Design and Analysis of Experiments

Lecture notes

Part II

Christoph Kopp

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

November 22, 2018

Contents

5 Blocking **42**
5.1 Types of blocks . . . 42
5.2 Complete block designs . . . 43
5.3 Fixed block effect models . . . 44
5.4 Random block effect models . . . 45

6 Fitting and interpreting mixed effects models **46**
6.1 Data structure . . . 46
6.2 Fitting the model . . . 46
6.3 Extracting model results . . . 48
6.4 Model diagnostics . . . 51
6.5 Confidence intervals and a hypothesis test . . . 53
6.6 Treatment comparisons . . . 54
6.7 Ignoring blocks . . . 56
6.8 The intraclass correlation coefficient . . . 57
6.9 The coefficient of determination . . . 57
6.10 An example: the `immer` data and the Friedman test . . . 58
6.11 Interactions of block effects and treatment effects . . . 61

7 Designs with factorial treatment structure **65**
7.1 Introduction . . . 65
7.2 Visualization . . . 66
7.3 A model for factorial two-way ANOVA with replications . . . 68
7.4 F tests . . . 72
7.5 Tests on individual coefficients . . . 74
7.6 Model diagnostics and related topics . . . 75
7.7 Fitted values and confidence intervals . . . 79
7.8 Post hoc tests . . . 80
7.9 Unbalanced designs . . . 81
7.10 Accounting for blocks and for heteroskedasticity . . . 82

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

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.

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` $\times$ `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

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.
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
# 1    10     A     1
# 2     7     A     1
# 3    20     A     2
# 4    14     A     2
# 5    14     A     3
# 6    12     A     3
```

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.

```
> 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.
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
#   152.229 169.746 -68.1145
#
# Random effects:
#  Formula: ~1 | block
#         (Intercept) Residual
# StdDev:    0.254084 0.579766
#
# Fixed effects: sqrt(count) ~ spray
#                Value Std.Error DF   t-value p-value
# (Intercept)  3.76068  0.196902 61  19.09923  0.0000
# sprayB       0.11595  0.236688 61   0.48990  0.6260
# sprayC      -2.51582  0.236688 61 -10.62925  0.0000
# sprayD      -1.59632  0.236688 61  -6.74441  0.0000
# sprayE      -1.95122  0.236688 61  -8.24382  0.0000
# sprayF       0.25794  0.236688 61   1.08978  0.2801
#  Correlation:
#        (Intr) sprayB sprayC sprayD sprayE
# sprayB -0.601
# sprayC -0.601  0.500
# sprayD -0.601  0.500  0.500
# sprayE -0.601  0.500  0.500  0.500
# sprayF -0.601  0.500  0.500  0.500  0.500
#
# Standardized Within-Group Residuals:
#       Min        Q1       Med        Q3       Max
# -2.481582 -0.521393 -0.147134  0.623448  1.974996
#
# Number of Observations: 72
# Number of Groups: 6
```

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).

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

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

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
#    3.760678    0.115953   -2.515822   -1.596325   -1.951217    0.257939
```

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  -0.3050331
# 2   0.2829657
# 3  -0.0164562
```

```
# 4  -0.1098823
# 5   0.1938799
# 6  -0.0454740
```

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  -0.4373813 -0.3050331 1.43388
# 2   0.4057392  0.2829657 1.43388
# 3  -0.0235963 -0.0164562 1.43388
# 4  -0.1575582 -0.1098823 1.43388
# 5   0.2780008  0.1938799 1.43388
# 6  -0.0652043 -0.0454740 1.43388
```

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.

fitted values

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
# 3.45565 3.45565 4.04364 4.04364 3.74422 3.74422

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

#       1       1       2       2       3       3
# 3.76068 3.76068 3.76068 3.76068 3.76068 3.76068
```

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.

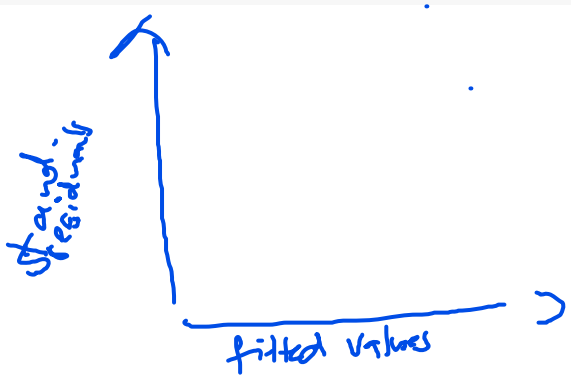
```
> head(resid(ins.lme))

#           1           1           2           2           3           3
# -0.29336766 -0.80989401  0.42849188 -0.30198669 -0.00256479 -0.28012057
```

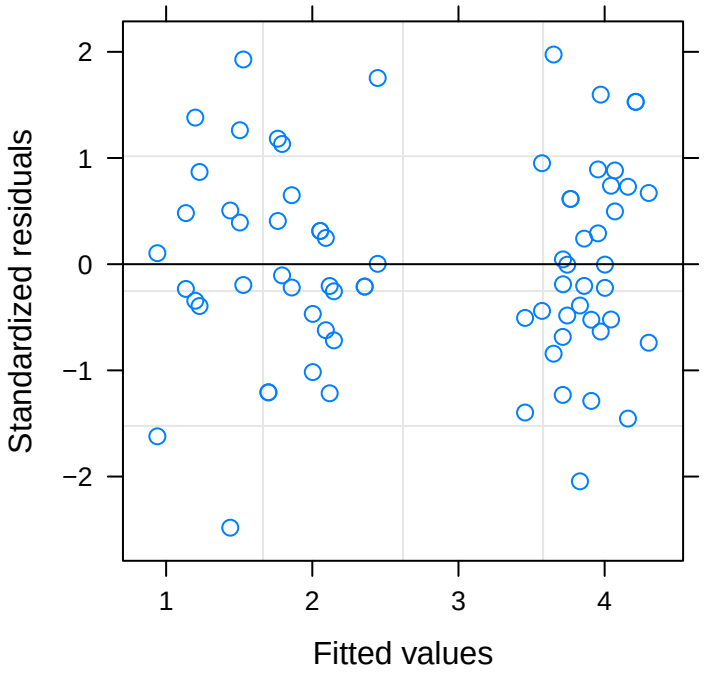
6.4 Model diagnostics

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 = 0.9898, p-value = 0.835
```

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)  3.366948  3.760678  4.154409
# sprayB      -0.357335  0.115953  0.589241
# sprayC      -2.989110 -2.515822 -2.042534
# sprayD      -2.069612 -1.596325 -1.123037
# sprayE      -2.424505 -1.951217 -1.477930
# sprayF      -0.215349  0.257939  0.731227
# attr(,"label")
# [1] "Fixed effects:"
#
#  Random Effects:
#   Level: block
#                    lower     est.    upper
# sd((Intercept)) 0.104136 0.254084 0.619946
#
#  Within-group standard error:
#    lower     est.    upper
# 0.485502 0.579766 0.692332
```

Given the small number of blocks, it is not surprising that the estimate for the block standard deviation is extremely 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    61 364.780  <.0001
# spray           5    61  52.621  <.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

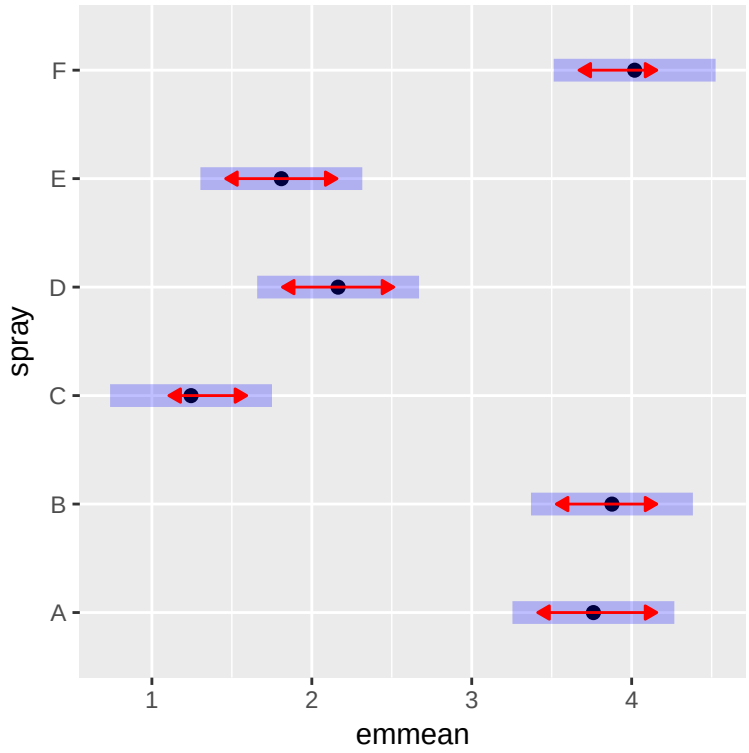
A very popular way of comparing treatments (especially in more complex settings) is to use what used to be called *least squares means* for decades and what is currently being renamed into *estimated marginal means*.
The idea is to define a *reference grid* of interesting values (for factors, these are 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     3.76068 0.196902  5 3.254525  4.26683
#  B     3.87663 0.196902  5 3.370478  4.38278
#  C     1.24486 0.196902  5 0.738704  1.75101
#  D     2.16435 0.196902  5 1.658201  2.67051
#  E     1.80946 0.196902  5 1.303308  2.31561
#  F     4.01862 0.196902  5 3.512464  4.52477
#
# 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    -0.115953 0.236688 61  -0.490  0.9964
#  A - C     2.515822 0.236688 61  10.629  <.0001
#  A - D     1.596325 0.236688 61   6.744  <.0001
#  A - E     1.951217 0.236688 61   8.244  <.0001
#  A - F    -0.257939 0.236688 61  -1.090  0.8836
#  B - C     2.631775 0.236688 61  11.119  <.0001
#  B - D     1.712278 0.236688 61   7.234  <.0001
#  B - E     2.067170 0.236688 61   8.734  <.0001
#  B - F    -0.141986 0.236688 61  -0.600  0.9907
#  C - D    -0.919497 0.236688 61  -3.885  0.0033
#  C - E    -0.564604 0.236688 61  -2.385  0.1776
#  C - F    -2.773760 0.236688 61 -11.719  <.0001
#  D - E     0.354893 0.236688 61   1.499  0.6659
#  D - F    -1.854263 0.236688 61  -7.834  <.0001
#  E - F    -2.209156 0.236688 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] 0.628345

> 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*

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

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

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

# block = pdLogChol(1)
#             Variance  StdDev
# (Intercept) 0.0645584 0.254084
# Residual    0.3361286 0.579766

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

# [1] 0.161119
```

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.

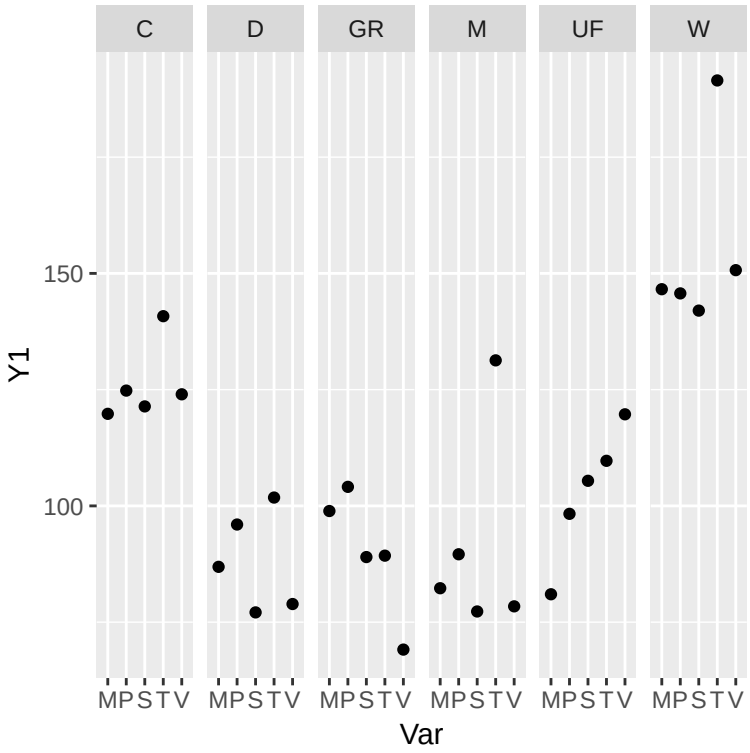
```
> library(MuMIn)
> r.squaredGLMM(ins.lme)
```

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        5   2757   689.2   0.817  0.526
# Residuals 25  21088   843.5

> sigma(lm.immer)

# [1] 29.0431
```

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  17830    3566  21.892 1.75e-07 ***
# Var        4   2757     689   4.231   0.0121 *
# Residuals 20   3258     163
# ---
# 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.[1]

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.

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

# Linear mixed-effects model fit by REML
#   Data: immer
#   Log-restricted-likelihood: -111.331
#   Fixed: Y1 ~ Var
# (Intercept)        VarP        VarS        VarT        VarV
#  102.583333    7.166667   -0.550000   24.816667    0.883333
#
# Random effects:
#  Formula: ~1 | Loc
#         (Intercept) Residual
# StdDev:     26.0886  12.7627
#
# Number of Observations: 30
# Number of Groups: 6

> anova(immer.lme)

#             numDF denDF  F-value p-value
# (Intercept)     1    20 100.0388  <.0001
# Var             4    20   4.2309  0.0121
```

We find a significant location effect. The standard deviation of the random intercepts is even bigger than the residual standard deviation, because the blocks are so different.

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

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 $\times$ 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
```

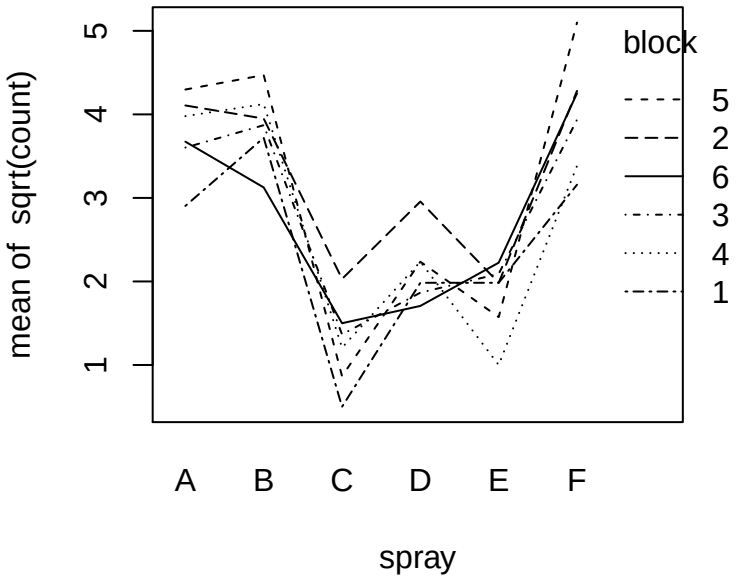
We find a significant variety effect. An extension of the Friedman test that handles missing values (making some assumptions) is known as Skillings-Mack test (see Hollander, Wolfe, and Chicken 2014) and implemented in several `R` packages.

6.11 Interactions of block effects and treatment effects

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

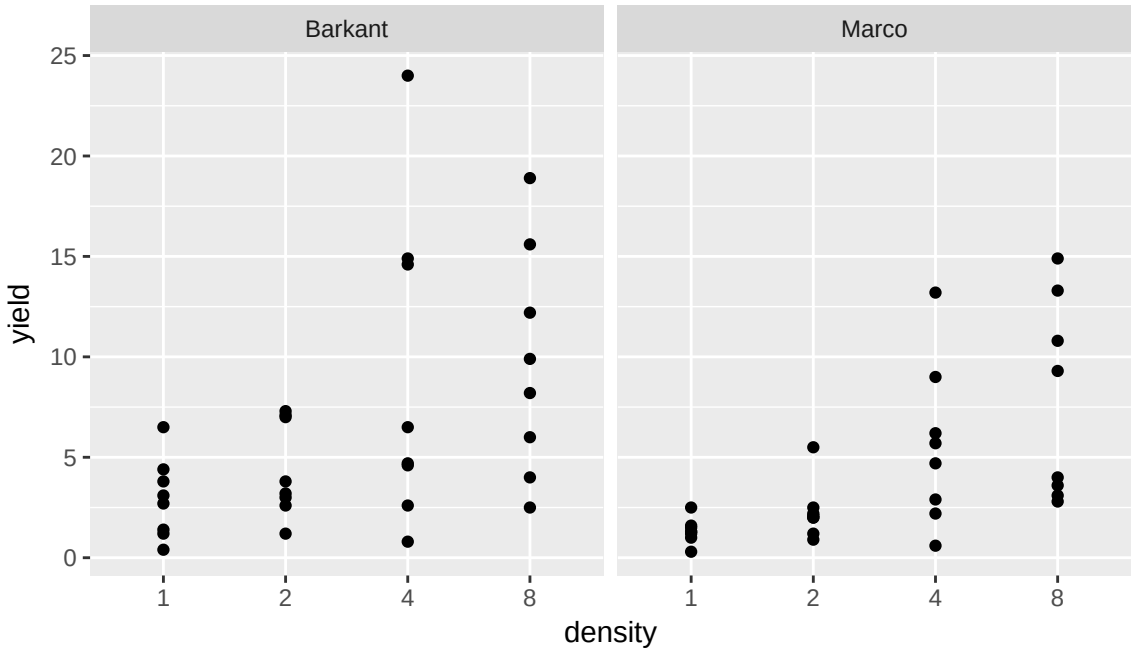
7.1 Introduction

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.

```
> 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:

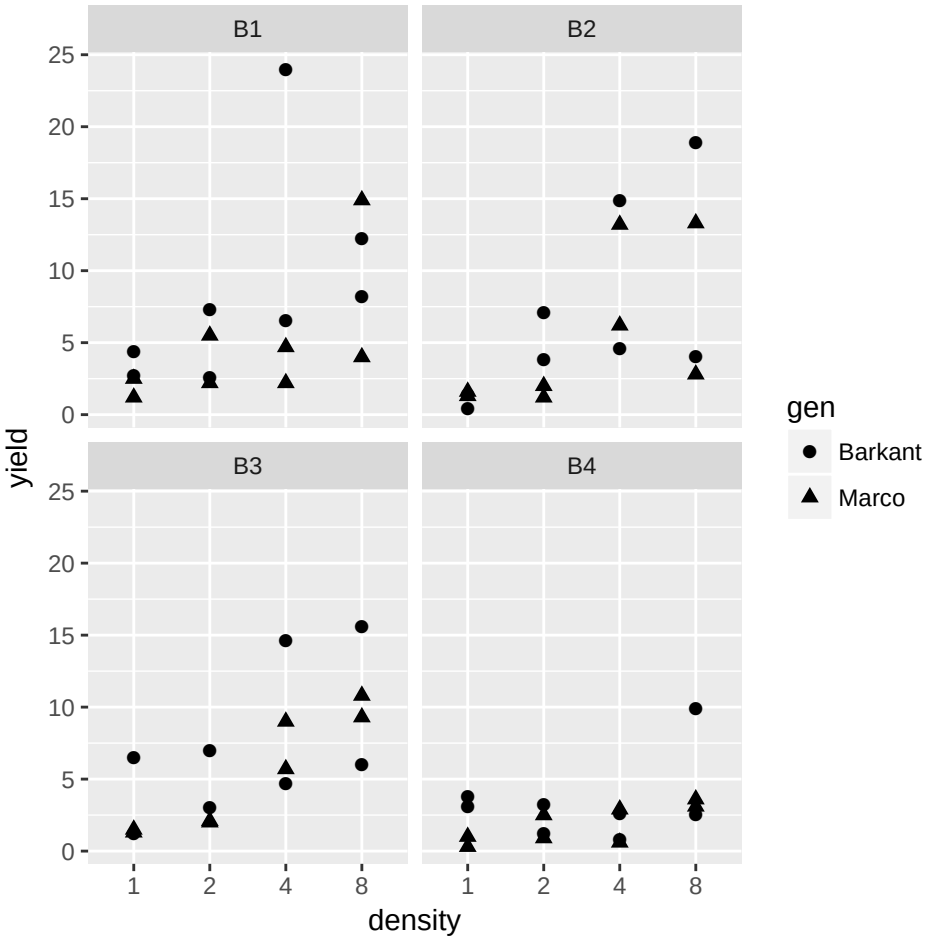
1. Is there a planting density effect? It seems so when inspecting the plot and the means: the yield seems to increase with the planting density.
2. Is there a genotype effect? The average Barkant values seem to be higher than the average Marco values.
3. Does the planting density effect (if there is 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))
```



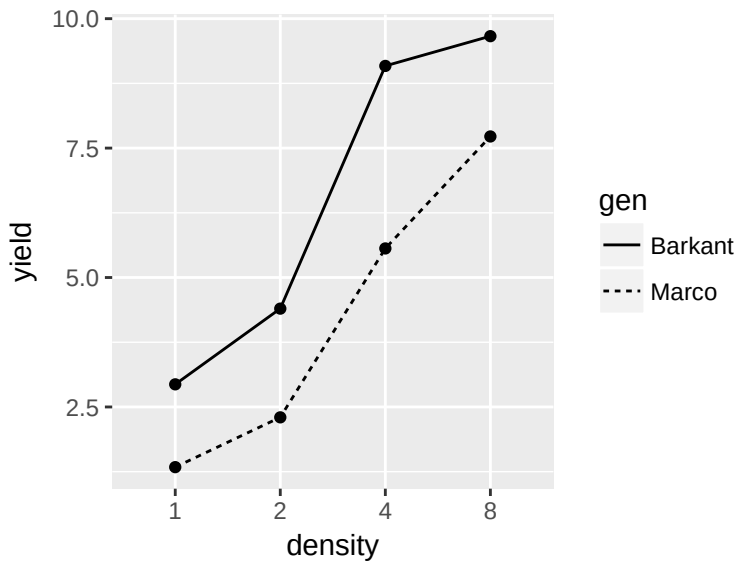
Block differences are clearly visible (Block 4 is suspicious) and an overall increase of the yield (and the variance) with the planting density. The Barkant variety seems to have produced a slightly higher average yield than Marco.

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`[2]. 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")
```

[2]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

Consider two factors X with levels $1, 2, \ldots, a$ and Z with levels $1, 2, \ldots, b$ 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 $k = 1, 2, \ldots, n_{ij}$) in the group with factor X being equal to i and factor Z being equal to j, where $1 \leq i \leq a$ and $1 \leq j \leq b$.
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 $Y_{ij1}, Y_{ij2}, \ldots, Y_{ijn_{ij}}$ for each $i = 1, \ldots, a$, $j = 1, \ldots, b$. 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

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_{ij} + \varepsilon_{ijk} \tag{1}$$

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.[3]

[3]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 $i = 1, \ldots, a$ and $\gamma_{1j} = 0$ for $j = 1, \ldots, b$. 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 (1), 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 (1) 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 (1) 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.[4]

> 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.[5]
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

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

[5] 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.

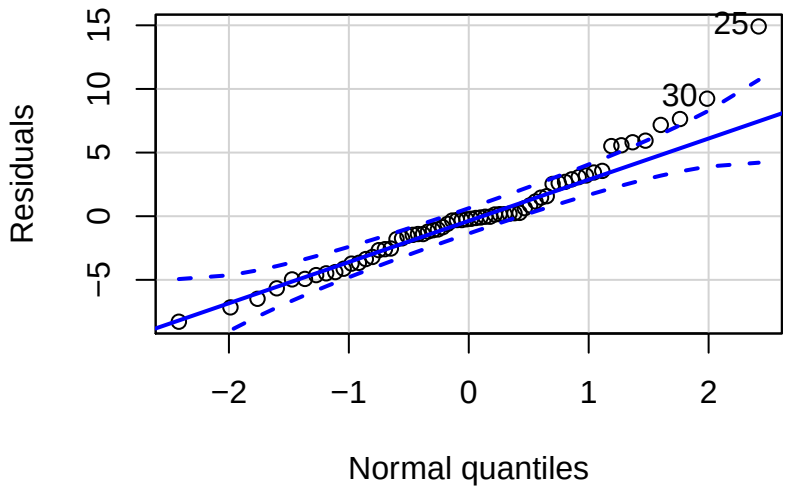
7.6 Model diagnostics and related topics

> 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.[6] 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 $p = 0.9696$.

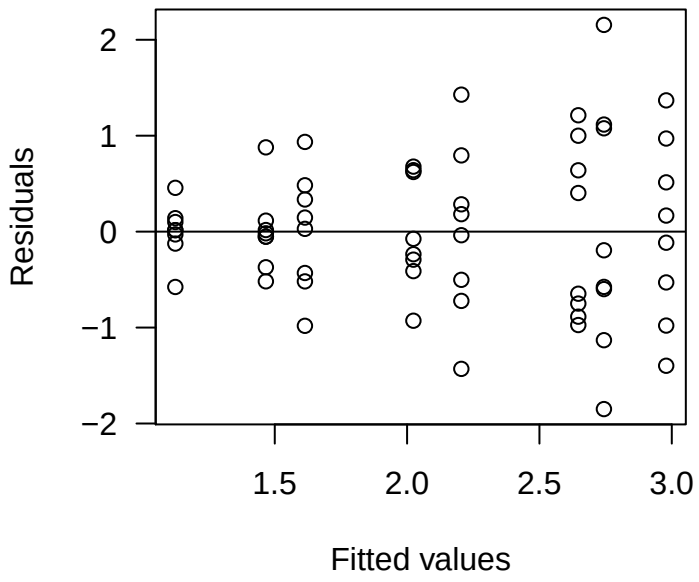
> All subsequent analyses are performed with the square root of the yield as dependent variable and then transformed back after analysis by squaring.

7.6.4 Residual plots

We plot the residuals vs. the fitted values.

[6] 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.[7]
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
```

[7] 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.

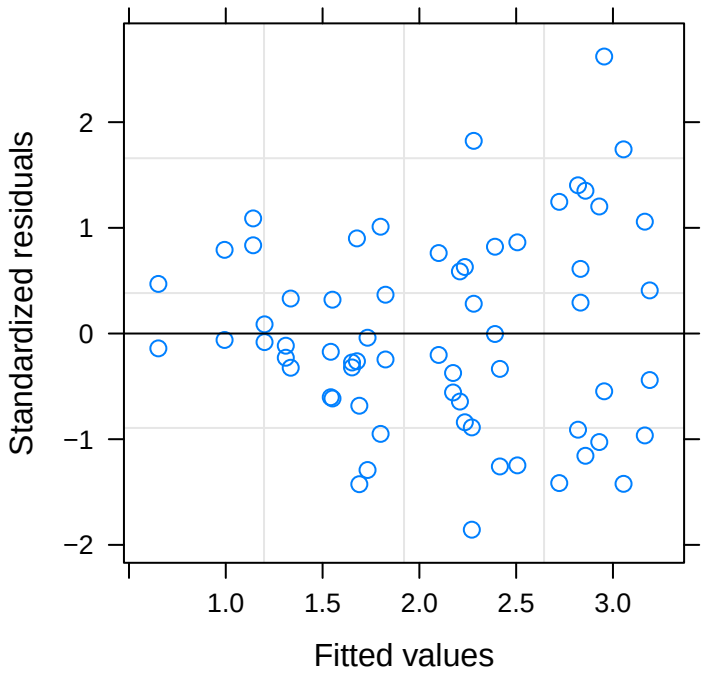
7.10 Accounting for blocks and for heteroskedasticity

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

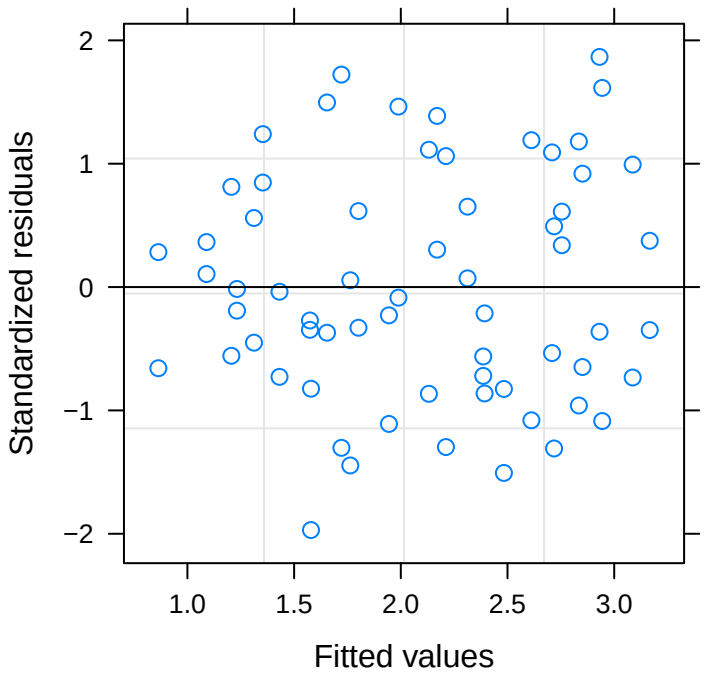
Normality looks perfect, but the variances still increase with the fitted values.

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.