount of litter on ten field plots which are either close to the forest or further away from it (this is the treatment, the forest edge is in the north). The plots are blocked in five blocks depending on the distance to a hedge in the west. In each block, one plot is close to the forest and one further away. Because data will be compared within blocks, it is crucial to let the same staff (untrained students) perform all the measurements in one block. Observer biases will affect the whole block and cancel out. If half of the staff measures only the forest plots and the other half only the open plots, observer biases will accumulate.*

In agronomy, many treatments are applied with a tractor. This gives rise to split plot designs, cf. Example 1.10. The same principle (using a factor whose values are expensive to change as main plot factor) is also applied in industrial experiments.

Principles for blocking

Statistical and practical considerations suggest that if possible, blocks should

1. all have the same size,
2. be sufficiently big to apply each treatment at least once per block.

It is not always feasible to satisfy these conditions. For example, if you want to compare five treatments in Example 5.1, you will run out of legs.

5.2 Complete block designs

We start with the simplest block designs.

what is this mean?

> In the *randomized complete block design*, every treatment is applied in every block exactly one time, allocating treatments randomly within blocks.

In RCBDs, block sizes have to be equal to the number of treatments. Example 5.1 is an RCBD if we interpret the untreated leg as control treatment.

> In the *generalized randomized complete block design*, every treatment is applied in every block exactly $r > 1$ times, allocating treatments randomly within blocks.

The insect sprays experiment is a GRCBD with $r = 2$.

Remark 5.1. Often, the word "randomized" is omitted when naming the designs since the randomization is taken for granted anyway.

If block sizes are not multiples of the number of treatments, we talk about *incomplete block designs*.

5.3 Fixed block effect models

The fixed block effect model for the RCBD claims that

$$Y_{ji} = \mu + \beta_j + \vartheta_i + \varepsilon_{ji}$$

where Y_{ji} is the measurement of treatment j in block i, where $j = 1, \ldots, k$ and $i = 1, \ldots, b$, μ is a constant, β_j is 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 treatment effects are the same in each block (we will later say that this model assumes that there is no `block` × `treatment` interaction). With only one observation per block and treatment combination, it is not possible to test this assumption.

The design might look similar to the factorial two-way ANOVA which we will discuss below, but there is an important distinction: here, the randomization happens only inside blocks.

The big drawback of the fixed effect model and the reason we do not pursue it further here (see Dean, Voss, and Draguljić 2017, Sect. 10 for a discussion) is that this model is usually not what we want.

1. In experiments which use complete block designs, the interest is usually not in the effect of the particular blocks used in the study (particular goats, particular pens). We do not care for ϑ_i, since in the future we usually can not use the same blocks again.

2. Estimation of the block effects uses too many degrees of freedom $(b - 1)$.

3. We do not care if the block effect is significant, we only want to correctly estimate the treatment effect while accounting for the plot structure induced by the blocks.

4. The assumption that error terms are independent may not be particularly realistic. It could be preferable to have correlated error terms within blocks. In technical terms, the fixed effect model claims that blocks only affect the mean, but not the correlations of the observations.

If you will use the exact same blocks again (e. g. if the block is a city), it can absolutely make sense to study fixed effect models.

5.4 Random block effect models usually block as random effect

The random block effect model for the RCBD claims that

$$Y_{ji} = \mu + \beta_j + b_i + \varepsilon_{ji}$$

where Y_{ji} is the measurement of treatment j in block i, where $j = 1, \ldots, k$ and $i = 1, \ldots, b$, μ is a constant, β_j is the *fixed* effect of treatment j, $b_i \sim \mathcal{N}(0, \sigma^2_{\text{block}})$ is the *random* effect of block i and ε_{ji} denotes the random error terms. It is assumed that $\varepsilon_{ji} \sim \mathcal{N}(0, \sigma^2)$ independently and that the b_i are independent of the ε_{ji}. The model has two random components: the error terms with variance σ^2 and the random block effects with variance σ^2_{block}.
In random block effect models, the levels of the blocking factor were chosen randomly from a (typically much bigger) set of possible levels. The main interest is not in the effect of blocks, only in the variation introduced by the blocking factor.
Because the model includes both fixed and random effects, it is called a *mixed effects model*. This special model is sometimes referred to as *random intercept model*. It also makes the assumption that there is no interaction between block and treatment effects.

6 Fitting and interpreting mixed effects models

Because mixed models can be very complex, there are several packages/approaches to fitting them. The two most popular packages are `nlme` and `lme4`. We focus on `nlme` here, but for our purposes, `lme4` is more or less similar. This chapter closely follows Pinheiro and Bates 2000.

eth zurich

We now go back to the insect sprays data set. It has two observations per block and treatment (so, it is a GRCBD), but this only requires an additional index in the notation and the interpretation of the results stays the same.

6.1 Data structure

We saw so far that the insect sprays data set should be square-root transformed. We now want to also keep track of the blocks. We manually add the blocks to the data set and show a few lines:

```
> InsectSprays$block <- factor(rep(rep(1:6, each=2), times = 6)) ## design
> head(InsectSprays)

#   count spray block spray.o block.o spray.sum
# 1    10     A     1       A       1         A
# 2     7     A     1       A       1         A
# 3    20     A     2       A       2         A
# 4    14     A     2       A       2         A
# 5    14     A     3       A       3         A
# 6    12     A     3       A       3         A
```

Note that the data set must have one variable for the blocks, which is sometimes called the *long* format. You may not have the measurements for each block in one separate column, which would be the *wide* format.

6.2 Fitting the model

To fit mixed effect models, we have to tell `R` what the fixed effects are, what the random effects are and if there is any structure (crossed/nested) among the random effects. At the moment, we only have one random effect, called a *random intercept per block*, and we tell `R` to fit the model as follows.

here means: one random intercept for each block

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

The square root transformation is example specific and not used in general. The `random` argument tells `R` that we want one random intercept for each block.

ML and REML estimation

The mathematical details of fitting mixed-effect models are beyond the scope of these notes, although for simple designs, it is possible to simplify the results. For us, the following is important:

- There are two approaches to estimation: maximum likelihood (ML) and restricted maximum likelihood (REML).
- The ML estimates are useful for comparing models with different fixed effects (for example: with or without interaction effects, or using different contrasts). REML estimates should not be used for this.
- ML tends to underestimate the variances. The REML method essentially differs from the ML method by giving different variance estimates. For large samples, the differences become smaller.
- Therefore, often ML estimation is used for model selection and the selected model is then re-fitted with the REML method to obtain better variance estimates. The REML method is the default of `lme`.

Some more details: the likelihood function is the probability density function of the data, given the parameters (all the fixed effect parameters, all the random effects and the variances), but interpreted as a function of the parameters, with the data fixed.
The ML method is based on maximizing the log-likelihood function as a function of the fixed and random effects. The REML method does the same for the restricted log-likelihood. we choose the model given the observed data is as likely as possible under the model
The maximization algorithm is a hybrid approach that starts with some iterations of the expectation-maximization (EM) algorithm to get near the global optimum and then switches to a Newton-Raphson approach. The reason for this hybrid approach is that the EM algorithm is very quickly computed and tends to bring parameter estimates close to the global optimum of the likelihood function quickly, but then tends to be slow to converge once it is near the optimum. The Newton-Raphson algorithm is more expensive to compute and can be very unstable far away from the optimum. Near the optimum, it is quick to converge.
It is possible to monitor and to control the algorithms during optimization, cf. Pinheiro and Bates 2000, Ch. 2.2, but we do not go into this because usually convergence of the algorithms is not a problem.

6.3 Extracting model results

6.3.1 The model summary

To extract the REML parameter estimates, we simply invoke

```
> summary(ins.lme)

# Linear mixed-effects model fit by REML
#  Data: InsectSprays
#        AIC      BIC    logLik
#   152.2289 169.7461 -68.11445
#
# Random effects:
#  Formula: ~1 | block
#         (Intercept) Residual
# StdDev:   0.2540835 0.579766
#
# Fixed effects: sqrt(count) ~ spray
#                 Value Std.Error DF    t-value p-value
# (Intercept)  3.760678 0.1969021 61  19.099225  0.0000
# sprayB       0.115953 0.2366885 61   0.489897  0.6260
# sprayC      -2.515822 0.2366885 61 -10.629254  0.0000
# sprayD      -1.596325 0.2366885 61  -6.744412  0.0000
# sprayE      -1.951217 0.2366885 61  -8.243821  0.0000
# sprayF       0.257939 0.2366885 61   1.089782  0.2801
#
# Standardized Within-Group Residuals:
#        Min         Q1        Med         Q3        Max
# -2.4815817 -0.5213930 -0.1471345  0.6234485  1.9749960
#
# Number of Observations: 72
# Number of Groups: 6
```

linear mixed-effects model
random effects

AIC:
BIC
logLik

Yji = μ + βj + bi + eji

SprayA is reference

SprayB:the difference compared with sprayA

what does this mean?

and see that $\hat{\sigma} = 0.579$, $\hat{\sigma}_{\text{block}} = 0.254$. This shows that the block effect is not negligible compared to the errors. We do not formally test if $\sigma^2_{\text{block}} = 0$, due to two reasons:

- We want the block effect to model plot structure. We are not interested in testing whether it is zero or not, we just want to account for it.
- Testing $\sigma^2_{\text{block}} = 0$ introduces statistical difficulties because 0 is at the boundary of the parameter space (variances can not be less than zero).

AIC quatifies how much information we are losing from the model, the small the better

The log-restricted-likelihood, the Akaike information criterion AIC and the Bayesian information criterion BIC are also given. The two latter terms take into account the number of parameters used in the model as

$$\begin{aligned} \text{AIC} &= -2\log L + 2n_{\text{par}} \\ \text{BIC} &= -2\log L + n_{\text{par}}\log n \end{aligned}$$

the high the logL, the smaller AIC,BIC the better

compare different models by looking AIC and BIC

where L is the likelihood, n_{par} is the total number of parameters estimated and n is sample size. For the insect sprays model, we have six estimates for the fixed effects (six groups) and two variance estimates (for σ^2 and for σ^2_{block}), so $n_{\text{par}} = 8$. Smaller values of the information criteria are preferable. Both criteria are used for model comparisons, but we do not need them for this example.
Below that, we find the REML estimates of the fixed effects with the usual t tests and the correlations of the fixed effects estimators. It is also possible to extract the fixed effects with

```
> fixef(ins.lme)

# (Intercept)      sprayB      sprayC      sprayD      sprayE      sprayF
#   3.7606784   0.1159530  -2.5158217  -1.5963245  -1.9512174   0.2579388
```

6.3.2 The random effects

Technically, the random effects are not *estimated*, but *predicted*. The difference is that the random effects are random variables, so we do not estimate them, but predict a value for them. How many random effects are there for the insect sprays data? One for each block. We can extract them with

```
> ranef(ins.lme)

#   (Intercept)
# 1 -0.30503308
# 2  0.28296568
# 3 -0.01645621
# 4 -0.10988229
# 5  0.19387989
# 6 -0.04547399
```

Because the model can have several levels of random effects, we need to specify which one we exactly want. So, to extract the random intercepts as a vector, you would use

```
> ranef(ins.lme)$`(Intercept)`
```

The random effects show how a particular block compares to a “typical” block with random effect of zero. Negative values mean that this block has less insects than a typical block, positive values mean higher counts.
Let us compare the block sample means (on the square root scale) to the overall mean to illustrate this.

```
> blk.centred <- with(InsectSprays, tapply(sqrt(count), block, mean) -
+                                   mean(sqrt(count)))
> blk.ranef <- ranef(ins.lme)$`(Intercept)`
> cbind(blk.centred, blk.ranef, ratio = blk.centred/blk.ranef)

#   blk.centred   blk.ranef    ratio
# 1 -0.43738127 -0.30503308 1.433881
# 2  0.40573924  0.28296568 1.433881
# 3 -0.02359626 -0.01645621 1.433881
# 4 -0.15755818 -0.10988229 1.433881
# 5  0.27800078  0.19387989 1.433881
# 6 -0.06520431 -0.04547399 1.433881
```

All the random effects predictions are essentially the deviations of the block means form the overall mean, but shrunk towards zero by the same factor.

6.3.3 Fitted values and residuals

Should the fitted values include the predictions for the random effects?

It depends on what you want to do. Usually, it makes sense to include the random effects. This is called *within-group* fitted values, they are produced by `fitted` by default. To exclude some or all random effects, you can use the `level` argument of the fitted command; setting `level` to zero only uses the fixed effects. We show the first few values only here:

```
> head(fitted(ins.lme))

#        1        1        2        2        3        3
# 3.455645 3.455645 4.043644 4.043644 3.744222 3.744222
```

fixed and random effect
E(Yb)=mean+

```
> head(fitted(ins.lme, level = 0))
```

only fixed effect

```
#        1        1        2        2        3        3
# 3.760678 3.760678 3.760678 3.760678 3.760678 3.760678
```

The same ideas apply to the residuals of the model, which are the observed values minus the fitted values. Accordingly, `resid` uses the random effects by default, but it also has a `levels` argument to override this. You can also do the calculations without auxiliary functions to be sure to obtain exactly what you want.

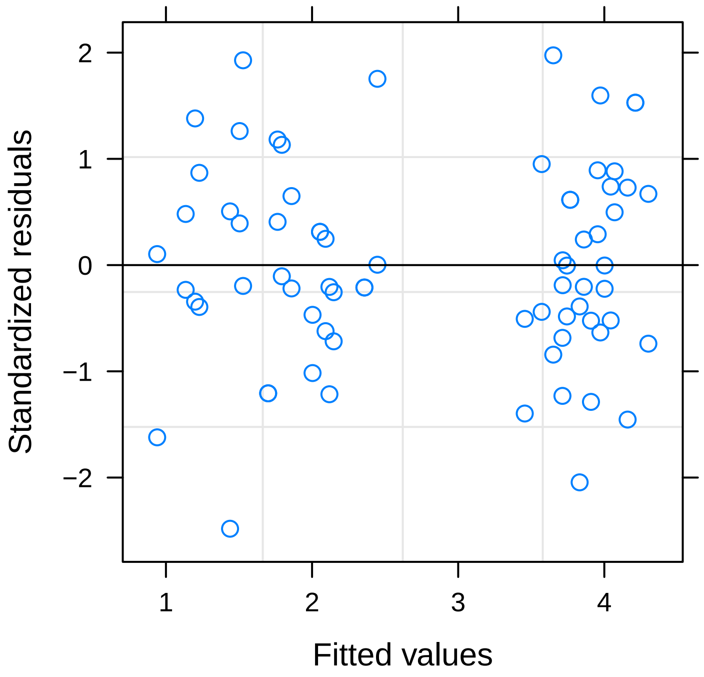
```
> head(resid(ins.lme))

#            1            1            2            2            3            3
# -0.293367658 -0.809894007  0.428491879 -0.301986689 -0.002564794 -0.280120566
```

6.4 Model diagnostics

Equal variances

To assess the model fit visually, it is useful to plot the standardized residuals versus the fitted values. Both terms incorporate the random effects, the term *standardized* residual means that $e_{ji} = y_{ji} - (\hat{\mu} + \hat{\beta}_j + \hat{b}_i)$ is divided by the estimated standard deviation of the error terms, $\hat{\sigma}$. This only changes the scale of the plot.

```
> plot(ins.lme)
```



The main thing to look for is whether the variance of the error terms seems to change with the fitted values (i. e. the residuals produce a wedge shape). Here, we see no problems with the equal variance assumption.

Formal statistical tests for equal variances are not widely used in mixed models. One rather relies on the residual plots. It is possible to explicitly model *how* the variance should be non-constant, this leads to so-called *heteroskedastic* models. The `lme` function can handle this. For example, it is possible to allow different variances according to the levels of a factor, cf. Chapter 7.10.

Normality

Assessing the normality of the residuals uses the same methods as for linear models: normal QQ plots and the Shapiro-Wilk test:

```
> ## qqPlot(resid(ins.lme))
> shapiro.test(resid(ins.lme))

#
#  Shapiro-Wilk normality test
#
# data:  resid(ins.lme)
# W = 0.98985, p-value = 0.8351
```

In principle, the same techniques apply to test the normality of the random effects:

```
> shapiro.test(ranef(ins.lme)$`(Intercept)`)
```

but typically, the number of blocks is much smaller than 30 (here, we have only six random effects), and it is useless to test random effect normality.

6.5 Confidence intervals and a hypothesis test

To quantify the precision of the estimators, confidence intervals are very useful. While we should be careful in using them if the data sets are very small and/or ill behaved (outliers, non-normality, . . .), they give a first impression about the precision.

```
> intervals(ins.lme)

# Approximate 95% confidence intervals
#
#  Fixed effects:
#                  lower       est.      upper
# (Intercept)  3.3669482  3.7606784  4.1544086
# sprayB      -0.3573348  0.1159530  0.5892408
# sprayC      -2.9891095 -2.5158217 -2.0425339
# sprayD      -2.0696124 -1.5963245 -1.1230367
# sprayE      -2.4245052 -1.9512174 -1.4779296
# sprayF      -0.2153491  0.2579388  0.7312266
# attr(,"label")
# [1] "Fixed effects:"
#
#  Random Effects:
#   Level: block
#                      lower      est.     upper
# sd((Intercept)) 0.1041355 0.2540835 0.6199464
#
#  Within-group standard error:
#     lower      est.     upper
# 0.4855018 0.5797660 0.6923322
```

Given the small number of blocks, it is not surprising that the estimate for the block standard deviation is extremely imprecise. how can you tell that this is imprecise?

It is possible to test the null hypothesis that all the fixed effect estimates, except the intercept, are zero. This is essentially the overall F test from linear models, conditional on the variance/covariance parameters. There are some discussions in the statistical literature about the degrees of freedom to use, but we follow what `lme` does. The F test can be obtained with

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

#              numDF denDF  F-value p-value
# (Intercept)      1    61 364.7804  <.0001
# spray            5    61  52.6215  <.0001
```

and yields a significant effect of the spray. The `type = "marginal"` argument specifies that *marginal* F tests are required, this is the recommended procedure, as in ANOVA models without blocks.

6.6 Treatment comparisons

Because the overall F test was significant, we may now want to compare the different insect sprays. Similar ideas as for linear models may be applied, cf. Chapter 3.

It is possible to use and set contrasts and try to use the `summary` for the required comparisons. Also `glht` may be used:

```
> library(multcomp)
> summary(glht(ins.lme, mcp(spray = "GrandMean")))
```

The multiple comparisons take the block random effects into account. Also Tukey's HSD may be produced this way.

6.6.1 Using lsmeans/emmeans

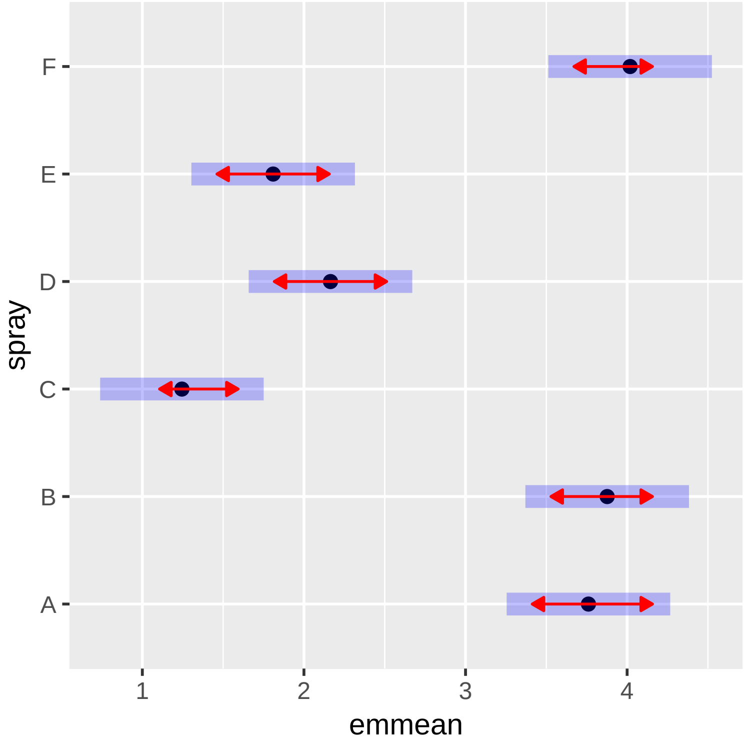
A very popular way of comparing treatments (especially in more complex settings) is to use what used to be called *least squares means* for decades and what is currently being renamed into *estimated marginal means*.

The idea is to define a *reference grid* of interesting values (for factors, these are just their levels) and then to compute the fitted values according to the model on the reference grid. The real usefulness of this method will become clearer once we have unbalanced data sets and more complex models. For now, we simply show how to obtain confidence intervals for each mean and how to perform pairwise comparisons (Tukey).

```
> library(emmeans)
> ref_grid(ins.lme)

# 'emmGrid' object with variables:
#     spray = A, B, C, D, E, F
# Transformation: "sqrt"

> (ins.emm <- emmeans(ins.lme, "spray"))

#  spray   emmean        SE df  lower.CL upper.CL
#  A     3.760678 0.1969021  5 3.2545253 4.266831
#  B     3.876631 0.1969021  5 3.3704783 4.382784
#  C     1.244857 0.1969021  5 0.7387036 1.751010
#  D     2.164354 0.1969021  5 1.6582008 2.670507
#  E     1.809461 0.1969021  5 1.3033079 2.315614
#  F     4.018617 0.1969021  5 3.5124641 4.524770
#
# Degrees-of-freedom method: containment
# Results are given on the sqrt (not the response) scale.
# Confidence level used: 0.95

> pairs(ins.emm)

#  contrast   estimate        SE df t.ratio p.value
#  A - B    -0.1159530 0.2366885 61  -0.490  0.9964
#  A - C     2.5158217 0.2366885 61  10.629  <.0001
#  A - D     1.5963245 0.2366885 61   6.744  <.0001
#  A - E     1.9512174 0.2366885 61   8.244  <.0001
#  A - F    -0.2579388 0.2366885 61  -1.090  0.8836
#  B - C     2.6317747 0.2366885 61  11.119  <.0001
#  B - D     1.7122775 0.2366885 61   7.234  <.0001
#  B - E     2.0671704 0.2366885 61   8.734  <.0001
#  B - F    -0.1419858 0.2366885 61  -0.600  0.9907
#  C - D    -0.9194972 0.2366885 61  -3.885  0.0033
#  C - E    -0.5646043 0.2366885 61  -2.385  0.1776
#  C - F    -2.7737605 0.2366885 61 -11.719  <.0001
#  D - E     0.3548928 0.2366885 61   1.499  0.6659
#  D - F    -1.8542633 0.2366885 61  -7.834  <.0001
#  E - F    -2.2091561 0.2366885 61  -9.334  <.0001
#
# P value adjustment: tukey method for comparing a family of 6 estimates
```

The package is under active development (see vignette) and also contains plot methods. A very nice plot may be obtained with

```
> plot(ins.emm, comparisons = TRUE)
```



According to the documentation, the blue bars depict confidence intervals for the marginal means. The red arrows are for the pairwise comparisons among the spray means. If an arrow from one spray overlaps an arrow from another spray, their mean difference is not significant, based on the `adjust` setting (which defaults to `"tukey"`).

6.7 Ignoring blocks

Let us examine what happens if we omit the blocking factor in the analysis:

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

# [1] 0.6283453

> sigma(ins.lme)

# [1] 0.579766
```

The fixed effects estimates (omitted here) are the same for this simple design, but the estimate $\hat{\sigma}$ of the error standard deviation (retrieved by `sigma`) is bigger in the model without block effects. This is because the variability that is due to the blocks is not properly accounted for.

The difference is not overwhelming here. For other data sets, it can happen that the wrong model without block effects has such a high $\hat{\sigma}$ that significant effects are missed.

6.8 The intraclass correlation coefficient

Observations from different blocks are independent according to the mixed model from Section 5.4. But observations from the same block i are correlated because they share the block random effect b_i.

The model implies that all observations within the same block have the same correlation, which is equal to the *intraclass correlation coefficient*

$$\mathrm{ICC} = \frac{\sigma^2_{\text{block}}}{\sigma^2_{\text{block}} + \sigma^2} .$$

For the insect sprays data, we obtain a sample ICC of

```
> (V <- VarCorr(ins.lme))

# block = pdLogChol(1)
#             Variance   StdDev
# (Intercept) 0.06455845 0.2540835
# Residual    0.33612856 0.5797660
```

```
> V <- as.numeric(V)
> V[1] / (V[1] +V[2])

# [1] 0.1611194
```

Due to the ICC definition used here, negative intraclass correlations are not possible in our model. Allowing this would require to specify the model in a different way, cf. Pinheiro and Bates 2000, Ch. 5.3. Negative intraclass correlations can be expected if there is competition inside blocks, for example for food or light.

6.9 The coefficient of determination

Because of the random effects, the decomposition of the sum of squares no longer works as it does for linear models. As a result, generally no values for R^2 are given in the context of mixed models, since it is not clear how to exactly treat random effects when we define an R^2, and the resulting quantities do not share all of the properties that R^2 does (some can take negative values, for example).
However, some reviewers still insist on something like an R^2 to quantify the fit of the model. They can usually be satisfied by giving them one or both of the marginal and conditional R^2 values defined by Nakagawa and Schielzeth. For details, read the package documentation.

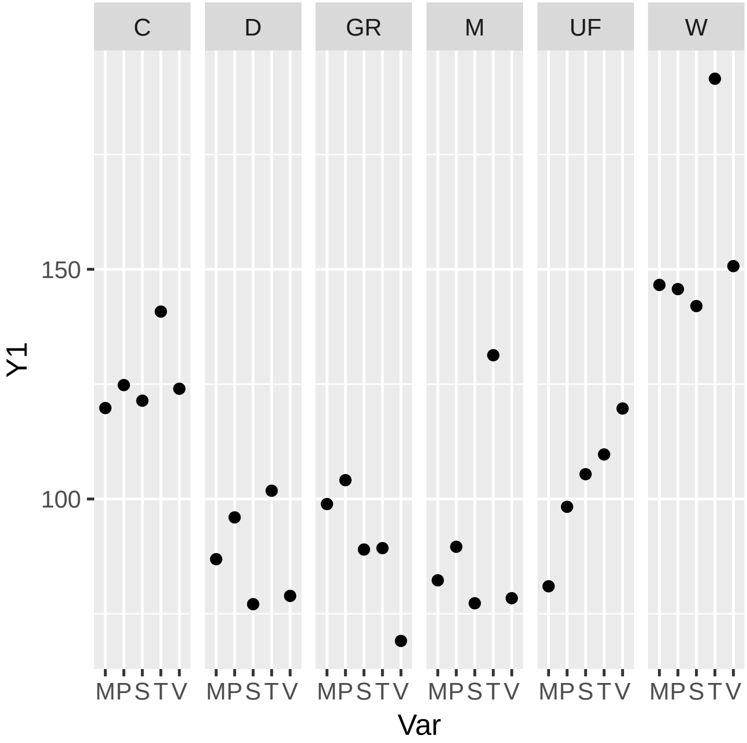
```
> library(MuMIn)
> r.squaredGLMM(ins.lme)
```

6.10 An example: the `immer` data and the Friedman test

Consider the `immer` data frame from `MASS`. It contains the yield of five varieties `Var` (M, P, S, T and V) of barley grown in six locations `Loc` (C, D, GR, M, UF and W) for two years. We focus on the yield `Y1` from the first year here. This is the whole data set:

```
       M     P     S     T     V
C  119.8 124.8 121.4 140.8 124.0
D   86.9  96.0  77.1 101.8  78.9
GR  98.9 104.1  89.0  89.3  69.1
M   82.3  89.6  77.3 131.3  78.4
UF  81.0  98.3 105.4 109.7 119.7
W  146.6 145.7 142.0 191.5 150.7
```

```
>     ggplot(immer, aes(x = Var, y = Y1)) +
+     geom_point() +
+     facet_grid(~Loc)
```



A terrible analysis

The importance of somehow accounting for the location can nicely be demonstrated in this example. Suppose we “forgot” about the location and ran this as a one-factorial ANOVA with factor variety and six repetitions per variety. Look at the data and try to guess what will happen.

> Because the values within locations are quite homogeneous, but the locations greatly differ from each other, ignoring the locations amounts to pooling heterogeneous data, which is always a Bad Thing. As punishment, residual variance is overestimated and we find no significant variety effect.

```
> anova(lm.immer <- lm(Y1 ~ Var, immer))

# Analysis of Variance Table
#
# Response: Y1
#           Df  Sum Sq Mean Sq F value Pr(>F)
# Var        4  2756.6  689.16   0.817 0.5264
# Residuals 25 21087.6  843.50

> sigma(lm.immer)

# [1] 29.04313
```

A slightly less terrible analysis

We could include location as a fixed effect. This does not respect the randomization, which only happened inside locations. But let us show the analysis.

```
> anova(lm(Y1 ~ Loc + Var, immer))

# Analysis of Variance Table
#
# Response: Y1
#           Df  Sum Sq Mean Sq F value    Pr(>F)
# Loc        5 17829.8  3566.0 21.8923 1.751e-07 ***
# Var        4  2756.6   689.2  4.2309   0.01214 *
# Residuals 20  3257.7   162.9
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Examining the residuals shows no clear pattern.) We see that both the location and the variety now have a significant effect.[20]
For the RCB design, usually the interest is *not* in the effect of the blocking factor. We know it will have an influence but that is not what we are interested in, so we would like to treat it as a blocking variable. Of course, if you are interested in exactly these locations, you should include them as a fixed factor.
So, a better analysis that properly treats location as a random blocking factor will use less degrees of freedom to estimate things we do not need, it will respect the randomization and actually answer our research question in a better way.

A parametric analysis

Let us include the location as a random effect.

[20] But interpret this with care as we estimated too many terms from the data that we have for the results to be very reliable.

```
> immer.lme <- lme(Y1 ~ Var, random = ~ 1 | Loc, data = immer)
> immer.lme

# Linear mixed-effects model fit by REML
#   Data: immer
#   Log-restricted-likelihood: -111.3314
#   Fixed: Y1 ~ Var
# (Intercept)        VarP        VarS        VarT        VarV
# 102.5833333   7.1666667  -0.5500000  24.8166667   0.8833333
#
# Random effects:
#  Formula: ~1 | Loc
#         (Intercept) Residual
# StdDev:    26.08863 12.76273
#
# Number of Observations: 30
# Number of Groups: 6

> anova(immer.lme, type = "marginal")

#             numDF denDF  F-value p-value
# (Intercept)     1    20 74.85450  <.0001
# Var             4    20  4.23088  0.0121
```

We find a significant variety effect. The standard deviation of the random intercepts is even bigger than the residual standard deviation, because the blocks are so different. The only drawback of this analysis is that the sample is very small, so assessing the model assumptions is very difficult.

```
> plot(immer.lme)
> qqPlot(resid(immer.lme))
> shapiro.test(resid(immer.lme))
```

The diagnostics look good, but it is not clear how much we should trust them in such a small sample. Perhaps the best thing to do is to use a nonparametric approach.

A nonparametric analysis

> Friedman's test provides a non-parametric test of the null hypothesis that the treatment effect is the same for every level of the treatment in an RCBD.

The test idea is to rank observations *within blocks* and then sum ranks for the treatments over all the blocks. The ranking makes the procedure robust to outliers (look at the T treatments in locations M and W). This robustness is paid for with some loss of power compared to the parametric model in case the data do come from a normal distribution. The Friedman test makes the assumption that the distributions of the dependent variables for the levels of the treatment are shifted variants of the same distribution, so it should not be used for wildly different distributions, nor when the treatments work differently some blocks (block × treatment interaction).

```
> friedman.test(Y1 ~ Var | Loc, immer)

#
#  Friedman rank sum test
#
# data:  Y1 and Var and Loc
# Friedman chi-squared = 10.933, df = 4, p-value = 0.02732
```

We find a significant variety effect. An extension of the Friedman test that handles missing values (making some assumptions) is known as Skillings-Mack test (see Hollander, Wolfe, and Chicken 2014) and implemented in several R packages.

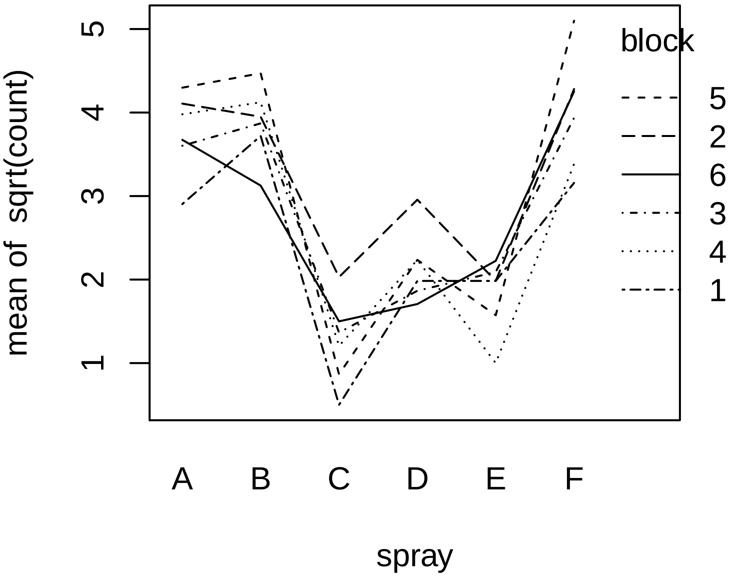
treatment effect is different between blocks

6.11 Interactions of block effects and treatment effects

judge the interaction in themodel

In case we have a GRCBD (more than one observation per treatment and block), we can answer the question whether the treatment effects are different by block.
For many research settings, it seems to make sense to assume that no such interaction between treatments and blocks exists based on background knowledge, but such knowledge is not always available.
The insect sprays data set is a GRCBD with two replications per treatment and block. We refer to the plot of the data in Section 2.1.4 which showed little evidence of such interactions.
Instead of plotting the raw data, sometimes one gets a better overview by plotting means for each combination of block and treatments. Such a plot is referred to as an *interaction plot* and meant to visualize the need for interaction effects.

```
> with(InsectSprays, interaction.plot(spray, block, sqrt(count)))
```



The first variable is on the x axis, here we chose the treatment. The second variable defines the trace variable, means are connected according to this variable. If there is no interaction, the traces should be more or less parallel. (Some amount of not being parallel is to be expected by chance even if there is no interaction.) It is not entirely clear to me whether we need interactions here based on this plot.

Fortunately, we have a better way of answering this question than by looking at plots. The key is to fit a model with *nested* random effects. The blocks are a random sample from the set of all possible blocks, so that any differences in treatment effects due to the blocks should also be treated as random effects. We can now formulate a model with two levels of random effects: one level for the blocks and one level for the treatments *within each block*.

This model may be written as

$$Y_{jir} = \mu + \beta_j + b_i + b_{ji} + \varepsilon_{jir}$$

where Y_{jir} is the r-th measurement of treatment j in block i, where $j = 1, \ldots, k$, $i = 1, \ldots, b$, $r = 1, 2$, μ is a constant, β_j is the fixed effect of treatment j, $b_i \sim \mathcal{N}(0, \sigma^2_{\text{block}})$ is the random effect of block i, $b_{ji} \sim \mathcal{N}(0, \sigma^2_{\text{trt(block)}})$ is the random effect of the treatment within the block and ε_{jir} denotes the random error terms.

The nesting of the random effects is formulated as follows in `lme`:

```
> ins.lme.x <- lme(sqrt(count) ~ spray, random = ~ 1 | block/spray,
+                  data = InsectSprays)
> ins.lme.x

# Linear mixed-effects model fit by REML
#   Data: InsectSprays
#   Log-restricted-likelihood: -67.43423
#   Fixed: sqrt(count) ~ spray
# (Intercept)      sprayB      sprayC      sprayD      sprayE      sprayF
#   3.7606784   0.1159530  -2.5158217  -1.5963245  -1.9512174   0.2579388
#
# Random effects:
#  Formula: ~1 | block
#         (Intercept)
# StdDev:   0.2394903
#
#  Formula: ~1 | spray %in% block
#         (Intercept)  Residual
# StdDev:   0.2706021 0.5254596
#
# Number of Observations: 72
# Number of Groups:
#            block spray %in% block
#                6               36

> ## ranef(ins.lme.x) ## run this code and look at the results!
```

The expression `block/spray` is to be read as “`block` and `spray` within `block`”. As you can check, at the block level, one random intercept per block is predicted, and at the second level, one random intercept per spray within each block is predicted. The log-likelihood of the model is now -67.43 (compared to -68.11 for the single level model), and we should assess the question whether this slightly better fit justifies the additional model complexity.

The standard approach to do this *if the models were fitted with the ML method* is to use *likelihood-ratio tests* which quantify whether the improvement of the likelihood is sufficient given the additional model complexity. In case the models have the same fixed effects (i. e. only the random effects differ), the method may also be used for REML fits. This works as follows for nested models: let L_2 denote the likelihood of the more complex model (with k_2 degrees of freedom) and L_1 the likelihood of the more simple model (with k_1 degrees of freedom). Likelihood ratio tests are based on the asymptotic chi-squared distribution of the likelihood ratio test statistic

$$2 \log \frac{L_2}{L_1} = 2(\log L_2 - \log L_1)\,.$$

This test statistic asymptotically has a χ^2 distribution with $k_2 - k_1$ degrees of freedom, and the p value of the test is then the probability of obtaining a bigger value than the observed value of the likelihood ratio test statistic from a χ^2 distribution with $k_2 - k_1$ degrees of freedom. It is known that the test can be conservative for random effects (boundary of the parameter space issues). More sophisticated methods are discussed in Pinheiro and Bates 2000, Ch. 2.4.1.
In `R`, the test is performed with

```
> anova(ins.lme, ins.lme.x)

#            Model df      AIC      BIC    logLik   Test  L.Ratio p-value
# ins.lme        1  8 152.2289 169.7462 -68.11445
# ins.lme.x      2  9 152.8685 172.5754 -67.43423 1 vs 2 1.360439  0.2435

> 1 - pchisq(1.360439, df = 9 - 8) # show p value calculation

# [1] 0.2434614
```

(You should give the simple model first.) The simpler model seems to be sufficient; in other words, there is no evidence for an interaction of treatment effects and blocks. We should thus proceed with the more simple model.

7 Designs with factorial treatment structure

7.1 Introduction

two genotype
two
factor structure
more times of all conbinations

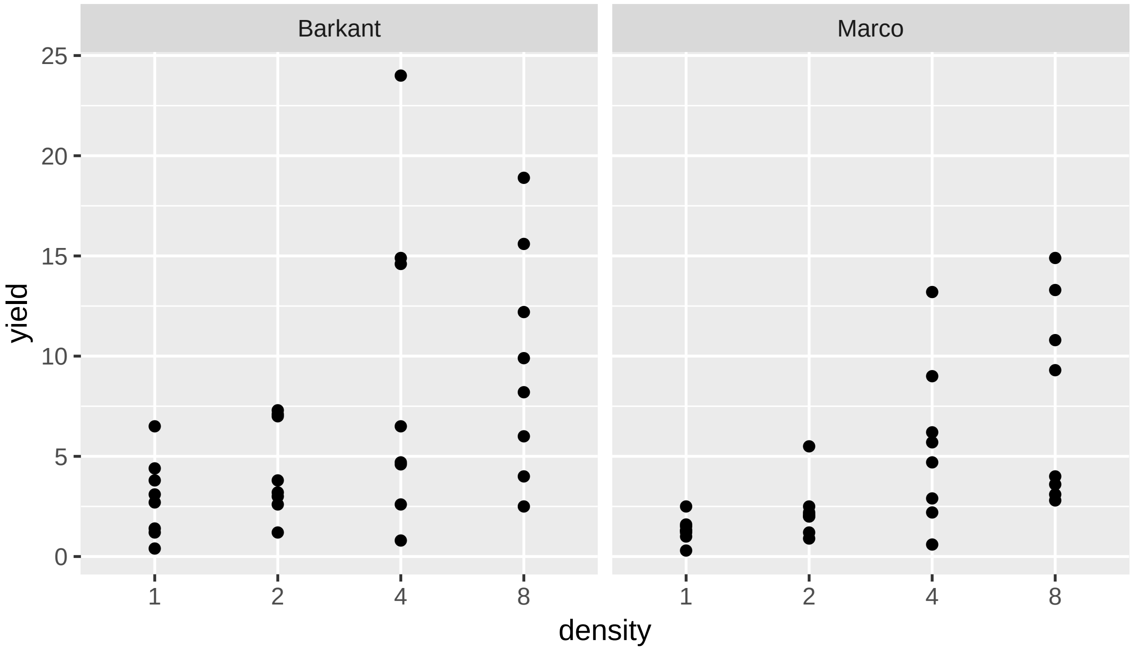
The turnip yield data

The question is how the planting density in kg/ha affects the mean yield of turnips of two genotypes. We turn the planting density into a factor to avoid problems later.

is there effect of density
is there effect of genotype
the treatment

```
> library(agridat)
> turnip <- mcconway.turnip
> turnip$density <- as.factor(turnip$density)
```

Consider the strip plot of the data:



The first and last four rows of the data frame look as follows:

```
> turnip[c(1:4,61:64),]

#         gen      date density block yield
# 1   Barkant 21Aug1990       1    B1   2.7
# 2   Barkant 21Aug1990       1    B2   1.4
# 3   Barkant 21Aug1990       1    B3   1.2
# 4   Barkant 21Aug1990       1    B4   3.8
# 61    Marco 28Aug1990       8    B1  14.9
# 62   Marco 28Aug1990       8    B2  13.3
# 63   Marco 28Aug1990       8    B3   9.3
# 64   Marco 28Aug1990       8    B4   3.6
```

And here are the means for each combination of density and genotype:

```
> with(turnip, tapply(yield, list(gen, density), mean))

#              1   2      4      8
# Barkant 2.9375 4.4 9.0875 9.6625
# Marco   1.3375 2.3 5.5625 7.7250
```

For the moment, ignore the fact that the data come from different blocks and the date variable for the sake of simplicity. We will show an analysis that incorporates the blocks as random effects below, but we want to focus on factorial treatment structures now. Three questions are interesting in this setting:

1. Is there a planting density effect? It seems so when inspecting the plot and the means: the yield seems to increase with the planting density.

2. Is there a genotype effect? The average Barkant values seem to be higher than the average Marco values.

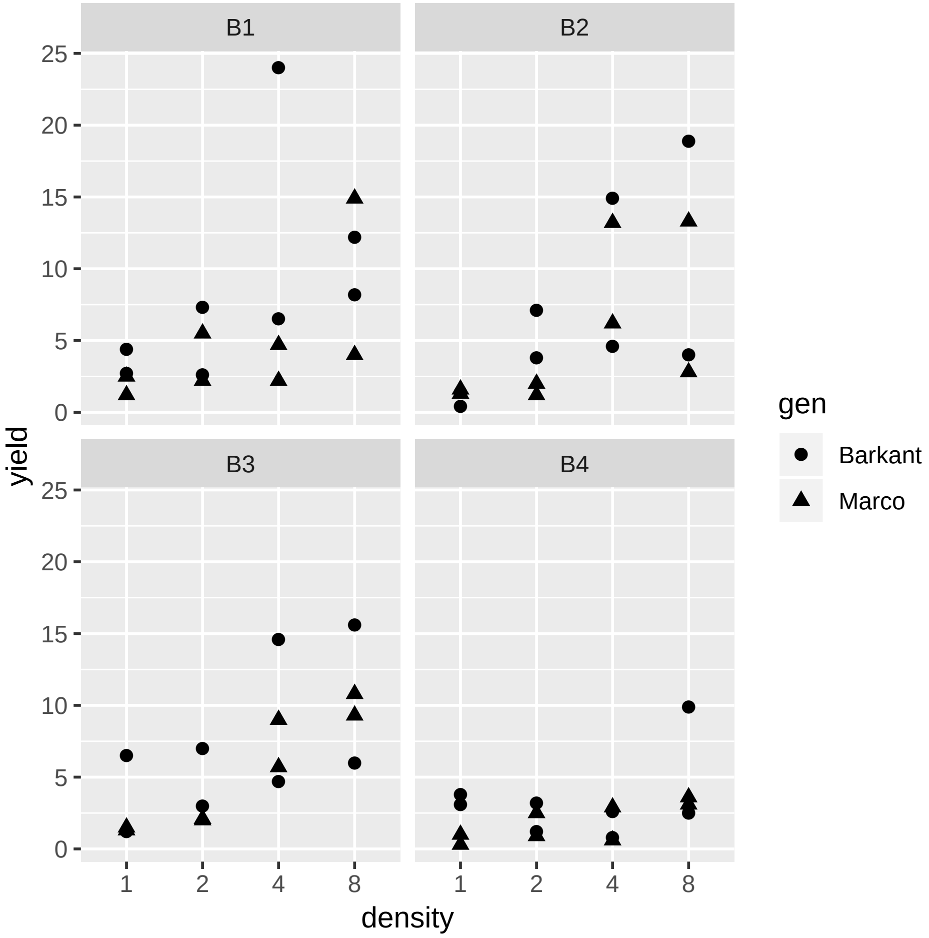
3. Does the planting density effect (if there is one) vary among the two genotypes? This does not appear to be the case for the turnip data.

We could do several one-way ANOVA to answer the first two questions (but not the third one). But this is not the best way of analyzing the data. We first visualize the data in a bit more detail and then introduce the analysis method.

7.2 Visualization

All the tools (strip plots, box plots, bar graphs and line plots) from the one-way setting apply here as well with minor modifications. One can use different colors, shadings or symbols to distinguish between the levels of one factor, as shown above for the box plot. The following code using the `qplot` function from `ggplot2` produces a first overview of the raw data:

```
> qplot(density, yield, data = turnip,
+       facets = ~block, shape = gen, size = I(2))
```



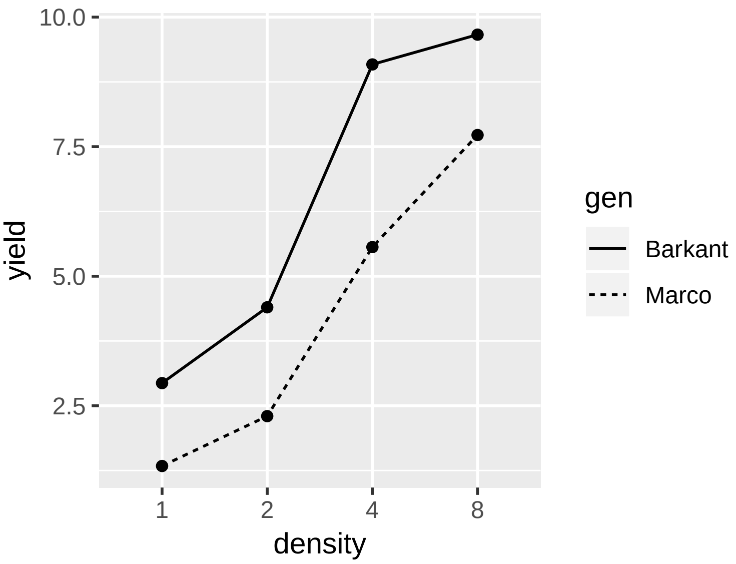
Block differences are clearly visible (Block 4 is suspicious) and an overall increase of the yield (and the variance) with the planting density. The Barkant variety seems to have produced a slightly higher average yield than Marco.

The *interaction plot* (seen above already) is often used in the two-way setting. One plots the group means for the two factors, connecting values of the same level of one factor with a line. Different shapes of the lines suggest a so-called *interaction effect*, i. e. that the effect of one factor on Y depends on the levels of the other factor.

Here is how to produce such a plot with `ggplot2`[21]. The `group = gen` statement ensures that points are connected by lines separately for each genotype.

```
> ggplot(turnip, aes(x = density, linetype = gen, group = gen, y = yield)) +
+     stat_summary(fun.y = mean, geom = "point") +
+     stat_summary(fun.y = mean, geom = "line")
```

[21] Plots for internal use need not be publication quality. For effective data analysis, it is often very useful to produce a few quick plots of the data without bothering about axis labels and legend placing. To publish the results, one should of course work much harder.



To me, the two lines look more or less parallel.

7.3 A model for factorial two-way ANOVA with replications

7.3.1 Notation

dentsity, genotype, block

Consider two factors X with levels $1, 2, \dots, a$ and Z with levels $1, 2, \dots, b$ whose influence on the numeric dependent variable Y is to be studied. In contrast to the setting with blocks, we are interested in the *fixed* effects of both factors here.
Let us adapt the notation a bit. Denote by Y_{ijk} the value of the k-th observation (where $k = 1, 2, \dots, n_{ij}$) in the group with factor X being equal to i and factor Z being equal to j, where $1 \le i \le a$ and $1 \le j \le b$.
To simplify the situation, we speak of *group* ij (instead of group (i, j)) to mean that the first factor takes the level i and the second takes the value j. Denote the n_{ij} measurements of variable Y from group ij by $Y_{ij1}, Y_{ij2}, \dots, Y_{ijn_{ij}}$ for each $i = 1, \dots, a$, $j = 1, \dots, b$. Let n denote total sample size.

The analysis is easier if the n_{ij} are all the same (i. e. if we have a balanced design). For the time being, we assume the design is balanced and discuss unbalanced designs in Section 7.9.

Because we ignore the blocks and the year for the sake of simplicity here, the turnip yield data has eight replications for every combination of density and genotype. It is thus a balanced factorial two-way layout with $r = 8$ replications per factor combination. In this chapter, we assume that $r > 1$, see Section 6.10 for $r = 1$.

7.3.2 Assumptions

As in the one-way situation, the parametric and nonparametric methods differ with respect to the assumptions on the distributions. Classical parametric two-way ANOVA assumes that the data come from a normal distribution with variance σ^2 which is the same for all groups ij.

7.3.3 The cell means model

The *cell means model* states that

$$Y_{ijk} = \mu_{ij} + \varepsilon_{ijk}$$

for all $1 \leq i \leq a$, $1 \leq j \leq b$ and $1 \leq k \leq r$ (where r denotes the number of replications which is the same for every group ij.) Each μ_{ij} is the theoretical mean of a group ij. The ε_{ijk} are independent normally distributed random variables with mean zero and variance $\sigma^2 > 0$.

> The cell means model corresponds to a one-way ANOVA using a factor with one level for each combination of the levels of the two factors. With the cell means model, we are forgetting the special structure of the groups defined by combinations of the levels of the two factors. As a result, the model is only suited to test whether there is any effect of the two factors *combined*.

We do not pursue the cell means model further here.

7.3.4 Effect models

> The *factor effect model* states that
>
> $$Y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_{ij} + \varepsilon_{ijk} \tag{2}$$
>
> where the ε_{ijk} are random variables, and all the other quantities are unknown, but fixed numbers. More precisely, α_i is called the *main effect* of level i of Factor X, β_j is called the main effect of level j of Factor Z and γ_{ij} is the *interaction effect* corresponding to the combination of these two levels.

To ensure that the coefficients can be uniquely estimated (with $a \cdot b = 8$ groups, we can only estimate 8 coefficients, but our model has $1 + a + b + a \cdot b = (a+1) \cdot (b+1) = 15$ coefficients), one has to choose some constraints. There are several ways to do this, we only explain the `R` default here.[22]

[22] Traditionally, a different set of contrasts is used in ANOVA, namely *sum contrasts*. They have some theoretical advantages. For consistency with regression, we use treatment contrasts.

Treatment contrasts:
Just as in one-way ANOVA, `R` by default chooses *treatment contrasts*, choosing a reference level for each factor (say, level 1). This means that the constraints are $\alpha_1 = 0$, $\beta_1 = 0$, $\gamma_{i1} = 0$ for $i = 1, \dots, a$ and $\gamma_{1j} = 0$ for $j = 1, \dots, b$. With this convention, we set $1 + 1 + a + b - 1 = a + b + 1 = 7$ coefficients to zero (-1 to account for the double counting of γ_{11}), so that we only have to estimate $a \cdot b$ coefficients which are now identifiable.
Let us show how to fit the model and read the output for this default method now.

> We use the formula `yield ~ gen * density` to fit a two-way ANOVA with interaction.

(`gen * density` is a shorthand for `gen + density + gen:density` which explicitly requires each factor and their interaction.)

> `R` automatically chooses a reference level for each factor (lexicographically if we do nothing about it), namely `Barkant` for the genotype and `1` for the density.

```
> turnip.full <- lm(yield ~ gen * density, data = turnip)
> coef(summary(turnip.full))

#                    Estimate Std. Error    t value    Pr(>|t|)
# (Intercept)          2.9375   1.522377  1.9295486 0.058734960
# genMarco            -1.6000   2.152966 -0.7431609 0.460490758
# density2             1.4625   2.152966  0.6792955 0.499748857
# density4             6.1500   2.152966  2.8565245 0.005999627
# density8             6.7250   2.152966  3.1235980 0.002827723
# genMarco:density2   -0.5000   3.044754 -0.1642169 0.870151687
# genMarco:density4   -1.9250   3.044754 -0.6322351 0.529806221
# genMarco:density8   -0.3375   3.044754 -0.1108464 0.912134467
```

The coefficient estimates are $\hat{\mu} = 2.94$, $\hat{\alpha}_2 = -1.60$, $\hat{\beta}_2 = 1.46$, $\hat{\beta}_3 = 6.15$, $\hat{\beta}_4 = 6.73$, $\hat{\gamma}_{22} = -0.50$, $\hat{\gamma}_{23} = -1.93$, $\hat{\gamma}_{24} = -0.34$.

Taking the expectation of (2), we see that the expected value for every observation Y_{ijk} from group ij is $\mu + \alpha_i + \beta_j + \gamma_{ij}$. If we let $i = j = 1$, we get that the expected value of any observation in group $(1, 1)$ is $\mu + \alpha_1 + \beta_1 + \gamma_{11} = \mu$ for treatment contrasts with both reference levels equal to one.

The estimate $\hat{\mu}$ of μ in (2) corresponds to the fitted value for the combination of the two reference levels. This value is called the `Intercept` and estimated as $\hat{\mu} = 2.9375$ kg/ha. It is the estimated yield for the $(\text{gen} = \text{Barkant}, \text{density} = 1)$ combination.

The `R` output contains no estimates of α_1, β_1 or γ_{11} because they are set to zero. We estimate $\hat{\alpha}_2 = -1.6000$ kg/ha. Looking at (2) again, we see that the expectation of every observation from group $(2, 1) = (\text{Marco}, 1)$ is equal to $\mu + \alpha_2$.

The estimate $\hat{\alpha}_2$ corresponds to the estimated effect of the Marco genotype at density 1 in comparison to the Barkant genotype at density 1 (which is the combination of the reference levels). So the yield of the Marco variety at density 1 is estimated to be 1.6 units lower (getting $2.9375 - 1.6000 = 1.3375$ kg/ha) than the yield of the Barkant variety at density 1.

More systematically, the fitted values for the combinations are

Variety	Density 1	Density 2	Density 3	Density 4
		Treatment contrasts		
Barkant	μ	$\mu + \beta_2$	$\mu + \beta_3$	$\mu + \beta_4$
Marco	$\mu + \alpha_2$	$\mu + \alpha_2 + \beta_2 + \gamma_{22}$	$\mu + \alpha_2 + \beta_3 + \gamma_{23}$	$\mu + \alpha_2 + \beta_4 + \gamma_{24}$

We estimate $\hat{\beta}_2 = 1.4625$ kg/ha, which is the effect of density 2 *in comparison to density 1 for the Barkant genotype.*

In the same manner, we can use $\hat{\beta}_3$ and $\hat{\beta}_4$ to compare the respective densities to density 1 for the Barkant genotype. These were the main effects.

How to interpret the interaction effects? If we want to compare (Barkant, 1) with (Marco, 2), we see from the table above that we have to compare μ (for Barkant, 1) with $\mu + \alpha_2 + \beta_2 + \gamma_{22}$. Note that α_2 is the main effect of the Marco genotype and that β_2 is the main effect of density 2. If $\gamma_{22} = 0$, the main effects are sufficient, so γ_{22} is a correction to account for the effect of the *combination* of Barkant and density 2.

If $\gamma_{ij} > 0$, then the combined effect of level i of factor X and level j of factor Z is bigger than the sum of the main effects; this is also called a positive interaction. If $\gamma_{ij} < 0$, this is called a negative interaction.

We will see below how to test whether the interaction effects are statistically significant or whether setting all interaction terms to zero is consistent with the data.

By the way: for this balanced design, the combination of the above estimates yields precisely the sample means.

```
> with(turnip, tapply(yield, list(gen, density), mean))

#              1   2      4      8
# Barkant 2.9375 4.4 9.0875 9.6625
# Marco   1.3375 2.3 5.5625 7.7250
```

7.4 F tests

7.4.1 The overall F test

The first question to ask is whether the model using the two factors and their interaction is any better than just predicting the overall mean for every observation. That latter model is called the *null model*, corresponding to the null hypothesis that all α, β and γ coefficients are equal to zero, which is tested with the *overall F test*.

The overall F test is performed and displayed in the last line of the model **summary**.

```
> summary(turnip.full)
> #F-statistic: 4.338 on 7 and 56 DF,  p-value: 0.0006739
```

For the **turnip** data, the null hypothesis of all coefficients except the intercept being zero is clearly rejected. If this had not been the case, we should have thought about changing our approach/variables, etc.

One should *not* proceed with the tests given below if the overall F test is not significant.

7.4.2 The ANOVA table and the F tests for main and interaction effects

To model the yield (after a significant overall F test),

- do we need the interaction of the two factors? This corresponds to testing the null hypothesis that all γ_{ij} terms are zero.

- Do we need the genotype factor? This corresponds to testing the null hypothesis that all α_i terms are zero.

- Do we need the planting density factor? This corresponds to testing the null hypothesis that all β_j terms are zero.

These three null hypotheses are tested with *partial F tests*, and the results are again summarized in an ANOVA table.

```
> anova(turnip.full)

# Analysis of Variance Table
#
# Response: yield
#              Df  Sum Sq Mean Sq F value    Pr(>F)
# gen           1   83.95  83.951  4.5279 0.0377628 *
# density       3  470.38 156.793  8.4565 0.0001002 ***
# gen:density   3    8.65   2.882  0.1555 0.9257472
# Residuals    56 1038.30  18.541
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

> *Using **`anova()`** is in general only correct for a* balanced *design. See Section 7.9 for the unbalanced case.*

We now get one F test for each factor and one for the interaction. The first thing to check is always the significance of the interaction effect (i. e. of $H_0 : \gamma_{ij} = 0$ for all i, j).

> A significant interaction effect should never be removed from the model, a non-significant interaction effect may be removed if you are not interested in it.
> If the interaction effect is significant, do not remove any of the involved main effects from the model, regardless of what the F tests say (ignore these F tests).
> If the interaction effect is not significant, you may exclude non-significant involved main effects to simplify the model if this is the aim.

For the `turnip` data, the null hypothesis that the interaction effect is zero can not be rejected: the data show no evidence of an interaction effect. If the main question was about the interaction effect (here: *does the density effect depend on the variety?*), then the main question is answered now.

If the primary interest was not in the interaction effect, but in the main effects, one common practice is to keep the interaction effect in the model if it is significant, but to fit a model without interaction effect if it is not significant. In that way, one can gain power for testing the main effects. To fit a model without interaction effects, but only with main effects (i. e. an *additive* model), use

```
> turnip.main <- lm(yield ~ gen + density, data = turnip)
```

For the `turnip` data, the two null hypotheses that the main effects are zero can be rejected (at a significance level of 5%).
Summarizing, we find clear evidence of a genotype and a planting density effect, but no evidence of an interaction effect. Accordingly, the interaction effect may be dropped from the model, but the genotype and the planting density effect are needed.

7.5 Tests on individual coefficients

If the partial F tests in Section 7.4.2 reject the null hypothesis that all coefficients belonging to some particular factor (or to the interaction) are zero, we can also test whether individual coefficients are zero.
The F test for the density factor was significant, which means that the null hypothesis that $\beta_2 = \beta_3 = \beta_4 = 0$ is rejected.[23]

> Remember what these coefficients mean: they correspond to the effect of the particular density, *compared to the reference density and for the reference level of the genotype*. Testing whether these coefficients are zero may not be answering a meaningful research question.

This does not mean that *all* these three coefficients are significantly different from zero, it only means that at least one coefficient is nonzero. To find the ones which are significantly different from zero, we test the null hypothesis that particular *individual coefficients* are zero with the corresponding t test given in the model `summary`, printed in Section 7.3.4 for the turnip data model.

Every row of the `Coefficients` block contains a coefficient `Estimate` with its `Standard Error`. To be explicit, consider the first row. It contains the estimate $\hat{\mu} = 2.9375$ with standard error 1.5224. Dividing the estimate by its standard error gives the t statistic. We may directly interpret the last column containing the p values belonging to the t test of the null hypothesis that a particular coefficient is zero.
For example, the p value for testing whether μ is significantly different from zero is 0.0587, so that we can not reject the null hypothesis that $\mu = 0$ at a significance level of 0.05. Keep in mind that μ is the theoretical mean of the yield for the Barkant variety at density 1.[24]
Slight modifications allow also other tests (for example, testing whether β_4 is significantly bigger than four against the alternative that it is not). It is also possible to conduct

[23] Remember that due to treatment contrasts, $\beta_1 = 0$.

[24] The model summary does not necessarily test hypotheses which are interesting in a particular research setting.

tests on linear combinations of coefficients (for example, to test $\beta_2 - \beta_3 = 0$, i. e. whether $\beta_2 = \beta_3$.) This can be done with the `glht` function from the `multcomp` library.

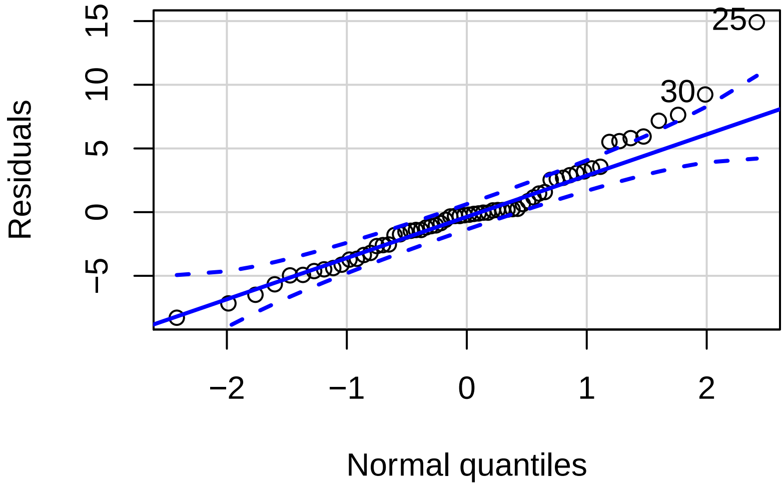
7.6 Model diagnostics and related topics

The right time to do model diagnostics is immediately after fitting the model. Here, we have postponed this because this data set has diagnostic problems, but we did not want to make the coefficient interpretation more complicated than necessary.

7.6.1 Graphical normality checking

This works exactly as for one-way ANOVA. Consider the normal QQ plot of the residuals:

```
> library(car)
> qqPlot(resid(turnip.full), xlab = "Normal quantiles", ylab = "Residuals")
```



```
# [1] 25 30
```

The QQ plot shows some problems with a perhaps slightly heavy upper tail and one outlier. Robust or nonparametric methods could be used to control the effect of the outlier.

Another strategy is to fit the model one time with and one time without the outlier and compare results to assess model robustness.[25] If the results change a lot when the outlier is in-/excluded, this indicates a problem.

7.6.2 Testing normality

The Shapiro-Wilk test is performed with

```
> shapiro.test(resid(turnip.full))
```

and yields a p value of 0.0132. The null hypothesis of normality is rejected, and the residuals indeed have problems with normality (perhaps in this case the Shapiro-Wilk test is rather influenced by the outlier than by the shape of the right tail of the residual distribution. Exclude the outlier and refit the model to find out. Diagnostic tests are often very sensitive to outliers in small samples.)

7.6.3 Accounting for non-normality

> We have two options now. We can abandon normal distribution based methods for e. g. nonparametric, robust or permutation methods, or we can try to find a transformation g such that an ANOVA of the sample $g(Y)$ produces residuals which looks as if they came from a normal distribution.

It is often not easy to find the right transformation, and systematic tools (such as Box-Cox transformations) can help. The transformation $g(y) = \sqrt{y}$ solves the non-normality problems. The square root function is called `sqrt` in `R`.

```
> turnip.sqrt <- lm(sqrt(yield) ~ gen * density, data = turnip)
> ## qqPlot(resid(turnip.sqrt))
> ## shapiro.test(resid(turnip.sqrt))
```

We obtain a perfect QQ plot and the Shapiro-Wilk test now gives $p = 0.9696$.

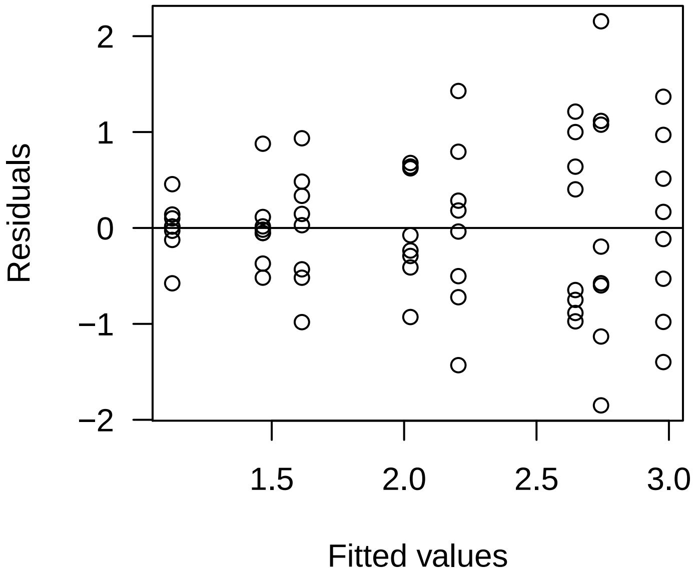
> All subsequent analyses are performed with the square root of the yield as dependent variable and then transformed back after analysis by squaring.

7.6.4 Residual plots

We plot the residuals vs. the fitted values.

[25] This idea is related to influence statistics discussed in the D3 module.

```
> plot(fitted(turnip.sqrt), resid(turnip.sqrt), xlab = "Fitted values",
+      ylab = "Residuals", las = 1); abline(h = 0)
```



The variance of the residuals increases still increases with the fitted values.

7.6.5 Testing variance homogeneity

The generic code to perform the two tests in two-way ANOVA with dependent variable Y and the two factors X and Z is

```
> bartlett.test(Y ~ interaction(X, Z), data = data)
> leveneTest(Y ~ interaction(X, Z), data = data)
```

To perform Bartlett's test and the Levene test in two-way ANOVA, the right-hand side of the formula should always be as in the code above, no matter whether the model did contain an interaction effect. The `interaction` term only specifies the factorial treatment structure.

The `sqrt` term on the left hand side of the formula below is specific to this example and should not be used in general.

```
> bartlett.test(sqrt(yield) ~ interaction(gen, density), data = turnip)
> leveneTest(sqrt(yield) ~ interaction(gen, density), data = turnip)
```

We obtain $p = 0.0074$ for Bartlett's and $p = 0.0072$ for Levene's test, both clearly rejecting the null hypothesis of equal variances.

7.6.6 Accounting for heterogeneous variances

We could try to find a transformation g such that the variances of $g(Y)$ are more comparable. But this could lead to new problems with the normality of the residuals. Instead, we use robust methods.

> These methods may also be used without previous use of a screening test for heteroskedasticity (such as Levene's Test). If there are doubts about equal variances, then it is prudent to use such methods regardless of tests for equal variances (which might have too little power to detect variance heterogeneity in small samples).

```
> library(car)
> Anova(turnip.sqrt, white.adjust = "hc3")

# Coefficient covariances computed by hccm()

# Analysis of Deviance Table (Type II tests)
#
# Response: sqrt(yield)
#              Df       F    Pr(>F)
# gen           1  8.8767  0.004265 **
# density       3 12.4203 2.461e-06 ***
# gen:density   3  0.0549  0.982910
# Residuals    56
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This method may be used for data with unequal variances, and it adjusts the p values for the unequal variances by using so-called heteroskedasticity-corrected covariance matrices. The option `white.adjust = "hc3"` requests a particular method which in simulation studies was observed to perform well also for small samples. See the help files for further references (the inventors were Huber and White).

> We are not removing the heteroskedasticity here, but we are using a statistical estimation method that can deal with it. It should be noted that this technique does not solve non-normality problems or problems with outliers.

7.7 Fitted values and confidence intervals

To obtain the fitted value (i. e. the mean) for each group according to our model, the right coefficients have to be added, as explained in Section 7.3.4. We show how to do this for the model with interaction terms. To show the principle, we do it “manually”.

```
> coef(turnip.sqrt)

#       (Intercept)          genMarco          density2          density4
#           1.61393          -0.48964           0.40986           1.12971
#          density8 genMarco:density2 genMarco:density4 genMarco:density8
#           1.36489          -0.06774          -0.04908           0.15760
```

What is the fitted value of $\sqrt{\text{yield}}$ for the Marco genotype with density 8? It is $1.61393 - 0.48964 + 1.36489 + 0.15760 = 2.64678$. Simply add to the intercept the coefficient for the Marco variety (main effect of genotype), the coefficient for the density 8 (main effect of density) and the interaction effect that corresponds to the combination of Marco with density 8. To go from here to yield, we square and get $2.64678^2 = 7.005$ for the predicted yield, compared with the sample mean of 7.725.

In practice, we use the following combination of `predict` with the convenience function `expand.grid` that creates a dummy data frame `dat` with all combinations of the given factors.[26]
That dummy data frame is then fed to `predict` after the name of the model, and the results are bound together (by columns) with `cbind`. We also show how to add 95% confidence intervals here.

```
> dat <- expand.grid(gen = levels(turnip$gen),
+                    density = levels(turnip$density))
> pred <- predict(object = turnip.sqrt, newdata = dat,
+                 interval = "confidence")
> (ci <- cbind(dat, pred^2))

#       gen density   fit    lwr   upr
# 1 Barkant       1 2.605 1.0773 4.796
# 2   Marco       1 1.264 0.3006 2.891
# 3 Barkant       2 4.096 2.0962 6.759
# 4   Marco       2 2.150 0.7929 4.171
```

[26] Instead of writing `c("Marco", "Barkant")`, we directly access the levels of e. g. the genotype factor with `levels(turnip$gen)`. The dummy data frame must contain values for each variable in the model; take care to use exactly the same names for the variables as in the original data set. The dummy data frame `dat` consists of the first two columns in the `R` output.

```
# 5 Barkant       4 7.528 4.6987 11.020
# 6   Marco       4 4.862 2.6534  7.733
# 7 Barkant       8 8.873 5.7736 12.637
# 8   Marco       8 7.005 4.2882 10.386
```

If we had not transformed the data, squaring in the last line would be omitted.

7.8 Post hoc tests

We now discuss some post hoc tests based on least squares means/estimated marginal means. Suppose you want to compare the yield of the two genotypes. You could now *average over the planting densities*, which should only be done if the genotype effect is more or less the same for each planting density. Since this appears to be the case given the interaction plot and the non-significant interaction effect, let us do this.

```
> emmeans(turnip.sqrt, pairwise ~ gen)

# NOTE: Results may be misleading due to involvement in interactions

# $emmeans
#  gen     emmean     SE df lower.CL upper.CL
#  Barkant  2.340 0.1438 56    2.052    2.628
#  Marco    1.861 0.1438 56    1.573    2.149
#
# Results are averaged over the levels of: density
# Results are given on the sqrt (not the response) scale.
# Confidence level used: 0.95
#
# $contrasts
#  contrast        estimate     SE df t.ratio p.value
#  Barkant - Marco   0.4794 0.2033 56   2.358  0.0219
#
# Results are averaged over the levels of: density
```

On average, over all the planting densities, the Barkant genotype produces significantly higher yields.
As `emmeans` tries to warn you, it is not necessarily a good idea to average over the planting density levels, because maybe for some densities the difference between the two genotypes is more pronounced than for others. It could even be that for a particular planting density, the Marco variety gives higher yields. Then, averaging may completely

distort the results. In the presence of interactions, it would be more sensible to compare the genotypes *separately for each planting density.* Here is how to do this.

```
> turnip.sqrt.gen <- emmeans(turnip.sqrt, pairwise ~ gen | density)
> turnip.sqrt.gen$contrasts

# density = 1:
#  contrast         estimate     SE df t.ratio p.value
#  Barkant - Marco    0.4896 0.4066 56   1.204  0.2336
#
# density = 2:
#  contrast         estimate     SE df t.ratio p.value
#  Barkant - Marco    0.5574 0.4066 56   1.371  0.1759
#
# density = 4:
#  contrast         estimate     SE df t.ratio p.value
#  Barkant - Marco    0.5387 0.4066 56   1.325  0.1906
#
# density = 8:
#  contrast         estimate     SE df t.ratio p.value
#  Barkant - Marco    0.3320 0.4066 56   0.817  0.4176
```

Because the standard errors are now higher, the separate comparisons no longer produce significant differences. This is why you should exclude non-significant interactions from a model. The model should not be more complex than necessary.
It is also possible to compare all the combinations of genotype and planting density pairwisely (for this design, it is questionable why anyone would want to do this).

```
> emmeans(turnip.sqrt, pairwise ~ gen + density)
```

In a different context, it could absolutely make sense to do this. For example, you could compare the average D1 exam scores for all the four combinations of “taking D2” and “taking D3”.

7.9 Unbalanced designs

In the unbalanced higher-way ANOVA, the sums of squares no longer add up as they do in the balanced case. This has lead to many discussions about the Right Way of defining sums of squares among statisticians.

> The main consequence for us is that we now need to distinguish between *sequential* and *marginal* F tests.

The sequential tests are performed by `anova()`, but they usually do not test the hypotheses that researchers are interested in: they test whether adding an effect *to the effects already in the model* significantly reduces the residual sum of squares. In consequence, the order of terms in the model formula matters for Type I tests, cf. Milliken and Johnson 2009, Ch. 10.2 or Fox and Weisberg 2011, Ch. 4.4.
Type II tests assess whether an effect significantly reduces the residual sum of squares *adjusted for all the other effects at the same or lower level.* The order of terms in the model equation does not matter.

> I recommend so-called Type II tests (marginal F tests) which may be obtained with `Anova(model, type = 2)`. Remember not to test main effects if they are involved in a significant interaction term.

To use `Anova`, load `car` first; `model` is our model fitted with `lm()`.
The results are the same because the turnip data are balanced.

```
> anova(turnip.sqrt)
> library(car)
> Anova(turnip.sqrt, type = 2)
```

In addition, there also exist Type III tests, which test slightly different hypotheses regarding the main effects, cf. Milliken and Johnson 2009, Ch. 10.4. We use a different strategy: if the interaction is significant, we do not test main effects. If the interaction is not significant, we can remove it, fit an additive model and test hypotheses about the main effects with Type II tests.

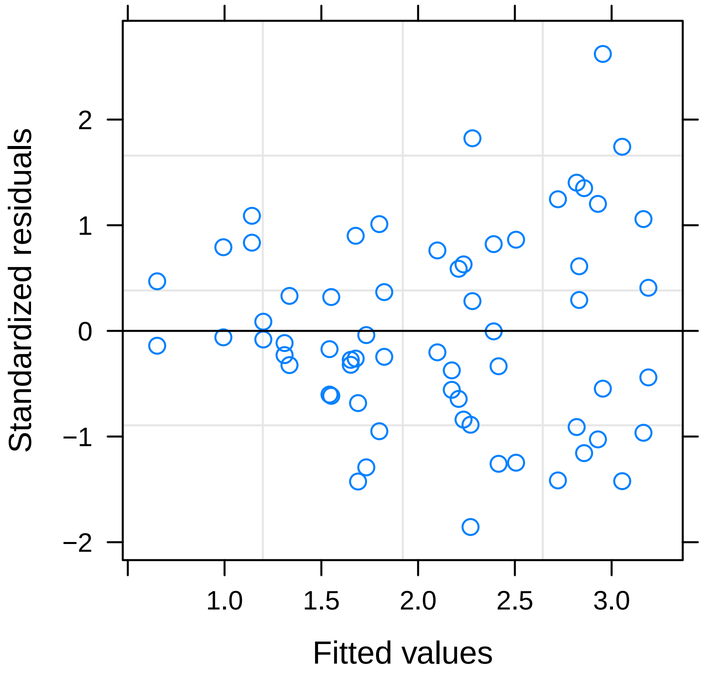
7.10 Accounting for blocks and for heteroskedasticity

Let us now show how to account for the blocking structure of the turnip data. With the ideas from Chapter 6, a simple strategy is to use a linear mixed-effects model with a random intercept per block:

```
> turnip.lme <- lme(sqrt(yield) ~ gen * density, random = ~ 1 | block,
+                   data = turnip)
```

This model combines the factorial treatment structure with the random intercept per block.
Let us run the diagnostics before we interpret the model:

```
> plot(turnip.lme)
```



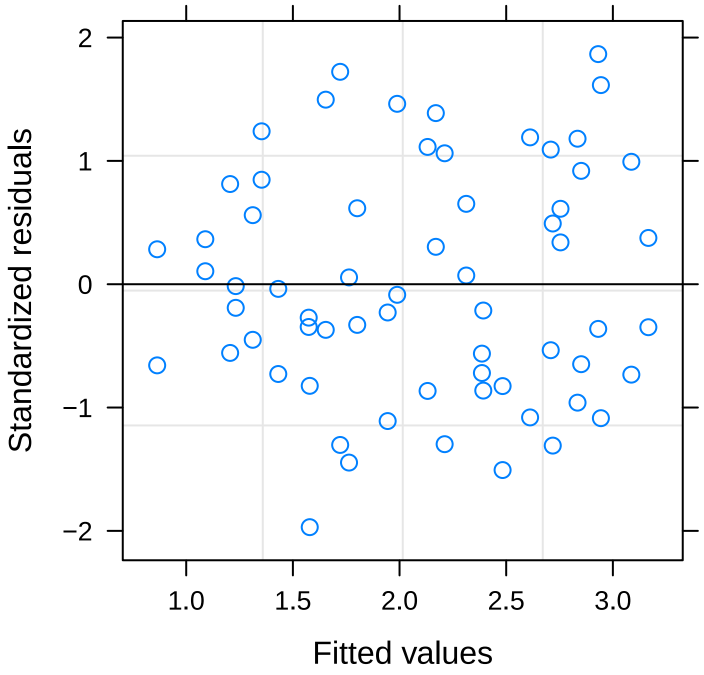
```
> ## qqPlot(resid(turnip.lme))
> ## shapiro.test(resid(turnip.lme))
```

Normality looks perfect, but the variances still increase with the fitted values.

Heteroskedastic within-group errors

It is possible to specify that the variance should be modeled in a specific way. This is implemented in the `varFunc` objects; we only show how to estimate a separate variance for each level of a factor here, which can be done with `varIdent`. Also more complicated variance models are possible.
Because the variance of the residuals seems to increase with planting density, we specify that the mixed model should estimate a separate variance for each level of the density. Note the `Variance function` part of the model output.

```
> turnip.lme.het <- lme(sqrt(yield) ~ gen * density, random = ~ 1 |block,
+                             data = turnip,
+                             weights = varIdent(form = ~ 1 | density))
> turnip.lme.het

# Linear mixed-effects model fit by REML
#   Data: turnip
#   Log-restricted-likelihood: -67.34
#   Fixed: sqrt(yield) ~ gen * density
#       (Intercept)          genMarco          density2          density4
#           1.61393          -0.48964           0.40986           1.12971
#          density8 genMarco:density2 genMarco:density4 genMarco:density8
#           1.36489          -0.06774          -0.04908           0.15760
#
# Random effects:
#  Formula: ~1 | block
#         (Intercept) Residual
# StdDev:      0.2333   0.4806
#
# Variance function:
#  Structure: Different standard deviations per stratum
#  Formula: ~1 | density
#  Parameter estimates:
#     1     2     4     8
# 1.000 0.961 2.194 1.809
# Number of Observations: 64
# Number of Groups: 4

> plot(turnip.lme.het)
```



The unequal variances problem appears to be completely solved. The analysis can now proceed as usual for linear mixed models:

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

#             numDF denDF F-value p-value
# (Intercept)     1    53   61.33  <.0001
# gen             1    53    4.15  0.0466
# density         3    53    6.40  0.0009
# gen:density     3    53    0.07  0.9744
```

We still conclude that the interaction is not needed and now simplify the model:

```
> turnip.lme.het.add <- lme(sqrt(yield) ~ gen + density, random = ~ 1 |block,
+                           data = turnip,
+                           weights = varIdent(form = ~ 1 | density))
> anova(turnip.lme.het.add, type = "marginal")
```

Both main effects remain significant, and we could now look into post hoc tests.

8 Classical analysis (ANOVA tables)

8.1 Completely randomized designs

interested in mean
model analysis the variance of mean

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*

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

how do you define the degree of freadom? when you have fixed and random effect in the model

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))
```

bloack used as random effect by tell as Error()

```
#
# Error: block
#           Df Sum Sq Mean Sq F value Pr(>F)
# Residuals  5   5.55    1.11
                 changed by bolck
#
# 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
# ---     66 before  before 26.1, changed dramatical, the difference
# 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)   mixed effect model   do not have aduste the model, directly use the model setting
> 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   364.8  <.0001
# spray           5    61    52.6  <.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.

same r k is less than v, each block doesnot have all the treatments

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. design is binary: each treatment in block: yes or no
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):

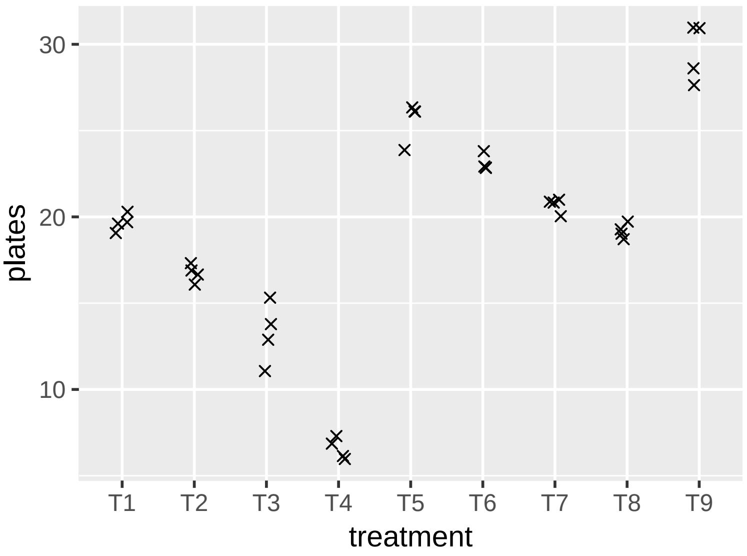
```
> library(ibd)
> set.seed(1) # to always get the same design
                 lambda:1
> bibd(7, 7, 3, 3, 1)$design
   v=7, b=7, r=3, k=3,
#          [,1] [,2] [,3]
# Block-1     5    6    7   treatment 7: 3 times
# Block-2     1    4    5
# Block-3     1    3    7
# Block-4     3    4    6
# Block-5     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)

                     how many plates will be washed
#   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 three different of treatment in each 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

interation between block nd treatment: treatment will be different from block to block

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

$$Y_{ji} = \mu + \beta_j + \vartheta_i + \varepsilon_{ji}$$

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.

fiexed effect

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)      what is the treatment effect after we adjust the block effect

# Analysis of Variance Table
#
# Response: plates
#           Df Sum Sq Mean Sq F value  Pr(>F)
# block     11    413    37.5    45.5 6.0e-10 *** sum of squear for block
# treatment  8   1087   135.9   164.8 6.8e-14 *** sum of square given the sum of square for
# Residuals 16     13     0.8                        block
# ---
# 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

$$Y^*_{ji} = Y_{ji} - (\hat{\vartheta}_i - \hat{\bar{\vartheta}}) \quad \text{overall mean}$$

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  0.3611     12.639
# 2   B01        T8     20    2        0         1  0.3611     19.639
# 3   B01        T4      7    1        0         1  0.3611      6.639
# 4   B02        T4      6    1        0         2 -0.4167      6.417
# 5   B02        T9     29    3        0         2 -0.4167     29.417
# 6   B02        T2     17    1        2         2 -0.4167     17.417
```

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). estimate the block effect and remove the effect

9.2.4 Mixed effects model – interblock analysis

block effect as random effect
block is not the main interest of the 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   13943  <.0001
# treatment       8    16     221  <.0001
```

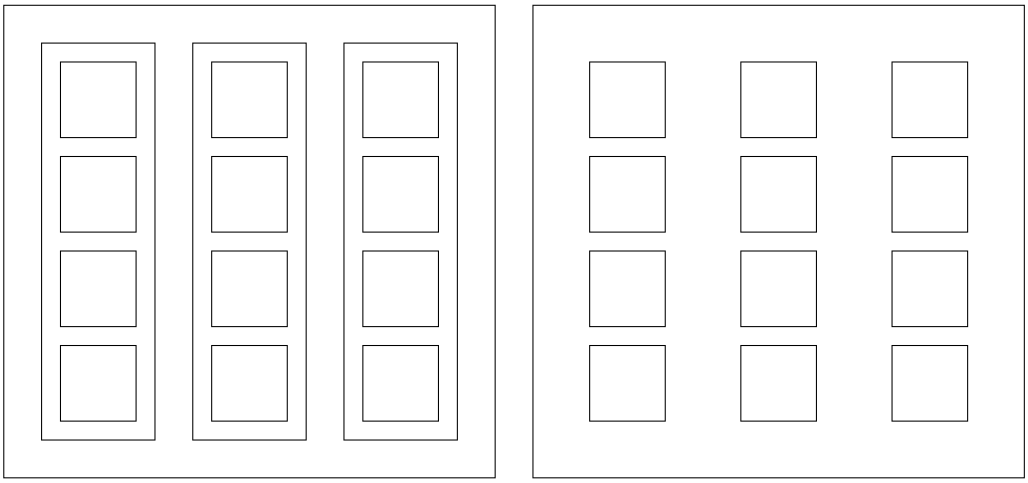
if it is a complete design, r code is the same!!

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: six fields, each one



r can not distinguish between this two design in input data

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.

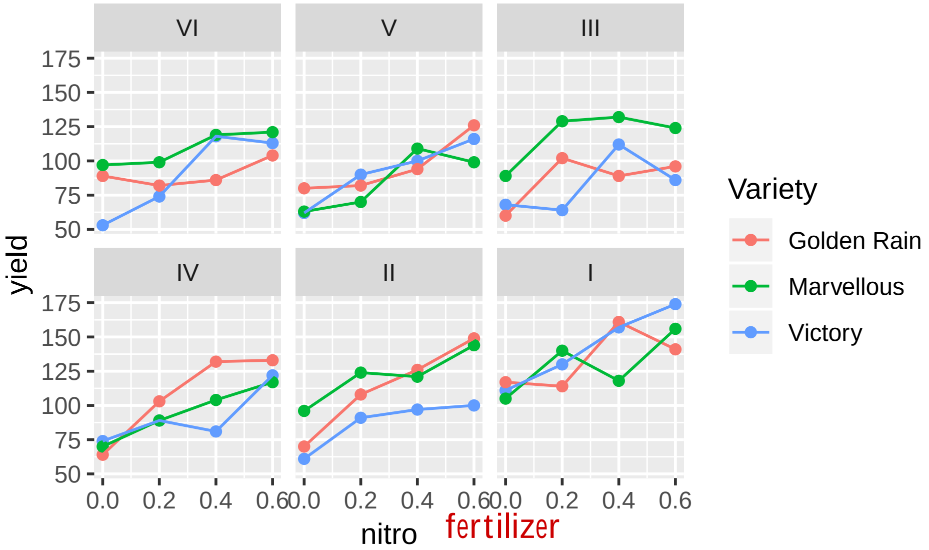
-:main plot fact become less percise

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?).

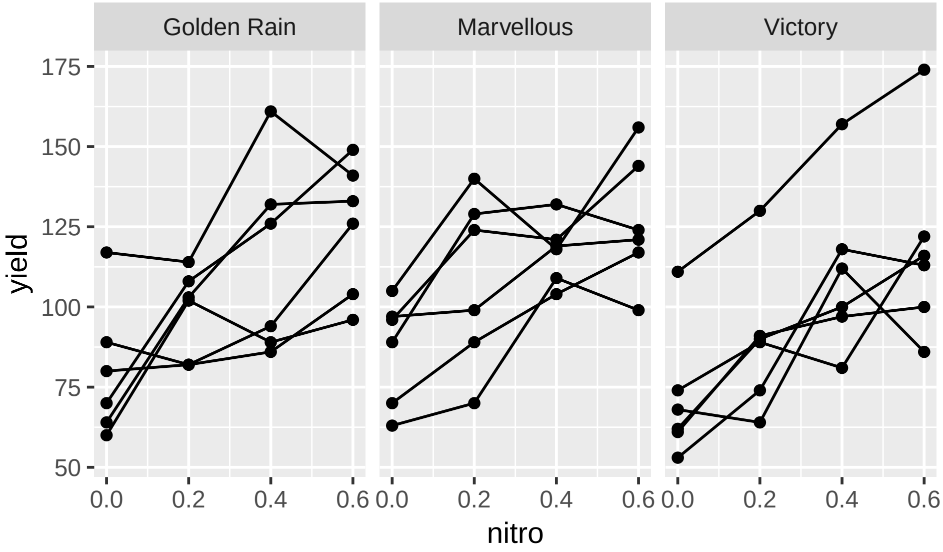
To guide the eye, it is possible to connect the plots from the same group with a line. We use color here; in scientific journals, using color is often expensive, so different symbols (`shape` aesthetic in `ggplot2`) tend to be used instead of colors.

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



fertilizer

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



focus on the oats
on the veriaty levels

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 two factors:

- 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,

additional random effect:

- $b_{i(k)} \sim \mathcal{N}(0, \sigma^2_{\text{W}})$ is the whole plot i random intercept *within* block k, new here: random effect for the plot, the plot is within block
- $\varepsilon_{ijk} \sim \mathcal{N}(0, \sigma^2)$ 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)
```

nested random effect
the random eddeft the variety is nested within random effect block

```
# Linear mixed-effects model fit by REML
#  Data: Oats
#   AIC   BIC logLik
#   559 590.4 -264.5
#
# Random effects:
#  Formula: ~1 | Block
#          (Intercept)
# StdDev:       14.64
                                       ^ σblock = 14:65,
                                      ^ σW = 10:30
#                                     and ^ σ = 13:31
#  Formula: ~1 | Variety %in% Block
#          (Intercept) Residual
# StdDev:         10.3    13.31
#
# Fixed effects: yield ~ Variety * nitroF
#                               Value Std.Error DF t-value p-value
# (Intercept) golden Rain, nitroF0
# (Intercept)                   80.00     9.107 45   8.784  0.0000
# VarietyMarvellous              6.67     9.715 10   0.686  0.5082
# VarietyVictory                -8.50     9.715 10  -0.875  0.4021
                         MEAN for the golden rain: intercept + 18.50
# nitroF0.2                     18.50     7.683 45   2.408  0.0202
                     MEAN for the marvellous: intercept + 6.67+ 18.50
# nitroF0.4                     34.67     7.683 45   4.512  0.0000
# nitroF0.6                     44.83     7.683 45   5.835  0.0000
# VarietyMarvellous:nitroF0.2  3.33    10.865 45   0.307  0.7604
# VarietyVictory:nitroF0.2     -0.33    10.865 45  -0.031  0.9757
# VarietyMarvellous:nitroF0.4 -4.17    10.865 45  -0.383  0.7032
# VarietyVictory:nitroF0.4      4.67    10.865 45   0.430  0.6696
# VarietyMarvellous:nitroF0.6 -4.67    10.865 45  -0.430  0.6696
# VarietyVictory:nitroF0.6      2.17    10.865 45   0.199  0.8428
#
# Standardized Within-Group Residuals:
#      Min        Q1       Med        Q3       Max
# -1.81301 -0.56145  0.01758  0.63864  1.57034
#
# 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    77.17   <.0001
# Variety          2    10     1.22   0.3344
# nitroF           3    45    13.02   <.0001
# Variety:nitroF   6    45     0.30   0.9322
```

variaty is not significant

interaction is not significant, then we can also remove the interacton effect

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:

numerator degree:
denumerator degree:

分子自由度是自变量的个数，记为k，则分母自由度为n-k-1，n为样本个数

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:

change from * to +. we remove the interaction effect between variaty and nitro

```
> 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   94.51  <.0001
# Variety         2    10    1.49  0.2724
# nitroF          3    51   41.05  <.0001
```

marginal: one at the time
sequential:

(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) 103.9722     6.641 51 15.6569 4.030e-21
# nitroF.L     32.9447     3.005 51 10.9627 5.121e-15
```

natural order exist
N={A,B,C,D}
Ni={A<b<c<d}
0<0.2<0.4<0.6

nitro 0??
linear effect

```
                qudratic effect
# nitroF.Q      -5.1667     3.005 51 -1.7193 9.163e-02
# nitroF.C      -0.4472     3.005 51 -0.1488 8.823e-01
             orthogonal cubic trend
```

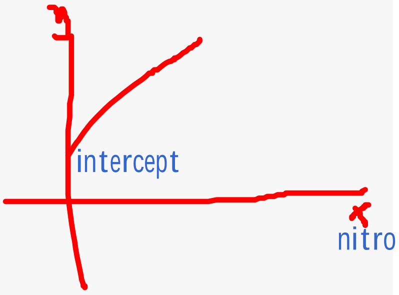
This has the consequence that now, so-called *orthogonal polynomial* contrasts are used for the nitrogen factor. Briefly, the first contrast (`nitroF.L`) estimates the linear trend, the second contrast (`nitroF.Q`) estimates the quadratic effect orthogonal (independent) of the linear term, and `nitroF.C` 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  79.39
# 2    0.2  98.89
# 3    0.4 114.22
# 4    0.6 123.39

> ## 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
#   603 614.3 -296.5
#
# Random effects:  this is the same like before, randon effect as block
#  Formula: ~1 | Block
#         (Intercept)
# StdDev:       14.51
#
```

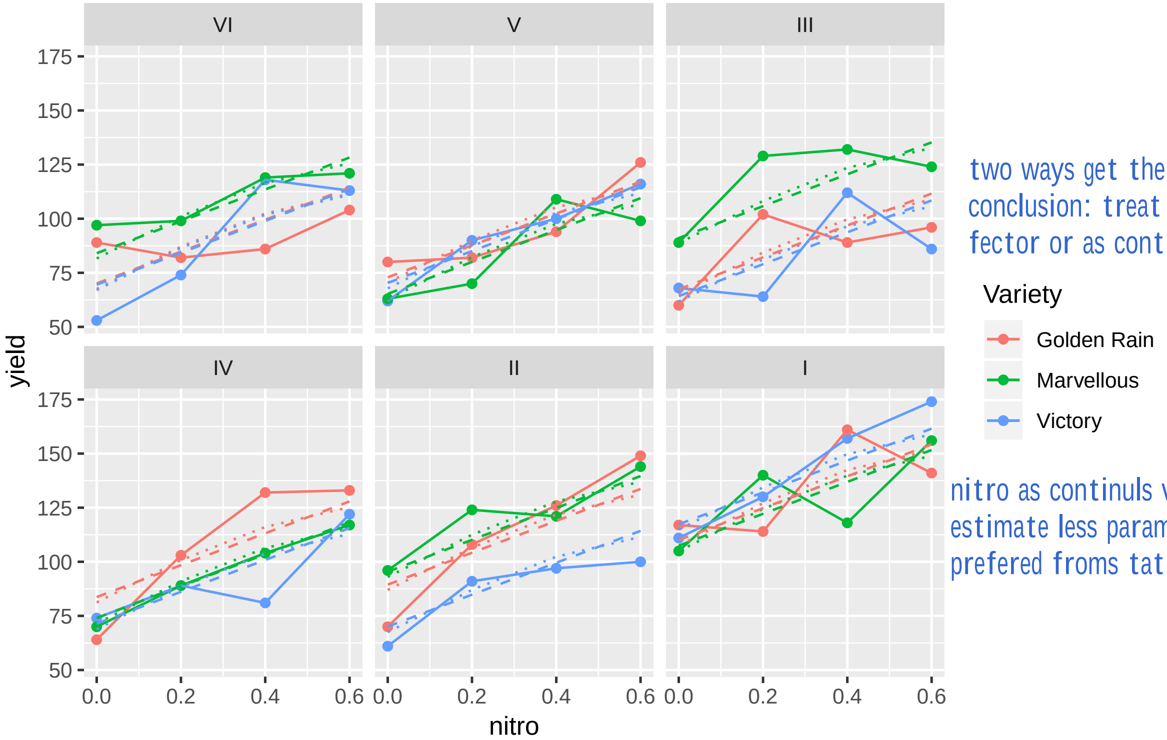
```
#  Formula: ~1 | Variety %in% Block nested random effect
#         (Intercept) Residual
# StdDev:          11    12.87
#
# Fixed effects: yield ~ nitro
#             Value Std.Error DF t-value p-value
# (Intercept) 81.87     6.945 53   11.79       0
# nitro       73.67     6.781 53   10.86       0
#
# Standardized Within-Group Residuals:
#     Min      Q1     Med      Q3     Max
# -1.7438 -0.6648  0.0171  0.5430  1.8030
#
# Number of Observations: 72
# Number of Groups:
#              Block Variety %in% Block
#                  6                 18
```



nitro change from factor to ordered factor(as continues variable)

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)
```



two ways get the same conclusion: treat nitro as a fector or as continuals variables

nitro as continuls variable, we estimate less paramater, which is prefered froms tatistic view

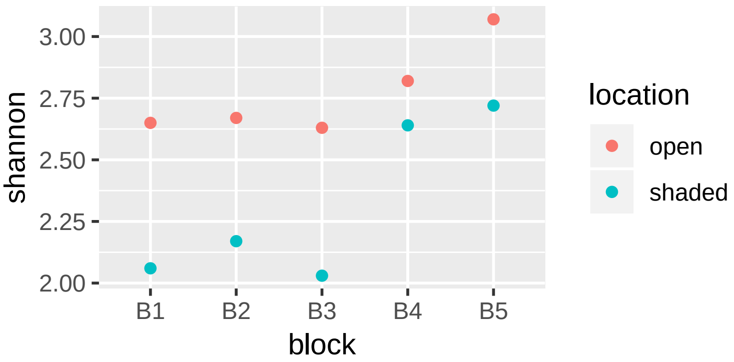
The data are connected by solid lines, the factor model fitted values by dotted lines, and the linear regression model fitted values by dashed lines.
The two models yield very similar fitted values in all blocks. In case a linear effect provides a reasonably accurate description of the nitrogen-yield 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

randome complex block design

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)
```

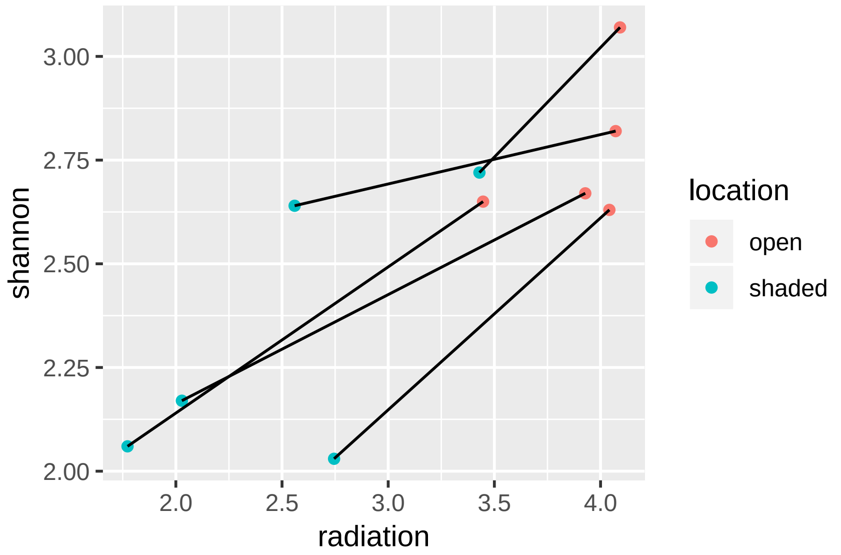
The Friedman test finds a significant location effect. To use covariates below, we use a parametric mixed effects model now, although the data set is perhaps too small.

```
> 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   0.11969  4  23.127 2.071e-05
# locationshaded -0.444   0.07979  4  -5.565 5.107e-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.

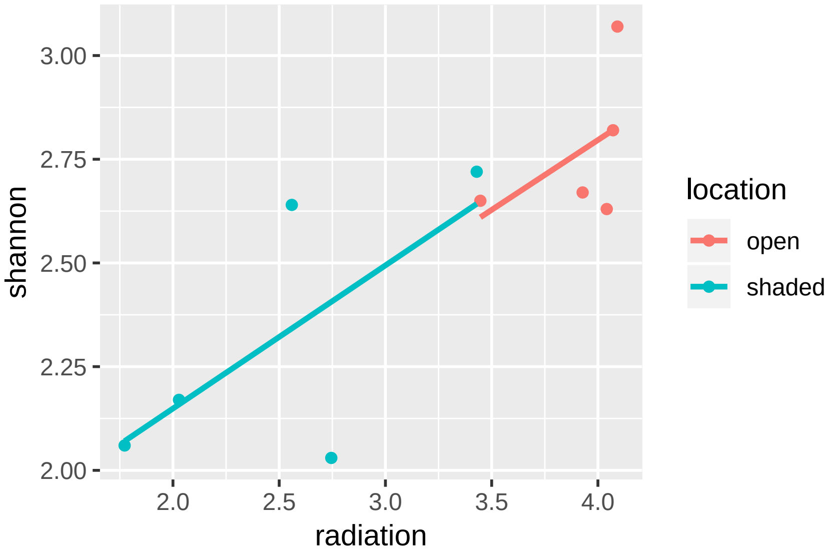
here we use * including the interaction effect

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

#                             Value Std.Error DF t-value p-value
# (Intercept)               1.95112    1.5909  4  1.2264  0.2873
# radiation                      0.20861    0.4054  2  0.5146  0.6581
# locationshaded                -0.26603    1.4583  2 -0.1824  0.8721
# radiation:locationshaded       0.04624    0.3654  2  0.1265  0.9109
```

interaction is not significant different

do anova test to test if the interaction can be removed or not

```
> 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

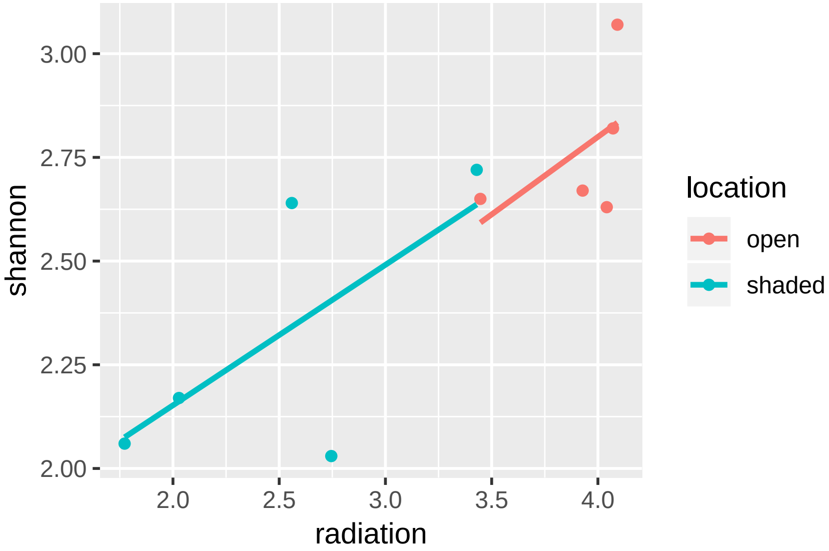
$$\text{shannon} = \begin{cases} 1.95 + 0.21 \cdot \text{radiation} & \text{for } \texttt{open} \text{ locations}\,, \\ (1.95 - 0.27) + (0.21 + 0.046) \cdot \text{radiation} & \text{for } \texttt{shaded} \text{ locations}\,. \end{cases}$$

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)
```

here we remove the interaction

```
> coef(summary(gasel.lme))

#                   Value Std.Error DF t-value p-value
# (Intercept)      1.8180    0.6181  4   2.941 0.04234
# radiation        0.2426    0.1557  3   1.558 0.21715
# locationshaded  -0.1022    0.2366  3  -0.432 0.69489

> 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

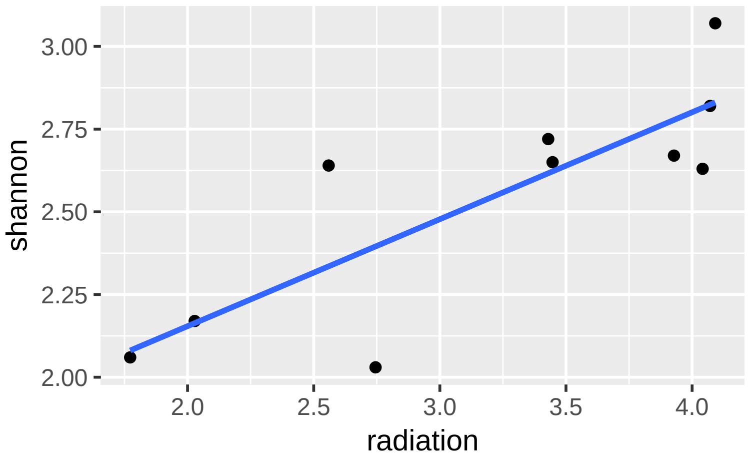
$$\text{shannon} = \begin{cases} 1.81 + 0.24 \cdot \text{radiation} & \text{for } \texttt{open} \text{ locations}, \\ (1.81 - 0.10) + 0.24 \cdot \text{radiation} & \text{for } \texttt{shaded} \text{ locations}. \end{cases}$$

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)
```

here we remove the location effect
just add the radiation

```
> fixef(gasel.lme)

# (Intercept)   radiation
#      1.5625      0.3062

> 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.

acova model

12 Extensions

The aim of this chapter is to give you the main idea (not a full analysis) of some important designs, and a pointer to alternative analysis methods.

12.1 Row-column designs

We start with Bailey 2008, Ex. 4.11, see also her Ch. 6 as well as Dean, Voss, and Draguljić 2017, Ch. 12 for more information.

Example 12.1. *Eight judges each taste and evaluate four wines. The order of the four tastings is randomized within each judge. A plot is one tasting by one judge, so there are 32 plots. A first analysis might only treat the judge as a block, but perhaps the tasters become more or less generous with increasing number of tastings, so perhaps the tasting order should also be used as a blocking variable.*

Designs with two systems of blocks are called *row-column* designs, one system of blocks is called "rows" and the other, "columns". The row and column factors are crossed (each combination of the two is observed). The most simple row-column designs satisfy the following conditions:

1. Each row meets each column in a single plot,
2. all treatments occur equally often in each row,
3. all treatments occur equally often in each column, and
4. there are m rows and n columns.

If these conditions are satisfied, there must be mn plots, the number of treatments t must be a divisor of m and n and each treatment is replicated mn/t times. Of course, it is also possible to have several plots for each combination of row and column blocking level.
For the statistical analysis, the blocks may be treated as having fixed or random effects depending on the context. If both blocking variables are modeled with random effects, we have *crossed random effects.*

12.1.1 Latin squares

The conditions given above are satisfied when $m = n = t$; such designs are called *Latin squares.* Latin squares (of order t) are an arrangement of t symbols in a $t \times t$ square such that each symbol occurs once in each row and once in each column. (Sudokus are order 9 Latin squares with additional local conditions for the small 3×3 squares.)

Example 12.2. *To compare the yield of four crops (A, B, C, D), four fields (I, II, III, IV) are available. Due to agronomic reasons, not every sequence of crops is sensible (crop rotation); the best rotation is A, B, C, D. Then, the following Latin square design could be used:*

	Field			
Year	*I*	*II*	*III*	*IV*
2018	*A*	*D*	*C*	*B*
2019	*B*	*A*	*D*	*C*
2020	*C*	*B*	*A*	*D*
2021	*D*	*C*	*B*	*A*

This Latin square satisfies the added crop rotation condition in each field over time.

A Latin square may be interpreted as a two-dimensional version of the randomized complete block design, since each blocking factor on its own defines an RCBD (ignoring the other factor). A number of designs is related to Latin squares.

Example 12.3. *In agronomy, field plots may be reused over several years. If in year one, a Latin square design was used, and the same treatments will be used on the same plots again in year two, then year two should be using an* orthogonal *Latin square to the first one. A pair of two Latin squares is called orthogonal if each letter of the first square occurs in the same position as each letter of the second square exactly once. (Not each Latin square has a Latin square orthogonal to it.) Two orthogonal Latin squares are often called* Graeco-Latin *squares because of the letters used:*

A	B	C
C	A	B
B	C	A

α	β	γ
β	γ	α
γ	α	β

Refer to Bailey 2008, Ch. 9.3 for further discussion.

Example 12.4. *Consider the following design with four subjects tasting and rating four brands of chocolate and three runs:*

	Run		
Subject	*1*	*2*	*3*
Ari	*A*	*B*	*C*
Bernhard	*B*	*C*	*D*
Chris	*C*	*D*	*A*
Denise	*D*	*A*	*B*

Each subject tries only three chocolate brands. With respect to subjects, we have a balanced incomplete block design. In each run, each chocolate will be tried by exactly one person, so that with respect to runs, a complete block design is used. Such a design is called a Youden *square (although it is not a square.) See Dean, Voss, and Draguljić 2017, Ch. 12 for further discussion.*

12.2 Confounded factorial experiments

We only give an example here to illustrate the main idea, see Dean, Voss, and Draguljić 2017, Ch. 13, 14.
Suppose we are in situation where each plot is very resource-intensive to run. We want to run a factorial experiment with three factors A, B, C, each of which has two levels, called 0 and 1. There are $2^3 = 8$ treatments (combinations of the three factors), which we write as ABC, substituting the values of the factors. For example, 010 means $A = 0, B = 1, C = 0$. Suppose we have only one plot per treatment (a single-replicate experiment), and the experiment is to be run in two blocks.

A short reminder about contrasts

Suppose we estimate any linear model of the kind

$$Y_{ji} = \mu + \beta_j + \varepsilon_{ji}\,.$$

Then any linear combination of the kind $\sum c_j \beta_j$ with $\sum c_j = 0$ is called a *contrast.* We have used contrasts e. g. in Section 3.1 for comparing the treatments with a control. To compare group 1 with group 2, we can use the contrast with $c_1 = -1, c_2 = 1$ and all other $c_i = 0$.

With 8 observations, we only have 7 degrees of freedom. Estimation of the main effects and interactions of the three factors costs 7 degrees of freedom, so no degrees of freedom are available for the error variance. Blocking makes the problem even worse, since estimating b block effects costs $b - 1$ degrees of freedom. As a result, $b - 1$ of the treatment contrasts can no longer be distinguished from block contrasts, they are said to be *confounded* with blocks.
The only way this can work is if we do not need to fit the full model. Suppose it is known that the factor A does not interact with any of the other factors. This means that we do not need the AB, AC, and the ABC interaction terms in the model, since we may assume that they are zero.
How would we have to combine the means of the eight treatments to estimate the contrasts? For example, for the A contrast, we have to subtract the means of the four treatments where $A = 0$ (these are 000, 001, 010, 011, the first four rows in the table below) from the mean of the four treatments where $A = 1$. Similarly for the main effects of B and C.
For the interaction effects, we may simply take the product of the corresponding columns. So, for AB, we multiply the values in the A column with the values in the B column. Similarly for the other two-way interaction effects (you can fill them in), including the three-way interaction.

	Contrast						
Treatment	A	B	C	AB	AC	BC	ABC
000	-1	-1	-1	1			-1
001	-1	-1	1	1			1
010	-1	1	-1	-1			1
011	-1	1	1	-1			-1
100	1	-1	-1	-1			1
101	1	-1	1	-1			-1
110	1	1	-1	1			-1
111	1	1	1	1			1

For the allocation of treatments to blocks, we have to choose one contrast which is confounded with the blocks. Let us choose ABC for this, for example. As you see in the table above, there are four treatments with ABC contrast equal to 1, and four with ABC contrast equal to -1. Using this to define blocks yields the following design:

Block 1	000	011	101	110
Block 2	001	010	100	111

As a result, the block difference is confounded with the ABC contrast. In other words, we had to sacrifice one contrast anyway and did it such that we now can not distinguish between the block difference and the ABC contrast. Since we are willing to believe that ABC is zero anyway, this is not a problem. Since we also assume that AB and AC are zero, we even have two degrees of freedom to estimate σ^2. Similar techniques are applicable in more complex settings.

12.3 Fractional factorial designs

We again give only the main idea and refer to Dean, Voss, and Draguljić 2017, Ch. 15 (whose treatment we follow) or to Bailey 2008, Ch. 15 for more. Especially in industry, for example in early stages of product development, there is often a large list of potentially important factors. It would be useful to find out which factors are the most important, for example to design further experiments. One approach to do this are *screening experiments.*

Let us focus here on the simplest case to explain the principle. Suppose you have p potentially important factors, each with exactly two levels. A factorial experiment thus involves 2^p treatments. In the most simple *fractional factorial experiment*, one instead only observes a fraction of one half of the treatments, i. e. 2^{p-1} treatments. This requires only half as many experimental units and can thus substantially reduce time, costs and workload required. But how to choose which treatments are omitted, and what is the price of this strategy?

Again, this is possible in case we are willing to make some compromises regarding the contrasts that we estimate. Suppose we run only Block 2 in the experiment in Section 12.2. Then the former contrasts look as follows:

B here

AC and B the same, can not tell the main effect from B and interaction effect of AC, this is disavatige of fractinal factorial

advantige is reduce the parameter

assume intraction is not important

Treatment	A	B	C	AB	AC	BC	ABC
001	-1	-1	1	1	-1	-1	1
010	-1	1	-1	-1	1	-1	1
100	1	-1	-1	-1	-1	1	1
111	1	1	1	1	1	1	1

Comparing the columns shows that $A = BC$, $B = AC$, and $C = AB$. The ABC column does not define a contrast ($\sum c_j \neq 0$), instead the weights corresponds to the mean. So ABC is confounded with the mean. This design is called a 2^{3-1} fractional factorial design and one often writes $I = ABC$ and calls ABC the *defining contrast* for the design; this contrast can not be estimated. The contrasts which are equal to each other (e. g. B and AC) are said to be *aliased*; they cannot be separated from each other by design. So one has to accept either that the corresponding contrast is a mixture of two effects or to assume that for example the interaction terms are equal to zero.

12.4 Alternative analysis approaches

Three main problems are common with regard to the assumptions underlying linear (mixed) models: outliers, heteroskedasticity, or non-normality of the residuals.

Because sample sizes are often small in the context of designed experiments, the problems above can have severe consequences and seriously distort the conclusions. We showed some countermeasures for simple designs in Chapter 4.

For more complex designs, it is sometimes difficult to find good alternative analysis methods in case the model assumptions are not fulfilled.

For independent observations (no plot structure), the *Brunner-Munzel* approach offers a *nonparametric* two- and higher-way factorial ANOVA for which works under much weaker conditions than classical ANOVA. To my knowledge, it is not yet implemented in an `R` package (but in SAS) but one can do this by application of other modeling functions in `R` as well.

The *Brunner-Langer* models (see Brunner, Domhof, and Langer 2002) are nonparametric techniques for *longitudinal data* and offer a way of dealing with temporally correlated data that does not rely on normality assumptions. There are a family of models for different situations. These models are implemented in the `nparLD` package.

We further refer to Wilcox 2012 for robust methods (in principle, any robust linear (mixed) model estimation technique can be used) to deal with heteroskedasticity or outliers.

Finally, Basso et al. 2009 discuss a permutation approach for two-way ANOVA. The permutation approach requires very little assumptions (which are often satisfied due to the randomization in experiments), but it only provides p values, not effect sizes.

until now we know how to plan and how to conduct a experiment
before collect data,

13 Power and sample size calculations

Suppose we are planning an experiment and we know already how many treatments we use, we know about plot structure and also about the statistical model/test that we will use.

A short reminder about hypothesis tests

In the classical hypothesis test setting, we have a null hypothesis H_0, an alternative hypothesis H_1 and a hypothesis test to choose between the two hypotheses. In the setting of experiments, the null hypothesis is usually that some treatment has no effect (e. g. the new treatment is not better than placebo), and the alternative is that there is a treatment effect. The hypothesis test can make two types of errors: Type I denotes the probability of rejecting H_0 although it is true (false alarm), and Type II denotes the probability of not rejecting H_0 although it is false (false non-alarm). We focus on level α tests here, which have the property that they reject a true H_0 with a probability of at most α (the significance level). We let β denote the probability of a Type II error. The power $1 - \beta$ of the test is a lower bound for the probability of rejecting a false null hypothesis. Refer to the statistical inference notes for more details.

We fix the significance level α as well. Two related questions are then often asked:

1. How big is the power $1 - \beta$ of the test if we use n observations?

2. How many observations are required to achieve a given power with the test?

The first question is about the power, given the sample size, whereas the second question is about the required sample size to achieve a given power. If we can solve the first question for each n, we can then find the minimal n such that the power reaches some desired level.

Power or sample size computations are meant to be performed *before* the experiment.[27] They are extremely important for financial and ethical reasons, especially for research on living organisms. A too small sample will yield low power, so that an important effect may be missed. A too big sample uses unnecessarily many patients. (See also early stopping.)

As explained in the notes on statistical inference, the power of a particular test is a generally a function of the significance level, the sample size and the amount of disagreement between the null and the alternative hypothesis. Of course, it also depends

[27] Plugging in the actually observed values in your sample *after* conducting the study to try to find out how high the power was is called *post hoc* power computation and in general not a good idea. Seriously. Don't do this.

on the distribution of the data, and on any particular plot structure (e. g. strength of correlation inside a block).

> Before designing an experiment, it is advisable to consult a statistician to determine the statistical test, to compute the required sample size for the different groups such that an effect of a given size is found with a desired power, at a given significance level. Preliminary information on the group means, the common variance and any plot structure is required to carry out these computations.

To obtain this information, you could consult the literature or conduct a pilot study, or simply try plausible hypothetical scenarios.
Some scenarios can be looked up in the statistical literature. Some very basic settings are covered in the `pwr` package. For mixed models, some convenience packages exist. There also exists specialized commercial software for the required calculations for more complex designs. We show a very generally applicable approach here.

13.1 Monte Carlo approach

RCBD

For the sake of having a concrete example, we focus on a randomized complete block design (not replicated) here. Suppose we have three treatments and we have an idea about the treatment means, the error variance and the block variance. If we fix the significance level, how many blocks should we observe to achieve a power of, say, 80%? Here is the idea:

1. Simulate a random sample under the alternative hypothesis.
2. Calculate the p value of the test.
3. Perform 1. and 2. a sufficient number of times count the proportion of significant results. Use this as Monte Carlo approximation to the power.

Below, we show implement this in a very simple way. As a little extra, we compare the power of the Friedman test with the power from the marginal F test of the linear mixed model with random block effect. The argument `runs` is the number of simulated samples.

SIMULATE DATA SET, knowing that there is a treatment effect

```
> compare.sim <- function(n.block, n.treatments, means, sds, sd.block,
+                           runs, sig.level = 0.05) {
+     df <- cbind(block = paste0("B", rep(1:n.block, each = n.treatments)),
+                 treatment = paste0("T", rep(1:n.treatments)))
+     df <- data.frame(df)
+     p.val <- data.frame(matrix(NA, nrow = runs, ncol = 2))
+     names(p.val) <- c("Friedman.p.value", "Mixed.model.pvalue")
+     sig <- function(x) length(x[x < sig.level]) / length(x)
+     for (i in 1: dim(p.val)[1]) {
+         ## Simulate data
+         df$y <- rnorm(n = dim(df)[1],
                                treatment effect plus block effect
+                          mean = rep(rnorm(n = n.block, mean = 0, sd = sd.block),
+                                     each = n.treatments) +
+                                 rep(rep(means), times = n.block),
+                          sd = rep(rep(sds), times = n.block))
+         ## Calculate p values
+         p.val[i, ] <- c(friedman.test(y ~ treatment | block,
+                                       data = df)$p.value,
+                         anova(lme(y ~ treatment, random = ~ 1 | block,
+                                   data = df))$`p-value`[2])
+     }
+     return(apply(p.val, MARGIN = 2, sig))
+ }
```

question: how many block we need to get the power of data analyse more than 80%
simulation with different block number with given values (sd.block, mean.treatment,)

To apply this now, we set the random number seed to get reproducible results and then pass the arguments. The treatment effects are reasonably strong. We start with a low number of runs here to keep runtime low. For more precise results, the number of runs should be increased.

```
> set.seed(17)
                     4 blocks the power is 60%
> compare.sim(n.block = 4, n.treatments = 3, means = c(1, 2.7, 3.2),
+             sds = c(1, 1, 1), sd.block = 1.5, runs = 100)

#   Friedman.p.value Mixed.model.pvalue
#               0.60               0.59
```

60% reject the null hypothesis, which is given the ha is true, and h0 is rejected

This suggests that four blocks provide a power of roughly 60% regardless of the test. Let us now find the required number of blocks to achieve a power of 80% using the Friedman test. This is where expensive software can make a difference with clever search strategies. We simply search on a grid until we are near the solution and then increase the number of runs. This is computationally inefficient, but this does not matter as long as we do not have to run too many power calculations.

```
> for (bl in 4:10) {
+     cat(paste("blocks: ", bl, sep = ""))
+     print(compare.sim(n.block = bl, n.treatments = 3, means = c(1, 2.7, 3.2),
+             sds = c(1, 1, 1), sd.block = 1.5, runs = 100))
+ }
```

```
# blocks: 4  Friedman.p.value Mixed.model.pvalue
#               0.62               0.70
# blocks: 5  Friedman.p.value Mixed.model.pvalue
#               0.67               0.77
# blocks: 6  Friedman.p.value Mixed.model.pvalue
#               0.84               0.95
# blocks: 7  Friedman.p.value Mixed.model.pvalue
#               0.86               0.94
# blocks: 8  Friedman.p.value Mixed.model.pvalue
#               0.86               0.99
# blocks: 9  Friedman.p.value Mixed.model.pvalue
#               0.93               0.98
# blocks: 10  Friedman.p.value Mixed.model.pvalue
#               0.96               1.00
```

It seems that six blocks are required. But 100 runs are not nearly enough, so we should increase the number of runs now to validate the results:

```
> for (bl in 5:7) {
+     cat(paste("blocks: ", bl, sep = ""))
+     print(compare.sim(n.block = bl, n.treatments = 3, means = c(1, 2.7, 3.2),
+             sds = c(1, 1, 1), sd.block = 1.5, runs = 5000))
+ }
```

Increasing the number of runs shows that six blocks actually only provide a power of roughly 76%, while seven block yield a power of about 81%, so seven blocks should be used. In practice, often a set of plausible scenarios is run (for example, with different effect sizes) to get an idea of the robustness of the results.

For this particular design, analytical results are available in case the data come from a normal distribution. As soon as we want to allow more flexibility, the analytical approach will have difficulties, while the Monte Carlo approach may still be applied, and not only to this design, but to any design.

13.2 Analytical results

As long as the data come from a normal distribution, there is no need for power simulations, one can usually find a formula for the power and simply plug in the parameters (usually involving the non-central F distribution). It is even possible to calculate the required sample size such that the confidence interval for a given contrast (say, treatments versus control) has a specified length.

The solutions are specific to the design used and are found in the literature, for example in Dean, Voss, and Draguljić 2017. If you have a standard design, then it is a good idea to consult the literature for the required sample size. As the saying goes, coding for one week can save you half a day in the library ...

13.3 Efficiency

In some situations, several designs would be possible, so it is interesting to compare them in statistical terms, especially in terms of their *relative efficiency.* We do not give the details here, see e. g. Oehlert 2010, Ch. 13.2.3, Bailey 2008, Ch. 11.6 or Hinkelmann and Kempthorne 2008, Ch. 9.3. The relative efficiency of design D_1 to design D_2 roughly tells us how many times more observations we would need to get statistically similar results if we used design D_2 instead of D_1. Efficiency may be used to help plan future experiments.

Complete vs. incomplete block designs

We use Bailey 2008, Ex. 11.6 to illustrate.

Example 13.1. *Suppose you have 7 treatments and 21 plots are available. Now you can either run the experiment as a complete block design, with three blocks and error variance σ^2_{RCBD}, or as a balanced incomplete block design with seven blocks, three observations per block and variance σ^2_{BIBD} (such a BIBD exists, as we showed in Chapter 9.2.1).*
Suppose blocks are days, and if you hurry, you may run all seven treatments in the lab in one day. But you will have to hurry, and use the whole day, so that moisture and temperature will perhaps fluctuate more. On the other hand, you could run the experiment on seven days and only observe three treatments per day. Thus, you need to hurry less and the conditions stay more constant because you use less than the whole day for the experiment. Is it possible that the smaller variance within a block could compensate the smaller efficiency of the BIBD relative to the RCBD?
Let us ignore degrees of freedom adjustments. In the above BIBD, the variance of each estimator of a difference between two treatments is $\frac{6}{7}\sigma^2_{BIBD}$. In the above RCBD, the variance of each estimator of a difference between two treatments is $\frac{2}{3} \cdot \sigma^2_{RCBD}$. Thus, the incomplete block design is preferred over the complete block design if and only if

$$\sigma^2_{BIBD} < \frac{7}{9}\sigma^2_{RCBD} \,.$$

If making the blocks smaller sufficiently reduces the variability, an incomplete block design can outperform a complete block design with the same number of observations. The references cited above contain more general information for comparing other designs.

14 Planning experiments

This chapter is essentially a copy of Bailey 2008, Ch. 14.3 with some additions from Dean, Voss, and Draguljić 2017, Ch. 2.2. The aim is to highlight important aspects in planning an experiment. The classical reference is Fisher 1949, a modern classic Box, Hunter, and Hunter 2005. A wealth of `R` specific information may be found at `http://stat.ethz.ch/CRAN/web/views/ExperimentalDesign.html`.
A research plan consists of a series of experiments used to investigate different aspects of some random phenomenon. Often, this involves several research questions building on each other. Here, we focus on the case of a single experiment.

14.1 The protocol

The aim is now to give a protocol that describes the experiment. Such a protocol is usually written in collaboration between the scientist and the statistician. At least the following questions should be addressed. Many of its elements can later be reused for a scientific publication.

14.1.1 What is the purpose of the experiment?

It is important to be precise here. For example,

- Is it about *estimating* the effect of a new fertilizer on the yield, compared to the standard?
- Is it the aim to *test* if whether the temperature and the moisture effect on the consistence of chocolate interact?
- Is it the aim to *model* the effect of the level of an drug on the outcome, while controlling for gender and age?
- Is it a screening experiment?
- Is it a dose-response study?

14.1.2 What are the treatments?

Here, the treatments are described in all technical details. How much of what is given how and when? Recall that treatments may be structured, for example they could have a factorial structure. Or treatment levels that correspond to a numeric variable could be equispaced. Any control treatments (or placebo) should be mentioned here. How many treatments are there?

14.1.3 What are the methods?

This is a non-statistical part that describes exactly how the treatments are applied and all that happens until the measurements are taken. The aim is that other scientists can replicate the study with these details. Are there several persons that apply the treatments? How are they allocated to experimental units?

14.1.4 What are the experimental units?

An exact description of the experimental units. Is there any relevant history (previous year study on the same plots?) Are there any relationships between the experimental units, such as relevant spatial or temporal proximity; are there any blocking factors? How many experimental units are there?

14.1.5 What are the observational units?

Are observational units the same as the experimental units? Then it should be mentioned; if it is not the case, there will most likely be several observational units per experimental unit. How many are there, and how are they defined in detail? All information about plot structure belongs here.

14.1.6 What measurements are going to be recorded?

What is going to be recorded? When does it happen, and to what precision? What are the measurement units? It makes sense to prepare a data sheet for the field and for the computer already at this stage. Each observational unit has a row and each measurement has a column. Are there several people involved in the measurements? Who measures which observational units? Are covariates/nuisance factors measured? How exactly?

14.1.7 What is the design?

Here, the design (what experimental unit gets which treatment) is given; in case a standard design is used, it is sufficient to give its name and the role of any factors. Special designs need to be described explicitly.
The amount of replication should be justified here, this is often very poorly done. Any pilot studies, sample size simulations, related experiments or such should be mentioned here. Blocking should be mentioned again here; how were blocks and block sizes chosen? Any relevant practical restrictions on the application of the treatments should be mentioned here. In case special designs with confounding or aliasing are used, the precise choice should be explained.

14.1.8 What randomization was used?

What method of randomization was used? You should always keep a record of the seed used in drawing random numbers.

14.1.9 What is the plan of the experiment?

The design has abstract treatments $T_1, T_2, \ldots$, what actual treatments were randomized to what explicitly named experimental units? In a field experiment, what plot obtained which actual treatment? Do not forget the north arrow on a map.

14.1.10 What statistical analysis is planned?

This needs to be decided anyway in order to calculate the required sample size. The more strata the design has, the more carefully this should be planned. Are the important factors measured at the proper level (think about split plots)? Are there enough degrees of freedom at each level? Do blocking factors have random or fixed effects, and in case of several factors, are they nested or crossed?

It is a very simple and good strategy to simulate some data in the planning stage of the experiment (for example, when simulating the sample size), then a proof-of-concept data analysis of the simulated data should reveal any systematic problems.

Is it planned to transform the data before analysis (due to experience in similar experiments)? It is not a problem to simulate heteroskedastic data and use this from the beginning.

(Many things may happen during the experiment that necessitate a change in the final analysis. For example, missings may turn the design into an unbalanced design, outliers may affect the results, and so on.)

References

R. A. Bailey (2008). *Design of Comparative Experiments.* Cambridge University Press.

D. Basso, F. Pesarin, L. Salmaso, and A. Solari (2009). *Permutation Tests for Stochastic Ordering and ANOVA.* Springer.

G. Beall (1942). The Transformation of Data from Entomological Field Experiments so that the Analysis of Variance Becomes Applicable. *Biometrika* 32, 243–262.

G. E. P. Box, J. S. Hunter, and W. G. Hunter (2005). *Statistics for Experimenters. Design, Innovation and Discovery.* 2nd ed. Wiley.

E. Brunner, S. Domhof, and F. Langer (2002). *Nonparametric Analysis of Longitudinal Data in Factorial Experiments.* Wiley.

R. A. Cribbie, R. R. Wilcox, C. Bewell, and H. J. Keselman (2007). Tests for Treatment Group Equality When Data are Nonnormal and Heteroscedastic. *Journal of Modern Applied Statistical Methods* 6, 117–132.

A. Dean, D. Voss, and D. Draguljić (2017). *Design and Analysis of Experiments.* 2nd ed. Springer.

R. A. Fisher (1949). *The Design of Experiments.* 5th ed. Oliver and Boyd.

J. Fox and S. Weisberg (2011). *An R Companion to Applied Regression.* 2nd ed. Sage.

K. Hinkelmann and O. Kempthorne (2008). *Design and Analysis of Experiments.* 2nd ed. Vol. 1. Wiley.

M. Hollander, D. A. Wolfe, and E. Chicken (2014). *Nonparametric Statistical Methods.* 3rd ed. Wiley.

G. A. Milliken and D. E. Johnson (2009). *Analysis of Messy Data.* 2nd ed. CRC Press.

G. W. Oehlert (2010). *A first course in design and analysis of experiments.*

J. Pinheiro and D. Bates (2000). *Mixed-Effects Models in S and S-PLUS.* Springer.

D. J. Saville and G. R. Wood (1991). *Statistical Methods: The Geometric Approach.* Springer.

W. Venables (1998). “Exegeses on Linear Models”. In: *S-Plus User’s Conference, Washington DC.*

W. Venables and B. Ripley (2002). *Modern Applied Statistics with S.* 4th ed. Springer.

R. R. Wilcox (2010). *Fundamentals of Modern Statistical Methods.* 2nd ed. Springer.

– (2012). *Introduction to Robust Estimation & Hypothesis Testing.* 3rd ed. Elsevier.

A Data set overview

Data set	Place	Topic
`InsectSprays`	Ch. 2	completely randomized design
`InsectSprays`	Ch. 6	randomized complete block design
`immer`	Ch. 6	Friedman test
`turnip`	Ch. 7	factorial designs
`detergent`	Ch. 9	incomplete designs
`Oats`	Ch. 10	split plot desings
`gasel`	Ch. 11	using covariates
`morley`	Graded Set 1	One-way ANOVA
`metal`	Graded Set 2	randomized complete block design
`catalyst`	Graded Set 3	balanced incomplete block design
`soybean`	Local Session 3	factorial designs
`paper`	Local Session 4	split plot designs
`warpbreaks`	Case study	factorial designs, transformations, post-hoc