Applied Analysis of Variance and Experimental Design (ANOVA)

Michael van Huffel

October 11, 2023

Version: October 11, 2023
This summary has been written based on the Lecture of ANOVA (401-0625-01L) by Prof. Dr. L. Meier (Autumn 22). There is no guarantee for completeness and/or correctness regarding the content of this summary. Use it at your own discretion

0 Contents

1 Learning from Data

We are in the (abstract) situation where we have a "system" or a "process" with many input variables (predictors) and an output (response). The predictor could be a fertilizer and the output the biomass of a plant. We would like to find a cause-effect relationships. With observational data, we can typically just make a statement about an association between two variables. A potential danger is the existence of confounding variables / confounders. A confounder is a common cause for two variables (e.g. turbulent weather cause the seatbelt sign to switch on and the plane to shake, but the seatbelt sign is not a cause of the shaking plane).

In an observational study, we only observe subjects / objects in an existing (uncontrolled) situation. There are cross-sectional studies (e.g. a "snapshot" of the population at a given time-point), cohort studies which are prospective (what will happen if...?, e.g. determining the risk of exposed vs. non-exposed smokers) and case-control studies which are retrospective (why did it develop this way?, e.g. comparison of habits). In an experimental study, we can observe the subjects / objects in a controlled setting. Before designing an experimental study, we must have a focused and precise research question that we want to answer with experimental data. The study then consists of:

- The different interventions which we perform on the system (the different treatments)
- Experimental units: The "things" to which we apply the treatments
- A method that assigns treatments to experimental units (typically randomization) Response(s): The output that we measure

We distinguish between the following types of predictors:

- Predictors that are of primary interest and that can (ideally) be varied according to our "wishes"
- Predictors that are systematically recorded such that potential effects can later be eliminated in our calculations ("controlling for...")
- Predictors that can be kept constant and whose effects can therefore be eliminated (e.g. using always the same measurement device)
- Predictors that we can neither record nor keep constant (e.g. some special soil properties)

Randomization (the random allocation of experimental units to the different treatments) ensures that the only systematic difference between the different treatment "groups" is the treatment and protects from confounders. Typically, the order (time), locations and instruments should also be randomized. When we already know that some experimental units are more alike than others before doing the experiment, we do a randomization "within" homogeneous blocks which is called blocking. Blocking (typically) increases precision of an experiment.

An experimental unit is defined as the "thing" to which we apply the treatments by randomization and "should be able to receive any treatment independently of the other units". A measurement unit is the unit on which the response is being measured. They don't need to be the same. E.g., if we randomize different food supplies to cages of animals, the experimental unit is the cage. However, the measurement unit will be given by an individual animal of the cage. Typically, values of measurement units are aggregated such that we get one value per experimental unit. The experimental units should ideally be a random sample from the population of interest.

The response should be chosen such that it reflects useful information about the process under study. If not directly measurable, a surrogate response is used (e.g. a cell count for disease progression).

Different experimental units will always give different responses to the same treatment. We should design our experiment such that we get an idea of this so-called experimental error, e.g. with multiple experimental units receiving the same treatment. When we do multiple measurements on the same experimental unit, we call them pseudoreplicates (which causes statistical tests on the data to be invalid).

Blinding means that the evaluators don't know which treatment is given to which experimental unit. With double-blinding, neither the evaluators nor the patients know the assignment (this protects us from bias due to expectations).

A control treatment is a standard treatment used as a baseline. A placebo is a "null treatment" for situations where the act of applying a treatment potentially has an effect. Categorical predictors are also called factors. We distinguish between unordered (or nominal; e.g. the eye color) and ordered (or ordinal; e.g. an income class) factors.

2 Completely Randomized Designs

2.1 One-Way Analysis of Variance:

Model: One-Way ANOVA

We have

- $g \geq 2$ treatments
- N experimental units that are assigned randomly to the g different treatment groups having n_i observations each.

Let Y_{ij} be the j th observation in treatment group i (where $i = 1, \ldots, g$ and $j = 1, \ldots, n_i$). In the cell means model, each treatment is allowed to have its own expected value and we assume:

$$Y_{ij} \sim N\left(\mu_i, \sigma^2\right), \text{ independent}$$

(where μ_i is the expected value of treatment group i and σ^2 the variance which is equal for all groups). This can be rewritten as:

$$Y_{ij} = \mu_i + \epsilon_{ij}, \quad \mu_i = \mu + \alpha_i, \quad Y_{ij} = \mu + \alpha_i + \epsilon_{ij}$$

with ϵ_{ij} i.i.d. $\sim N\left(0, \sigma^2\right)$

Source	**df**	**SS**	**Mean squares**	**F-ratio**
Treatment	$g-1$	SS_{Trt}	$MS_{\text{Trt}} = \frac{SS_{\text{Trt}}}{g-1}$	$\frac{MS_{\text{Trt}}}{MS_E}$
Error	$N-g$	SS_E	$MS_E = \frac{SS_E}{N-g}$	

α_i is also called the i th treatment effect. This model isn't identifiable anymore because $g+1$ parameters $(\mu, \alpha_1, \ldots, \alpha_g)$ for g different means $(\mu_1, \ldots, \mu_g)$. Only $g-1$ elements of the treatment effects are allowed to vary freely, it has $g-1$ degrees of freedom (df). Therefore, side constraints are introduced, which are set in R with:

```
options(contrasts = c("contr.sum", "contr.poly"))
```

- **Weighted sum-to-zero**, Side-Constraint: $\sum_{i=1}^{g} n_i \alpha_i = 0$, Interpretation of $\mu : \mu = \frac{1}{N} \sum_{i=1}^{g} n_i \mu_i$, R: -
- **Sum-to-zero**, Side-Constraint: $\sum_{i=1}^{g} \alpha_i = 0$, Interpretation of $\mu : \mu = \frac{1}{g} \sum_{i=1}^{g} \mu_i, \mathbf{R} :$ *contr.sum*
- **Reference group**, Side-Constraint: $\alpha_1 = 0$, Interpretation of $\mu : \mu = \mu_1, \mathbf{R} :$ *contr.treatment* where the first level in the output of levels is the reference group, can be changed with *relevel*, e.g.:

```
d$col<- relevel (d$col, ref = "M")
```

Note: The interpretation of the parameters $\mu, \alpha_1, \ldots, \alpha_{g-1}$ strongly depends on the parametrization that is being used.

The model is fitted using the least squares criterion, i.e.:

$$\widehat{\mu}, \widehat{\alpha}_i = \operatorname{argmin}_{\mu, \alpha_i} \sum_{i=1}^{g} \sum_{j=1}^{n_i} \left(y_{ij} - \mu - \alpha_i\right)^2$$
$$\widehat{\mu}_i = \operatorname{argmin}_{\mu_i} \sum_{i=1}^{g} \sum_{j=1}^{n_i} \left(y_{ij} - \mu_i\right)^2$$

Which gives us $\widehat{\mu}_i = \widehat{\mu} + \widehat{\alpha}_i = \bar{y}_{i\cdot}$. The error variance (of residuals) is estimated by the mean squared error:

$$\widehat{\sigma}^2 = MS_E = \frac{1}{N-g} SS_E \quad SS_E = \sum_{i=1}^{g} \sum_{j=1}^{n_i} \left(y_{ij} - \widehat{\mu}_i\right)^2$$

Or:

$$MS_E = \frac{1}{N-g} \sum_{i=1}^{g} \left(n_i - 1\right) s_i^2 \quad s_i^2 = \frac{1}{n_i - 1} \sum_{j=1}^{n_i} \left(y_{ij} - \widehat{\mu}_i\right)^2$$

Where s_i^2 is the empirical variance in treatment group i. The denominator $N-g$ ensures that $\widehat{\sigma}^2$ is an unbiased estimator (the error estimate has $N-g$ degrees of freedom). In R,

```
fit <- aov(response ~ predictor, data = data)
```

is used for fitting the model. The coefficients can be viewed with coef(fit), e.g.:

```
##  (Intercept)    grouptrt1    grouptrt2
##  5.032          -0.371       0.494
```

(Intercept) contains $\widehat{\mu} = 5.032$ (interpretation dependent on side constraints, without constraints expected value of control group). *grouptrt1* is $\widehat{\alpha}_2 = -0.371$ and *grouptrt2* $\widehat{\alpha}_3 = 0.494$. The function *dummy.coef(fit)* shows all coefficients, e.g.:

```
##  Full coefficients are
##
##  (Intercept):      5.032
##  group:            ctrl1   trt1    trt2
##                    0.000   -0.371  0.494
```

We can get the estimated cell means $\widehat{\mu}_i$ with *predict(fit, newdata =data.frame (group =c("ctrl", "trt1", "trt2")))*:

```
##      1       2       3
##      5.032   4.661   5.526
```

The output with *contr.sum* looks as follows:

```
options(contrasts = c("contr.sum", "contr.poly"))
fit2 <- aov(weight ~ group, data = PlantGrowth)
coef(fit2)
```

```
##  (Intercept)    group1    group2
##  5.073          -0.041    0.412
```

Now, *(Intercept)* is the global mean, *group1* the difference of the first (control) group and *group2* the difference of the second group. With *dummy.coef*, the full picture can be retrieved again:

```
##  Full coefficients are
##
##  (Intercept):      5.073
##  group:            ctrl1    trt1     trt2
##                    -0.041   -0.412   0.453
```

2.2 Tests:

Our null hypothesis is that all groups share the same mean, i.e.:

$$Y_{ij} = \mu + \epsilon_{ij}, \epsilon_{ij} \text{ i.i.d. } \sim N\left(0, \sigma^2\right).$$
$$H_0 : \mu_1 = \ldots = \mu_g$$

This is the single mean model and is a special case of the cell means model with $\alpha_1 = \ldots = \alpha_g = 0$. The alternative is therefore:

$$H_A : \mu_k \neq \mu_l \text{ for at least one pair } k \neq l.$$

The total variation of the response around the overall mean can be decomposed into variation "between groups" and variation "within groups":

$$\underbrace{\sum_{i=1}^{g} \sum_{j=1}^{n_i} \left(y_{ij} - \bar{y}_{\cdot\cdot}\right)^2}_{SS_T} = \underbrace{\sum_{i=1}^{g} \sum_{j=1}^{n_i} \left(\bar{y}_{i\cdot} - \bar{y}_{\cdot\cdot}\right)^2}_{SS_{\text{Trt}}} + \underbrace{\sum_{i=1}^{g} \sum_{j=1}^{n_i} \left(y_{ij} - \bar{y}_{i\cdot}\right)^2}_{SS_E}$$

Where SS_T is the total sum of squares, SS_{Trt} the treatment sum of squares (between groups) and SS_E the error sum of squares (within groups). SS_{Trt} can also be interpreted as the reduction in residual sum of squares when comparing the cell means with the single mean model. This information can be summarized in the ANOVA table (see Appendix). If all the groups share the same (theoretical) mean, we expect the treatment sum of squares to be small. The idea is now to compare the variation between groups with the variation within groups. Under H_0, it holds that:

$$F = \frac{MS_{\text{Trt}}}{MS_E} \sim F_{g-1, N-g}$$

Where:

$$MS_{\text{Trt}} = \frac{SS_{Trt}}{g-1}$$

Under H_0, MS_{Trt} is also an estimator for σ^2 and therefore $F = \frac{MS_{Trt}}{MS_E} \approx 1$.

We reject the null hypothesis if the observed F value lies in a "extreme" region of the corresponding distribution, more precise we reject H_0 in favor of H_A if F is larger than the 95% quantile. The F-test is a omnibus test because it compares all group means simultaneously. Increasing the denominator degrees of freedom will decrease the corresponding quantile (which gives more power). In R, *summary(fit)* gives the ANOVA table and p-value. As the global test can also be interpreted as a test for comparing two different models, namely the cell means and the single means model, there's also another approach. *anova* can be used to compare the two models:

```
## Fit single mean model (1 means global mean):
fit.single <- aov(weight ~ 1, data = PlantGrowth)
## Compare with cell means model:
anova(fit.single, fit)
```

To perform statistical inference for the individual α_i's, *summary.lm(fit)* (for tests; retrieves all the parameters including standard errors) and *confint(fit)* (for confidence intervals) can be used. Interpretation depends on the side-constraint, an example output of *summary.lm(fit)* looks like this:

```
Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)           3.3600    0.1965   17.10 1.39e-07
treatmentCommercial   4.1200    0.2779   14.82 4.22e-07
```

2.3 Checking Model Assumptions:

Statistical inference (p-values, confidence intervals, . . .) is only valid if the model assumptions are fulfilled. So far, this means

- are the errors independent?
- are the errors normally distributed?
- is the error variance constant?
- do the errors have mean zero?

The errors ϵ_{ij} can't be observed, but the residuals r_{ij} can be used as estimates:

$$r_{ij} = y_{ij} - \widehat{\mu}_i.$$

In a QQ-plot, the empirical quantiles are plotted against the theoretical quantiles (of a standard normal distribution). We should more or less see a straight line if the distribution assumption is correct. This is done in R with *plot(fit, which = 2)*. If the QQ-plot suggests non-normality, we can try to use a transformation of the response to accommodate this problem.

The Tukey-Anscombe plot plots the residuals r_{ij} vs. the fitted values $\widehat{\mu}_i$. It allows us to check whether the residuals have constant variance and whether the residuals have mean zero. For the one-way ANOVA situation we could also read off the same information from the plot of the data itself and the residuals always have mean zero (per group). In R, the plot is generated by *plot(fit, which = 1).*

2.4 Transformations Affect Interpretation:

Whenever we transform the response we implicitly also change the interpretation of the model parameters. Therefore, while it is conceptually attractive to model the problem on an appropriate scale of the response, this typically has the side effect of making interpretation (much) more difficult. For example, if we use the logarithm,

$$\log\left(Y_{ij}\right)=\mu+\alpha_i+\epsilon_{ij}$$

all the α_i 's (and their estimates) have to be interpreted on the log-scale. For example, if we us contr.treatment and we have $\widehat{\alpha}_2=1.5$. This means: on the log-scale we estimate that the average value of group 2 is 1.5 larger than the average value of group 1 (additive shift). What about the original scale? We know that $\mathbb{E}\left[\log\left(Y_{ij}\right)\right]=\mu+\alpha_i$, but the expected value on the original scale does (in general) not directly follow the transformation, i.e. $\mathbb{E}\left[Y_{ij}\right]\neq e^{\mu+\alpha_i}$. However, we can make a statement about the median. On the log-scale the median is equal to the mean (because we have a symmetric distribution around $\mu+\alpha_i$) Hence,

$$\operatorname{median}\left(\log\left(Y_{ij}\right)\right)=\mu+\alpha_i$$

In contrast to the mean, any quantile directly transforms with a strictly monotone increasing function. As the median is nothing else than the 50%-quantile, we have

$$\operatorname{median}\left(Y_{ij}\right)=e^{\mu+\alpha_i}$$

Similarly, for the ratio

$$\frac{\operatorname{median}\left(Y_{2j}\right)}{\operatorname{median}\left(Y_{1j}\right)}=\frac{e^{\mu+\alpha_2}}{e^{\mu}}=e^{\alpha_2}$$

Hence, we can make a statement that on the original scale the median of group 2 is $e^{\widehat{\alpha}_2}=e^{1.5}=4.48$ as large as the median of group 1 . This means that additive effects on the log-scale become multiplicative effects on the original scale. Unfortunately, the statement is only about the median and not the mean on the original scale.

If we also consider a confidence intervals for α_2, e.g. [1.2, 1.8], the transformed version $\left[e^{1.2}, e^{1.8}\right]$ is a confidence interval for e^{α_2} which is the ratio of medians on the original scale.

3 Contrasts and Multiple Testing

3.1 Contrasts:

The F-test is rather unspecific. It basically gives us a “Yes/No” answer for the question “is there any treatment effect at all?”. It doesn’t tell us what specific treatment (or treatment combination) is significant. Such kinds of questions can typically be formulated as a so-called contrast. For instance, we could have the null hypothesis / alternative:

$$\begin{aligned}H_0&:\mu_1-\mu_2=0\\H_A&:\mu_1-\mu_2\neq 0\end{aligned}$$

This can be encoded with a vector $c\in\mathbb{R}^g$:

$$H_0:\sum_{i=1}^{g}c_i\mu_i=0$$

Typically, the side constraint $\sum_{i=1}^{g}c_i=0$ is enforced which ensures that the contrast is about differences between treatments and not about the overall level of our response. A contrasts true (but unknown) value $\sum_{i=1}^{g}c_i\mu_i$ is estimated with:

$$\sum_{i=1}^{g}c_i\widehat{\mu}_i$$

In R, *glht* of the package *multcomp* is used (on a normal ANOVA fit):

```
library(multcomp)
fit.gh <- glht(fit, linfct =
        mcp(group = c(1, -1/2, -1/2)))
summary(fit.gh)
```

This gives us a p-value if the contrast(s) is (are) zero or not.

confint gives a confidence interval for the contrast(s). To test many contrasts at the same time:

```
M <- rbind(new.vs.old   = c(1/2, -1/2, 1/2, -1/2),
           co2.vs.mixed = c(1, 0, -1, 0))
fit.mc <- glht(fit, linfct = mcp(treatment = M))
summary(fit.mc, test = adjusted("none"))
```

Every contrast has an associated sum of squares:

$$SS_c=\frac{\left(\sum_{i=1}^{g}c_i\bar{y}_{i\cdot}\right)^2}{\sum_{i=1}^{g}\frac{c_i^2}{n_i}}$$

With one degree of freedom (therefore $MS_c=SS_c$). This is the square of the t-statistic of the corresponding null hypothesis for the model parameter $\sum_{i=1}^{g}c_i\mu_i$ (without the MS_E). We have:

$$\frac{MS_c}{MS_E}\sim F_{1,N-g}$$

Two contrasts c and c^* are orthogonal if

$$\sum_{i=1}^{g}\frac{c_ic_i^*}{n_i}=0.$$

In this case, the corresponding estimates are (statistically) independent. This means that if we know something about one of the contrasts, this does not help us in making a statement about the other one. If we have g treatments, we can find $g-1$ different orthogonal contrasts (one dimension is already used by the global mean). A set of orthogonal contrasts partitions the treatment sum of squares meaning that if $c^{(1)},\ldots,c^{(g-1)}$ are orthogonal contrasts it holds that:

$$SS_{c^{(1)}}+\cdots+SS_{c^{(g-1)}}=SS_{\text{Trt}}$$

Multiple contrasts are all orthogonal if and only if for the matrix C that represents them, C^TC is diagonal.

3.2 Multiple Testing:

If we simply want to do all pairwise t-tests using pooled standard deviation (without adjustment for multiple testing), we can use:

```
with(d, pairwise.t.test(y, tr, p.adjust.method = "none")
```

The (overall) error rate increases with increasing number of tests. If we perform m tests using an individual significance level α, the probability of making at least one false rejection (if all $H_{0,j}$ are true) is given by (independence tests):

$$\begin{aligned}P\left[\bigcup_{j=1}^{m}\underbrace{\{\text{test } j \text{ falsely reject } H_{0,j}\}}_{A_j}\right]&=1-P\left[\bigcap_{j=1}^{m}A_j^c\right]\\&\overset{iid}{=}1-\prod_{j=1}^{m}P[A_j^c]\\&=1-(1-\alpha)^m\end{aligned}$$

where $(1-\alpha)^m$ is the probability of guessing correctly all the m tests. More generally (any dependence)

$$P\left[\bigcup_{j=1}^{m}\underbrace{\{\text{test } j \text{ falsely reject } H_{0,j}\}}_{A_j}\right] \leq \sum_{j=1}^{m} P[A_j] = m\cdot\alpha$$

If we perform m tests, whereof m_0 null hypotheses are true, we have the following potential outcomes:

	H_0 **true**	H_0 **false**	**Total**
Reject	V	S	R
Accept	U	T	$m-R$
Total	m_0	$m-m_0$	m

V is the number of TYPE 1 error, while T is the number of TYPE 2 error. The family-wise error rate is defined as the probability of rejecting at least one of the true H_0's:

$$\text{FWER} = P(V \geq 1)$$

We say that a procedure controls the family-wise error rate in the strong sense at level α if FWER $\leq \alpha$ for any configuration of true and non-true null hypotheses. The false discovery rate (FDR) is the expected fraction of false discoveries:

$$\text{FDR} = E\left[\frac{V}{R}\right].$$

Controlling FDR at level 0.2 means that in our list of "significant findings" we expect only 20% that are not "true findings" (so called false positives). If a procedure controls FWER at level α, FDR is automatically controlled at level α too. On the other side, a procedure that controls FDR at level α might have a much larger error rate regarding FWER. We call a set of confidence intervals simultaneous confidence intervals at level $(1-\alpha)$ if the probability that all intervals cover the corresponding true parameter value is $(1-\alpha)$.

3.2.1 Bonferroni:

In this approach, we use a more restrictive (individual) significance level of $\alpha^* = \alpha/\text{m}$. It controls the family-wise error rate in the strong sense. Equivalently, we can also multiply the "original" p-values by m and keep using the original α. The confidence intervals based on the adjusted significance level are simultaneous (we use level $1-\alpha/m$ instead of $1-\alpha$). In R, we do the adjustment by

```
summary(fit.gh, test = adjusted("bonferroni"))
```

3.2.2 Bonferroni-Holm:

Bonferroni-Holm is less conservative and uniformly more powerful than Bonferroni. It works as follows:

1. Sort p-values from small to large: $p_{(1)} \leq p_{(2)} \leq \ldots \leq p_{(m)}$
2. For $j = 1, 2, \ldots$. Reject null hypothesis if $p_{(j)} \leq \frac{\alpha}{m-j+1}$
3. Stop when reaching the first non-signifcant p-value.

In R, we use

```
summary(fit.gh, test $=$ adjusted ("holm"))
```

This is a so-called step-down procedure ("stepping-down the sequence of hypotheses").

3.2.3 Scheffé:

The Scheffé procedure controls for the search over any possible contrast. This means we can try out as many contrasts as we like and still get honest p-values. The price for this very nice property is low power. It works as follows: Calculate F-ratio as if ordinary contrast and use the distribution $(g-1)\cdot F_{g-1,N-g}$ instead of $F_{1,N-g}$ to calculate p-values / critical values. In R, this has to be done manually

```
fit.sch <- glht(fit,
linfct = mcp(group = c(1/2, -1, 1/2)))

## p-value according to Scheffe (g = 3, N - g = 27)
pf((summary(fit.sch)$test$tstat)^2 / 2, 2, 27,
    lower.tail = FALSE)
```

3.2.4 Tukey Honest Significant Difference (THSD):

A special case for a multiple testing problem is the comparison between all possible pairs of treatments. There are a total of $g*(g-1)/2$ pairs. We could perform all pairwise t-tests with the function pairwise t test (that uses a pooled standard deviation estimate from all groups). There is a better (more powerful) alternative which is called Tukey Honest Significant Difference. Think of a procedure that is custom tailored for this situation. It gives us both p-values and confidence intervals. In *R*, this is done using

```
TukeyHSD(fit)
```

Note:

- Tukey HSD is better (more powerful) than Bonferroni if all pairwise comparisons are of interest.
- If only a subset: Re-consider Bonferroni.
- Use it generally for pairwise comparison, akso if F test doesn't reject the global null hypotesis

3.2.5 Multiple Comparison with a Control (MCC):

In the same spirit, if we want to compare all treatment groups with a control group, we have a so called multiple comparisons with a control problem. The corresponding procedure is called **Dunnett procedure** which constructs simultaneous confidence intervals for the differences $\mu_i - \mu_g, i = 1, \ldots, g-1$ (assuming group g is control group). Implemented in the add-on package *multcomp.*

```
fit.dunnett <- glht(fit,
            linfct = mcp(treatment = "Dunnett"))
```

Relationship to the F-Test Pairwise comparisons etc. can also be done if the omnibus F-test isn't significant, as they have built-in multiple-testing correction and conditioning on a significant F-test makes them over-conservative. Moreover, the conditional error or coverage rates can be (very) bad.

3.2.6 Statistical Significance vs. Practical Relevance:

- An effect that is statistically significant is not necessarily of practical relevance.
- Instead of simply reporting p-values, one should always consider the corresponding confidence intervals.
- Background knowledge should be used to judge when an effect is potentially relevant.

4 Factorial Treatment Structure

In practice, treatments are often combinations of the level of two or more factors which is called factorial treatment structure. If we see all possible combinations of the levels of two (or more) factors, we call them crossed. In R, we can count the number of observations for every combination of the levels with *xtabs(∼ factor1 + factor2, data = data)*. In a factorial treatment structure, there are typically questions about both factors and / or their possible interplay (we could also use the cell means / one-way ANOVA model for analysis, but then we would ignore the special structure and answering these questions using contrasts is complicated).

Such kind of data can be visualized with an interaction plot (R: *interaction.plot(x.factor = factor1, trace.factor = factor2, response = response))*. For every combination, the average response is calculated. The *x.factor* is plotted on the x-axis, and settings corresponding to the same level of *trace.factor* are connected with a line. Parallel lines indicate no interplay in interaction plot.

5 Two-Way ANOVA model

Two way ANOVA factor A1 A2, B1, B2, B3 with interactions, we get the same mode if we fit a one way ANOVA with 6 levels!

Model: Two-Way ANOVA

- Factor A with a levels, factor B with b levels
- n replicates for every combination of A and B (a balanced design).
- Total of $N = a \cdot b \cdot n$ observations.

$$Y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}$$

where α_i is the main effect of factor A at level i, β_j is the main effect of factor B at level $j, (\alpha\beta)_{ij}$ is the interaction effect between A and B for the level combination i, j (it is not the product $\alpha_i\beta_j$) and ϵ_{ijk} are i.i.d. $N\left(0, \sigma^2\right)$ errors. **Sum-to-zero constraints are used**, $\sum_{i=1}^a \alpha_i = 0$, $\sum_{j=1}^b \beta_j = 0, \sum_{i=1}^a (\alpha\beta)_{ij} = 0$ and $\sum_{j=1}^b (\alpha\beta)_{ij} = 0$.

Src	df	SS	Mean squares	F-rat
A	$a-1$	SS_A	$MS_A = \frac{SS_A}{a-1}$	$\frac{MS_A}{MS_E}$
B	$b-1$	SS_B	$MS_B = \frac{SS_B}{b-1}$	$\frac{MS_B}{MS_E}$
AB	$(a-1)(b-1)$	SS_{AB}	$MS_{AB} = \frac{SS_{AB}}{(a-1)(b-1)}$	$\frac{MS_{AB}}{MS_E}$
Error	$ab(n-1)$	SS_E	$MS_E = \frac{SS_E}{ab(n-1)}$	

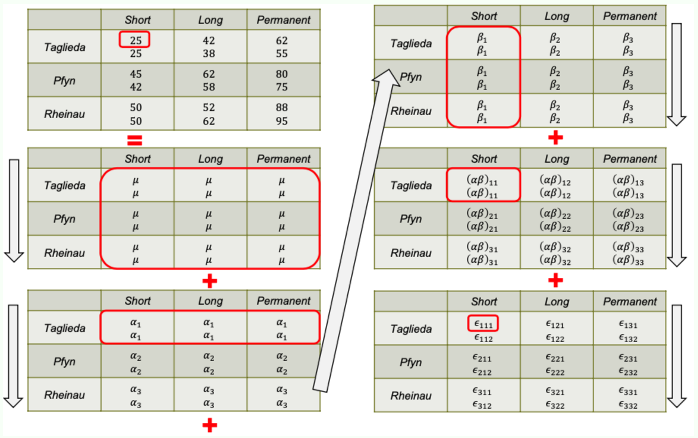
Note $df_{residuals} = N - df_A - df_B - df_{AB} - 1$

Interpretation of Main Effects

- Fitting model with interaction or without yields the same MSE and SS for main effects but different p val and F-value.
- Main effects are nothing else than the average effect when moving from row to row (column to column).
- The interaction effect is the difference to the main effects model, i.e. it measures how far the treatment means differ from the main effects model.
- If there is no interaction, the effects are additive → parallel lines in interaction plot

Example: Needleweight

- "Is effect of light exposure location specific?" (→ so-called **interaction** between light exposure and location)
- "What is the effect of light exposure averaged over all locations?" (→ so-called **main effect** of light exposure)
- "What is the effect of location averaged over all exposure levels?" (→ so-called **main effect** of location)



	Short	Long	Permanent
Taglieda	25 25	42 38	62 55
Pfyn	45 42	62 58	80 75
Rheinau	50 50	52 62	88 95

=

	Short	Long	Permanent
Taglieda	μ μ	μ μ	μ μ
Pfyn	μ μ	μ μ	μ μ
Rheinau	μ μ	μ μ	μ μ

+

	Short	Long	Permanent
Taglieda	α_1 α_1	α_1 α_1	α_1 α_1
Pfyn	α_2 α_2	α_2 α_2	α_2 α_2
Rheinau	α_3 α_3	α_3 α_3	α_3 α_3

+

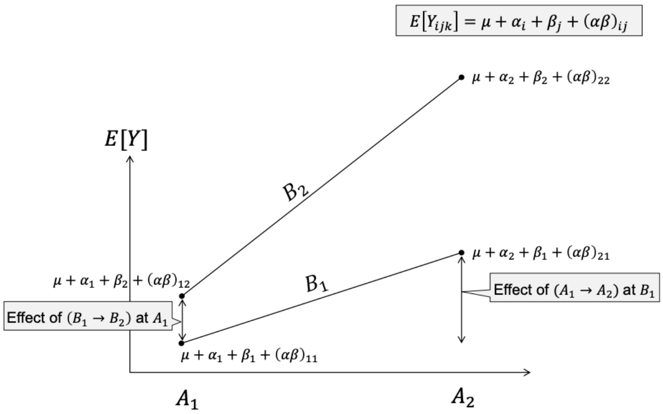
	Short	Long	Permanent
Taglieda	β_1 β_1	β_2 β_2	β_3 β_3
Pfyn	β_1 β_1	β_2 β_2	β_3 β_3
Rheinau	β_1 β_1	β_2 β_2	β_3 β_3

+

	Short	Long	Permanent
Taglieda	$(\alpha\beta)_{11}$ $(\alpha\beta)_{11}$	$(\alpha\beta)_{12}$ $(\alpha\beta)_{12}$	$(\alpha\beta)_{13}$ $(\alpha\beta)_{13}$
Pfyn	$(\alpha\beta)_{21}$ $(\alpha\beta)_{21}$	$(\alpha\beta)_{22}$ $(\alpha\beta)_{22}$	$(\alpha\beta)_{23}$ $(\alpha\beta)_{23}$
Rheinau	$(\alpha\beta)_{31}$ $(\alpha\beta)_{31}$	$(\alpha\beta)_{32}$ $(\alpha\beta)_{32}$	$(\alpha\beta)_{33}$ $(\alpha\beta)_{33}$

+

	Short	Long	Permanent
Taglieda	ϵ_{111} ϵ_{112}	ϵ_{121} ϵ_{122}	ϵ_{131} ϵ_{132}
Pfyn	ϵ_{211} ϵ_{212}	ϵ_{221} ϵ_{222}	ϵ_{231} ϵ_{232}
Rheinau	ϵ_{311} ϵ_{312}	ϵ_{321} ϵ_{322}	ϵ_{331} ϵ_{332}


$E[Y_{ijk}] = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij}$
$E[Y]$
$\mu + \alpha_2 + \beta_2 + (\alpha\beta)_{22}$
B_2
$\mu + \alpha_2 + \beta_1 + (\alpha\beta)_{21}$
$\mu + \alpha_1 + \beta_2 + (\alpha\beta)_{12}$
B_1
Effect of $(A_1 \to A_2)$ at B_1
Effect of $(B_1 \to B_2)$ at A_1
$\mu + \alpha_1 + \beta_1 + (\alpha\beta)_{11}$
A_1
A_2

Parameters are estimated using least squares, which gives: $\widehat{\mu} = \bar{y}_{...}, \widehat{\alpha}_i = \bar{y}_{i..} - \bar{y}_{...}, \widehat{\beta}_j = \bar{y}_{.j.} - \bar{y}_{...}$ and $\widehat{(\alpha\beta)}_{ij} = \bar{y}_{ij.} - \widehat{\mu} - \widehat{\alpha}_i - \widehat{\beta}_j$. We estimate therefore the expected value of the response Y_{ijk} for A at level i and B at level j as

$$\widehat{\mu} + \widehat{\alpha}_i + \widehat{\beta}_j + \widehat{(\alpha\beta)}_{ij} = \bar{y}_{ij.}$$

which is simply the cell mean. In *R*, we use

```
aov(response ~ factor1 * factor2, data = data)
```

which is equivalent to *response* ∼ *factor1* + *factor2* + *factor1:factor2*. *response* ∼ *factor1* + *factor2* would fit a main effects model.

Attention if we fit

```
aov(response ~ factor2 * factor1, data = data)
```

it is different than the previous fit if the design is unbalanced (see unbalanced data section). In that case, the type I sum of squares used in *aov()* is dependent on the order of how the variables appear in the model formula. The interaction has the same p-value because it is adjusted *factor1* and *factor2* in both cases (similar can be applied for fitting main effects only)

Example: Score of ski

```
Full coefficients are

(Intercept):    70.47
ski:            S1      S2      S3
                0.00    1.38    11.83
boot:           B1      B2
                0.00    -9.83
ski:boot:       S1:B1   S2:B1   S3:B1   S1:B2   S2:B2   S3:B2
                0.00    0.00    0.00    0.00    7.72    2.65
```

The estimated satisfaction score of ski type S1 combined with boot type B2 is $70.47 - 9.83 = 60.64$. The estimated satisfaction score of ski type S2 combined with boot type B2 is $70.47 + 1.38 - 9.83 + 7.72 = 69.74$. $S1$ and $B1$ are the reference level.

5.1 Tests:

As for the one-way ANOVA case, the total sum of squares SS_T can be partitioned into different sources

$$SS_T = SS_A + SS_B + SS_{AB} + SS_E$$

where

- $SS_A = \sum_{i=1}^a bn \left(\widehat{\alpha}_i\right)^2$ ("between rows")
- $SS_B = \sum_{j=1}^b an \left(\widehat{\beta}_j\right)^2$ ("between columns")
- $SS_{AB} = \sum_{i=1}^a \sum_{j=1}^b n \widehat{(\alpha\beta)}_{ij}^2$ ("correction")
- $SS_E = \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^n \left(y_{ijk} - \bar{y}_{ij.}\right)^2$ ("within cells")
- $SS_T = \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^n \left(y_{ijk} - \bar{y}_{...}\right)^2$ ("total")

The two-way ANOVA table is in the appendix. Tests can be constructed based on the corresponding F-distribution: For the interaction $H_0 : (\alpha\beta)_{ij} = 0$ for all $i, j; H_A :$ at least one $(\alpha\beta)_{ij} \neq 0$; Under H_0

$$\frac{MS_{AB}}{MS_E} \sim F_{(a-1)(b-1),ab(n-1)}$$

For the main effect A: $H_0 : \alpha_i = 0$ for all $i; H_A :$ at least one $\alpha_i \neq 0$; Under H_0 :

$$\frac{MS_A}{MS_E} \sim F_{a-1,ab(n-1)}$$

For the main effect B: $H_0 : \beta_j = 0$ for all $j; H_A :$ at least one $\beta_j \neq 0$; Under H_0 :

$$\frac{MS_B}{MS_E} \sim F_{b-1,ab(n-1)}$$

The ANOVA table is read bottom to top. First, it is checked if the interaction term is needed or not. If there is no evidence of interaction, we continue with the inspection of the main effects. If there is interaction, we would (typically) stop inspecting whether the main effects are significant or not, i.e. drop no terms from the model. With appropriate contrasts, some specific questions could be inspected. In R, this is often done with the *interaction* function which generates a hyper-factor that has as levels all possible combinations of the provided factors, e.g.:

```
d$exp.loc <- interaction(d$location, data$exposure
```

Then, a normal ANOVA analysis with a contrast can be used where the hyper-factor is the factor.

If we continue with individual analyses after the two-way model, a significant interaction term means we should do the individual models (of one factor) per level (of the other factor). We can improve the tests by "re-using" the MS_E with the corresponding degrees of freedom of the full model.

5.2 Single Replicates:

If we only have a single observation in each "cell" $(n = 1)$ we cannot do statistical inference anymore with a model including the interaction as we have no idea of the experimental error (for every treatment combination we only have one observation). Reason: Perfect fit, all residuals are zero (or: # parameters = # observations). However, we can still fit a main effects only model. If the data generating mechanism actually contains an interaction, we are fitting a wrong model. The consequences are that the corresponding tests will be too conservative, meaning p-values will be too large. This is not a problem as the type 1 error rate is still controlled. We "just" lose power.

Quite often, we can get rid of interactions if we look at the problem on a different scale, i.e., if we transform the response appropriately. A famous example is the logarithm. Effects that are multiplicative on the original scale become additive on the log-scale, i.e., no interaction is needed on the log-scale.

Model: Tukey One-Degree of Freedom Interaction

A very special form of interaction where only one parameter λ is used (here, $\alpha_i\beta_j$ is the product of the main effects):

$$Y_{ij} = \mu + \alpha_i + \beta_j + \lambda\alpha_i\beta_j + \epsilon_{ij}$$

5.3 Unbalanced Data:

With unbalanced data, no independent estimates anymore and the sum of squares cannot be uniquely partitioned into different sources anymore (for some part of the variation, it is not clear to what source we should attribute it). The problem is solved by using a model comparison approach. The sum of squares of a factor can be interpreted as the reduction of residual sum of squares when adding the factor to the model. But now (in contrast to the balanced case), it matters if the other factors are in the model. With $SS(B \mid 1, A)$, the reduction in residual sum of squares when comparing the model $(1, A, B)(= y \sim A + B)$ with $(1, A)(= y \sim A)$. The 1 is the overall mean μ. Interpretation of the corresponding test is as follows: "Do we need factor B in the model if we already have factor A (or after having controlled for factor A)?" There are different ways / types of model comparison approaches:

1. Type I (sequential): Sequentially build up model (depends on the "ordering" of the model terms!): $SS(A \mid 1), SS(B \mid 1, A), SS(AB \mid 1, A, B)$
2. Type II (hierarchical): Control for the influence of the largest hierarchical model not including the term of interest: $SS(A \mid 1, B), SS(B \mid 1, A), SS(AB \mid 1, A, B)$
3. Type III (fully adjusted): Control for all other terms: $SS(A \mid 1, B, AB), SS(B \mid 1, A, AB), SS(AB \mid 1, A, B)$

summary of an *aov* or *anova* gives type 1 (hence the order matters). In a model such as $\sim A+B+A:B$, R will report the difference in sums of squares between the models $\sim 1, \sim A, \sim A+B$ and $\sim A+B+A:B$. If the model were $\sim B+A+A:B$, R would report differences between $\sim 1, \sim B, \sim A+B$, and $\sim A+B+A:B$. In the first case the sum of squares for A is comparing ~ 1 and $\sim A$, in the second case it is comparing $\sim B$ and $\sim B+A$. In a nonorthogonal design (i.e., most unbalanced designs) these comparisons are (conceptually and numerically) different.

For type II, the function *Anova* of *car* can be used, for example:

```
library(car)
Anova(fit, type = "II")
```

For type III, the command *drop1* (which reports deletion of single terms) can be used (but contrast option has to be *contr.sum* in this case). The following command can be used to get the type III sum of squares for all variables in the model:

```
drop1(fit, scope = ~., test = "F")
```

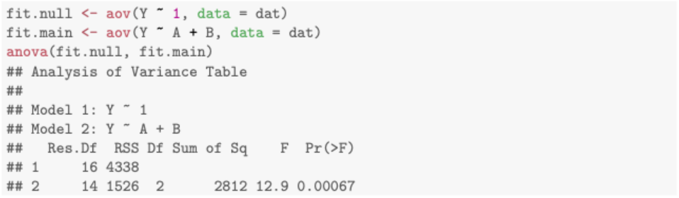
It can be shown that for type I and II the null hypothesis is weighted by the sample sizes wheres for type III it is $H_0 : \alpha_1 = \ldots = \alpha_a$ which means that the unweighted means are equal (which we generally want to test).

Note:

- For main effects only models, Type II and Type III coincide.
- Type I and Type II SS do not typically coincide for main effects if model is unbalanced.
- The SS of the interaction term is the same for all 3 approaches and so it is the corresponding p-value (approach based on reduction of SS once the other variables are already in the model).
- If there is a significant interaction, tests of the corresponding main effects are typically difficult to interpret (better use individual models).
- With balanced data, we always get the same result, no matter what type we use.

Example: Data puzzle

Given unbalanced dataset we fit main effects $Y \sim A * B$. Can we simutaneously drop A and B from the model?



```
fit.null <- aov(Y ~ 1, data = dat)
fit.main <- aov(Y ~ A + B, data = dat)
anova(fit.null, fit.main)
## Analysis of Variance Table
##
## Model 1: Y ~ 1
## Model 2: Y ~ A + B
##   Res.Df  RSS Df Sum of Sq    F  Pr(>F)
## 1     16 4338
## 2     14 1526  2      2812 12.9 0.00067
```

Because of the p-value of 0.00067, we conclude that we cannot simultaneously drop A and B from the model. Why could happen that the main effects are not significant (using for example *Anova(...)*)? Due to the unbalancedness, the two factors A and B are highly correlated. Basically, if we put one factor into the model, it already accounts for most of the variation and there is no variation left to explain for the second factor.

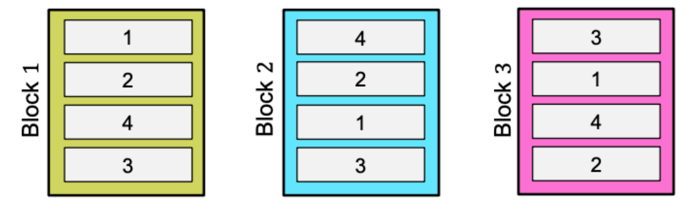
6 Block Designs

Quite often we already know that experimental units are not homogeneous. Using a completely randomized design in such a situation would still be a valid procedure. However, making explicit use of the special "structure" of the experimental units (blocking) typically helps reducing variance. For instance, in a paired t-test we use blocking (e.g. on persons and therefore eliminate the person-to-person variation). This is extended to $g > 2$.

6.1 Randomized Complete Block Designs:

Model: RCBD

Assume we have r blocks containing g experimental units each.



The randomized complete block design (RCBD) uses a restricted randomization scheme: Within every block (e.g. location), the treatments are randomized to the experimental units (e.g. plots of land). The design is called complete because we see the complete set of treatments within every block. In the most basic form, we assume that we don't have replicates within a block (i.e. we only see every treatment once in every block).

Main effects model:

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

where the α_i 's are the treatment effects and β_j 's the block effects. According to this model we implicitly assume that blocks only cause additive shifts.

Typically, we are not making inference about blocks (we already knew beforehand that blocks will be different!).

Note:

- RCBD tries to ensure homogeneity within blocks.
- Given block df_{block}, treatment df_{treat} and residual with df_{res},

$$\text{\#observations per block} = \frac{df_{block} + df_{treat} + df_{res} + 1}{\text{\#blocks}}$$

- The error corresponding to a randomized complete block design with r blocks has $r-1$ less degrees of freedom than the error corresponding to a completely randomized design on the same number of experimental units (**Negative consequence**).
- The number of experimental units in each block of an RCBD cannot be less than the number of treatment, otherwise the block would be incomplete.
- Blocking must be done before randomization, otherwise we are not guaranteed to have that each treatment appears in each block.

6.1.1 Interaction of Treatment with Block Factor:

- The blocking may result in (very) large differences between units from different blocks (this is OK because we used blocking for this reason!).
- The treatment effects are constant from block to block.
- If we want to fit a model with interaction, we need more than one observation per treatment and block combination.

Instead of a single treatment factor we can also have a factorial treatment structure within every block, e.g. a two-factor factorial which we would model as $Y \sim Block + A * B$.

Source	**df**
Block	$r-1$
A	$a-1$
B	$b-1$
AB	$(a-1)\cdot(b-1)$
Error	$(ab-1)\cdot(r-1)$
Total	$rab-1$

Here, we could actually test the interaction between A and B even if every level combination of A and B appears only once in every block (only have one replicate per AB combination per block.). As we have multiple blocks, we have multiple observations for every level combination of A and B ! However, a randomized complete block design can only be used with one blocking factor.

Note

- In an RCB (randomized complete block design), we can only test interactions between block and treatment if we have replicates.
- Blocking can increase precision, even if the p-value corresponding to the block factor is not significant.

6.2 Precision:

In a RCB design, the squared standard errors are $\frac{\sigma^2_{RCB}}{r}$ (where r is the number of blocks) and in a completely randomized design $\frac{\sigma^2_{CRD}}{n}$. If we want to have the same precision, we have to ensure that:

$$\frac{\sigma^2_{RCB}}{r} = \frac{\sigma^2_{CRD}}{n}$$

Therefore, if we knew both squared standard errors, we would have to use a ratio of $\frac{n}{r} = \frac{\sigma^2_{CRD}}{\sigma^2_{RCB}}$. σ^2_{RCB} is estimated by the MS_E of our RCB and σ^2_{CRD} can be estimated using a weighted average of MS_E and MS_{Block}. The relative efficiency is then defined as:

$$RE = \frac{\hat{\sigma}^2_{CRD}}{\hat{\sigma}^2_{RCB}}$$

And gives us the ratio $\frac{n}{r}$ (which is interpreted as how many experimental units would be needed by a CRD to achieve the same efficiency / precision). Easier for a quick check is to look at the ratio $\frac{MS_{Block}}{MS_E}$, because:

$$\frac{MS_{Block}}{MS_E} > 1 \Leftrightarrow \text{Relative Efficiency } > 1$$

Example: Two factor factorial

Suppose one designed a two-factor factorial with factors A and B in a randomized complete block design with a blocking factor C. Which of the following models should we use if we assume an additive effect of the block factor? Remark: α_i are the coefficients of A, β_j the coefficients of B and γ_k the coefficients of C.

$$Y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \gamma_k + \epsilon_{ijk}$$

6.3 Multiple Block Factors:

We can also block on more than one factor.

6.3.1 Latin Square Design:

- Each treatment (the Latin letters) appears exactly once in each row and exactly once in each column (i.e. every treatment appears exactly once for each level of any of the two block factors).
- A Latin Square blocks on both rows and columns simultaneously.
- A Latin Square needs to have g treatments (the Latin letters), two block factors having g levels each (the rows and the columns)
- Total of g^2 experimental units.
- We're seeing only g^2 out of g^3 possible combinations (but the subset we see is selected in a smart, balanced way).
- Use Fisher-Yates algorithm to find a Latin Squares

Use main effects model with treatment, row and column effects.

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

to analyze data from a Latin square design. Here, α_i 's are the treatment effects and β_j (rows) and γ_k (columns) are the block effects with the usual side-constraints. The design is balanced having the effect that our usual estimators and sums of squares are "working". In R we would use the model formula $Y \sim$ Block1 + Block2 + Treat.

Latin Squares can have few degrees of freedom for the error if g is small, making detection of treatment effects difficult:

g	df of MS_E
3	2
4	6
5	12

Example: Latin Squares machines

Operator	$P1$	$P2$	$P3$	$P4$	$P5$
Mon	E	B	C	A	D
Tue	B	D	E	C	A
Wed	A	C	D	B	E
Thu	C	E	A	D	B
Fri	D	A	B	E	C

Often, one blocking factor is time: Think of testing 5 different machines (A, B, C, D, E) on 5 days with 5 operators (response: yield of machine).

6.3.2 Graeco-Latin Squares:

If we have another blocking criterion with g levels (denoted by Greek letters, e.g. with levels $\alpha, \beta, \gamma, \delta$), we can use a Graeco-Latin Squares design. The conditions are that the Latin letters (treatments) occur once in each row and column and the Greek letters (third block factor) occur once in each row and column, i.e. we have two superimposed Latin Squares. In addition, each Latin letter occurs exactly once with each Greek letter. We use the main effects model to analyze the data:

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

Where α_i is the treatment, β_j the block factor 1 (rows), γ_k the block factor 2 (columns), δ_l the block factor 3 (Greek letters).

	C_1	C_2	C_3	C_4
R_1	$A\alpha$	$B\gamma$	$C\delta$	$D\beta$
R_2	$B\beta$	$A\delta$	$D\gamma$	$C\alpha$
R_3	$C\gamma$	$D\alpha$	$A\beta$	$B\delta$
R_4	$D\delta$	$C\beta$	$B\alpha$	$A\gamma$

7 Random and Mixed Effects Models

Up to now, treatment effects (the α_i's) were fixed, unknown quantities that we tried to estimate. This means we were making a statement about a specific, fixed set of treatments.

7.1 Random Effects models:

7.1.1 One-Way ANOVA:

In this point of view, treatments are random samples from a large population of treatments. For example, a random sample of school classes that were drawn from all school classes in a country or machines that were randomly sampled from a large population of machines. Typically, we are interested in making a statement about some properties of the whole population.

Model: Random Effect

$$Y_{ij} = \mu + \alpha_i + \epsilon_{ij}$$

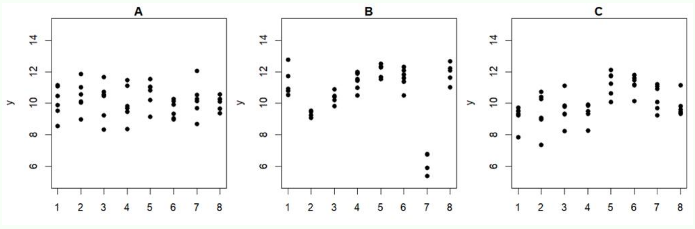
where α_i i.i.d. $\sim N\left(0, \sigma_\alpha^2\right), \epsilon_{ij}$ i.i.d. $\sim N\left(0, \sigma^2\right)$. The α_i is also called a random effect. This introduces a new parameter σ_α^2 which is the variance of the random effect.

The expected value of Y_{ij} is $E\left[Y_{ij}\right] = \mu$ (but $E\left[Y_{ij} \mid \alpha_i\right] = \mu + \alpha_i$). The variance is $\operatorname{Var}\left(Y_{ij}\right) = \sigma_\alpha^2 + \sigma^2$ and the correlation:

$$\operatorname{Cor}\left(Y_{ij}, Y_{kl}\right) = \begin{cases} 0 & i \neq k \\ \sigma_\alpha^2 / \left(\sigma_\alpha^2 + \sigma^2\right) & i = k, j \neq l \\ 1 & i = k, j = l \end{cases}$$

Observations from a different "group" (e.g. from a different machine) are uncorrelated, while observations from the same "group" are correlated with an intraclass correlation (ICC) of $\sigma_\alpha^2 / \left(\sigma_\alpha^2 + \sigma^2\right)$. It is large if $\sigma_\alpha^2 \gg \sigma^2$ which means that observations from the same "group" (e.g. machine) are very similar to each other. Therefore, values sharing the same α_i are correlated (while all are independent in the fixed effects model). The same holds for multiple random effects. For them, the correlation is the sum of shared variance components divided by the sum of all variance components. Parameter estimation for σ_α^2 and σ^2 is typically done using restricted maximum likelihood estimation (REML). In R, the lmer function is used. A random effect is specified using (1/ column) which means that all variables sharing the same column (could also be an interaction) will get the same random effect α_i.

Example: ICC: A 0, B 0.9, C 0.5



An example call looks like this

```
library(lmerTest)
fit.sire <- lmer(weight ~ (1 | sire), data = animals)
```

The output of *summary* contains:

```
## Random effects:
##  Groups    Name         Variance  Std. Dev.
##  sire      (Intercept)  116.7     10.81
##  Residual               463.8     21.54
```

Which means $\widehat{\sigma}_\alpha^2 = 116.7$ and $\widehat{\sigma}^2 = 463.8$. With *confint(fit.sire, oldNames = FALSE)*, it's possible to get confidence intervals. The intervals are usually longer than if we would fit the model with the normal aov function because we are making inference about the whole population (whereof the fixed effects model makes inference about the specific ones we have seen / measured). Estimates (conditional means) for the random effects can be retrieved with ranef. These can then also be plotted in a QQ-plot to verify the normality assumption, e.g.:

```
## QQ-plots of random effects
qqnorm(ranef(fit.sire)$sire[,1], main = "sire")
## QQ-plots of residuals
qqnorm(resid(fit.sire), main = "residuals")
```

With *exactRLRT(fit.sire)* from the *RLRsim* library, we can do (simulation based) tests for variance components of random effects (i.e. if they are zero), which we are generally not that interested in (more in the confidence intervals).

Note: we can check the model assumptions via TA plot, QQ plot of residuals or QQ plot of estimated random effects.

Example: Tea vendor

Suppose we fit a model using the random effect model. Then If we instead consider all the factors fixed, fitting the model so as

```
fit2 <- aov(Rating ~  Type + Subject, data = tea)
coef(fit2)
## (Intercept)       Type2       Type3       Type4    Subject2    Subject3
##  0.04753849  1.72158070  2.99164302  2.33998167  0.29019204  1.44925105
##    Subject4    Subject5    Subject6    Subject7    Subject8    Subject9
## -0.55735453  1.01772272  0.39756117  1.24372812  2.31235864 -0.38621618
##   Subject10
##  1.94805373
```

The number 0.05 + 1.72 is an estimate of how the reference subject (Subject1) rates the second tea type in expectation, not the expected value across all potential subjects (averaged).

Example: Lost number of partecipants

Assume you have done an analysis using random effects but after some time the data got lost and the number of random people selected in therefore not the same anymore. Does the loss of data have an influence on the calculation of the ANOVA table? What R-function would you use?

The decomposition of the total sum of squares into orthogonal components does not work anymore. drop1 can be used to get type 3 sum of squares.

7.1.2 Multiple factors:

Model: Random effect with multiple factors

It's also possible to model multiple factors (with interaction) as random models, e.g.

$$Y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}$$

with

$$\alpha_i \text{ i.i.d. } \sim N\left(0, \sigma_\alpha^2\right), \beta_j \text{ i.i.d. } \sim N\left(0, \sigma_\beta^2\right),$$
$$(\alpha\beta)_{ij} \text{ i.i.d. } \sim N\left(0, \sigma_{\alpha\beta}^2\right)$$

In *R*, the model is fitted like this:

```
fit.trigly <- lmer(y ~ (1 | day) + (1 | machine)
    + (1 | machine:day), data = trigly)
```

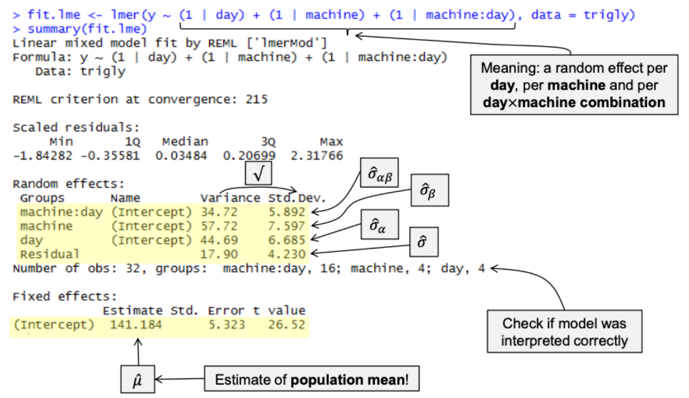
The more random effects two observations share, the larger the correlation. It is given by

$$\frac{\text{sum of shared variance components}}{\text{sum of all variance components}}$$

The correlation between two (different) observations from the same operator on different machines is given by

$$\frac{\sigma_\beta^2}{\sigma_\alpha^2 + \sigma_\beta^2 + \sigma_{\alpha\beta}^2 + \sigma^2}$$

Example: Machine Performance



```
> fit.lme <- lmer(y ~ (1 | day) + (1 | machine) + (1 | machine:day), data = trigly)
> summary(fit.lme)
Linear mixed model fit by REML ['lmerMod']
Formula: y ~ (1 | day) + (1 | machine) + (1 | machine:day)
   Data: trigly

REML criterion at convergence: 215

Scaled residuals:
     Min       1Q   Median       3Q      Max
-1.84282 -0.35581  0.03484  0.20699  2.31766

Random effects:
 Groups      Name        Variance Std.Dev.
 machine:day (Intercept) 34.72    5.892
 machine     (Intercept) 57.72    7.597
 day         (Intercept) 44.69    6.685
 Residual                17.90    4.230
Number of obs: 32, groups:  machine:day, 16; machine, 4; day, 4

Fixed effects:
            Estimate Std. Error t value
(Intercept)  141.184      5.323   26.52
```

Meaning: a random effect per **day**, per **machine** and per **day×machine combination**
$\sqrt{}$
$\hat{\sigma}_{\alpha\beta}$
$\hat{\sigma}_\beta$
$\hat{\sigma}_\alpha$
$\hat{\sigma}$
Check if model was interpreted correctly
$\hat{\mu}$
Estimate of **population mean!**

Source	Percentage	Interpretation
Day	$\frac{44.7}{155} = 29\%$	Day to day operational differences (e.g., due to daily calibration).
Machine	$\frac{57.7}{155} = 37\%$	Variability in machine performance.
Interaction	$\frac{34.7}{155} = 22\%$	Variability due to inconsistent behavior of machines over days (calibration inconsistency within the same day?).
Error	$\frac{17.9}{155} = 12\%$	Variation in serum samples.

7.2 Nesting:

When we have nested factors, a factor can have a different meaning depending on another factor. Let's consider this example: The strength of a chemical paste product was measured for a total of 60 samples coming from 10 randomly selected delivery batches each containing 3 randomly selected casks ("Fässer"). Hence, two samples were taken from each cask. Cask 1 in batch 1 has nothing to do with cask 1 in batch 2 and so on. The "1" of cask has a different meaning for every batch. Hence, cask and batch are not crossed. We say cask is nested in batch.

Model: Nesting

We have that

$$Y_{ijk} = \mu + \alpha_i + \beta_{j(i)} + \epsilon_{k(ij)}$$

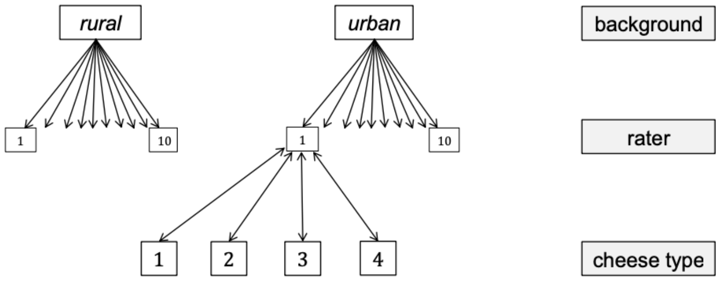
where α_i is the (random) effect of batch and $\beta_{j(i)}$ is the (random) effect of cask within batch with the usual assumptions α_i i.i.d. $\sim N\left(0, \sigma_\alpha^2\right), \beta_{j(i)}$ i.i.d. $\sim N\left(0, \sigma_\beta^2\right)$.

There are multiple ways to tell *lmer* about the nesting structure. We can use the notation *(1/ batch/cask)* which means that we want to have a random effect per batch and per cask within batch. We could also use *(1 | batch)* + *(1 | cask:batch)* which means that we want to have a random effect per batch and a random effect per combination of batch and cask (which is the same as a nested effect). If we already have a combination of batch and cask in the data (e.g. sample with values A:a, A:b, B:a, ...), (1 | batch) + (1 | sample) can also be used. But we can't use *(1 | batch)* + *(1 | cask)* because then all casks with the same number (across batches) would share the same effect. In a fully nested design, every factor is nested in its predecessor.

7.2.1 Example: Cheese tasting:

Setup: Four 50-pound blocks of different cheese types are available, 10 students at random with rural background, 10 students at random with urban background. Each rater will taste 8 bites of cheese (presented in random order). The eight bites consist of two from each cheese type. Hence, every rater gets every cheese type twice.
Factors $A \times B(A) \times C$

- A: background, levels = "rural", "urban"
- B: rater, levels = 1, ...,10 (or 20); nested in background
- C: cheese type, levels = 1, 2, 3, 4



Use model

$$Y_{ijkl} = \mu + \alpha_i + \beta_{j(i)} + \gamma_k + (\alpha\gamma)_{ik} + (\beta\gamma)_{jk(i)} + \epsilon_{l(ijk)}$$

Term	Interpretation
α_i	Main effect of background.
$\beta_{j(i)}$	Random effect of rater: Allows for an individual "general cheese liking" level of a rater.
γ_k	Main effect of cheese type.
$(\alpha\gamma)_{ik}$	Fixed interaction effect between background and cheese type: Allows for a background specific cheese type preference.
$(\beta\gamma)_{jk(i)}$	Random interaction between rater and cheese type: Allows for an individual deviation from the population average "cheese type" effect.

7.2.2 Example: Cardiac Valve types:

Setup: four different cardiac valve types (factor type with 4 levels) and six different pulse rates (factor pulse with six levels). Two different valves of each type (factor valve with 8 levels) were chosen and run at all six pulse rates. The experiment was carried out in a completely randomized fashion (meaning the valves had to be put in the machine and taken out again quite often).

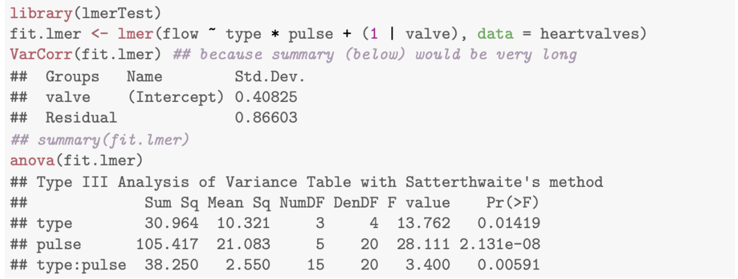
Valve Type (type):		1		2		3		4	
Valve number (valve):		1	2	3	4	5	6	7	8
Pulse rate:	1	2	3	4	2	6	5	7	5
	2	4	4	4	4	5	5	5	4
	3	5	7	4	3	5	6	6	5
	4	3	5	5	3	8	10	9	10
	5	7	7	8	5	9	9	10	11
	6	6	6	6	7	7	8	8	9

Model: This is a mixed effects model of type: type × valve (type) × pulse. That is, factor valve is nested in factor type and both factor type and valve are crossed with factor pulse. There is only one observation per cell, so the interaction term valve:pulse cannot be separated from the error term. Hence, this interaction term is not added to the model. The model is:

$$Y_{ijk} = \mu + \alpha_i + \beta_{j(i)} + \gamma_k + (\alpha\gamma)_{ik} + \epsilon_{ijk}$$

$i = 1, \ldots, 4, j = 1, 2,$ (for each value of i), $k = 1, \ldots, 6,$ where:

- μ is the global mean,
- α_i is the fixed effect of the type,
- $\beta_{j(i)}$ i.i.d. $\mathcal{N}\left(0, \sigma_\beta^2\right)$ is the random effect of the individual valve,
- γ_k is the fixed effect of the pulse rate,
- $(\alpha\gamma)_{ik}$ is the fixed effect of the interaction between type and pulse rate, and
- ϵ_{ijk} i.i.d. $\mathcal{N}\left(0, \sigma^2\right)$ is the error term.



```
library(lmerTest)
fit.lmer <- lmer(flow ~ type * pulse + (1 | valve), data = heartvalves)
VarCorr(fit.lmer) ## because summary (below) would be very long
##  Groups   Name        Std.Dev.
##  valve    (Intercept) 0.40825
##  Residual             0.86603
## summary(fit.lmer)
anova(fit.lmer)
## Type III Analysis of Variance Table with Satterthwaite's method
##             Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## type        30.964  10.321     3     4  13.762   0.01419
## pulse      105.417  21.083     5    20  28.111 2.131e-08
## type:pulse  38.250   2.550    15    20   3.400   0.00591
```

Since valve is already encoded in nested notation in our data set, we fit the model using (1 | valve). Otherwise, we would have to use (1 — valve:type).

7.3 Mixed Effects Model:

Mixed effects models are models which contain both random and fixed effects. For instance, if we have three brands of machines and six workers are chosen randomly among the employees of a factory to operate each machine three times.

Model: Mixed Effects

$$Y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}$$

Where α_i is the fixed effect of machine i (with the usual side constraint), β_j is the random effect of worker j and $(\alpha\beta)_{ij}$ is the corresponding (random) interaction with β_j i.i.d. $\sim N\left(0, \sigma_\beta^2\right), (\alpha\beta)_{ij}$ i.i.d. $\sim N\left(0, \sigma_{\alpha\beta}^2\right)$.

The model is fitted as

```
fit <- lmer(score ~ Machine + (1 | Worker) +
    (1 | Worker:Machine), data = Machines)
```

The *lmerTest* allows to get p-values (global F test) for the fixed effect (Machine) by calling *anova(fit)* with the following sample output:

```
##      Sum Sq Mean Sq NumDF DenDF F.value    Pr(>F)
## Mach 38.051  19.025     2    10  20.576 0.0002855
```

The degrees of freedom come from the fact that we are comparing the variation between different machines (with 2 df) to the variation due to the interaction between machine and workers (having $2 * 5 = 10$ df).

With *rand(fit)*, we can do conservative tests for the variance components of the random effects. The interpretation of the intercept (in the *coef* output) depends on the chosen model. For a model with only fixed effects, it corresponds to the reference (e.g. reference treatment, reference subject) whereas for a mixed model, it corresponds to the reference of the fixed effect (e.g. treatment) and the expected value over all elements of the random effect (e.g. subjects).

7.3.1 What if We Use a Purely Fixed Effects Model?:

If we use a purely fixed effects model and a mixed effects model, the p-values of the purely fixed effects model will be much more significant. This is because in the mixed effects model, we're doing inference about the population average (but see only some part of the population) whereas in the fixed effects model, we're making a statement about the observed data.

Example: Mixed effects

Assume that we fit a mixed effects model with (crossed) factors A and B. Factor A has a fixed effect on the response and factor B has a random effect on the response. There is also a random effect specifically for the interaction between A and B. The fixed effect and confidence intervals of A are interpreted as the population average of the effect of A (across all levels of factor B).

8 Split-Plot Designs

If instead of Split plot I fit a two way ANOVA, the p-values will be two optimistic!.
A split-plot design is a special case of a design with factorial treatment structure. The standard split-plot design consists of two experiments with different experimental units of different "sizes".

Example: Split splot fitting

We consider a split-plot experimental design with factor A applied on the whole plots and factor C applied on the split plots. There is an additional factor B which represents complete blocks on the whole-plot level. To fit a model we use

```
lmer(Y ~ B + A * C + (1 | B:A))
```

We typically apply a split-plot design if it is more difficult to vary one treatment factor compared to another treatment factor. Indeed the treatment on the whole-plot level is more difficult to apply than the treatment on the split-plot level.

8.1 Example: fertilization of plots of lands:

Imagine a farmer that randomizes and applies two fertilization "schemes" ("control" and "new") to eight plots of land. In addition, each plot is divided into four subplots. In each plot, four different strawberry varieties are randomized to the subplots. He is interested in the effect of fertilization scheme and strawberry variety on fruit mass. In total, there are $8 * 4 = 32$ observations. To set up a correct model we have to follow the randomization procedure that was applied. There were two randomizations involved here:

- fertilization schemes were randomized and applied to *plots* of land
- strawberry varieties were randomized and applied to *subplots.*

Hence, an experimental unit for fertilizer is given by a *plot* of land, while for strawberry variety, the experimental unit is given by a *subplot*. Fertilizer is the so-called whole-plot factor and strawberry variety the split-plot factor. A whole-plot is given by a plot of land and a split-plot by a subplot of land. As we have two different sizes of experimental units, we also need two error terms to model the corresponding experimental errors. We need one error term "acting" on the plot level and another one on the subplot level. Let Y_{ijk} be the mass of the kth replicate of a plot with fertilization scheme i and strawberry variety j. We use the model

$$Y_{ijk} = \mu + \alpha_i + \eta_{k(i)} + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk},$$

where α_i is the fixed effect of fertilization scheme, β_j the fixed effect of strawberry variety and $(\alpha\beta)_{ij}$ is the corresponding interaction term. $\eta_{k(i)}$ is the whole-plot error where the nesting structure ensures that we get a whole-plot error per plot of land. ϵ_{ijk} is the split-plot error (the "usual" error). We assume $\eta_{k(i)}$ i.i.d. $\sim N\left(0, \sigma_\eta^2\right)$, ϵ_{ijk} i.i.d. $\sim N\left(0, \sigma^2\right)$. $\alpha_i + \eta_{k(i)}$ can be thought of as the "reaction" of an individual plot on the i th fertilization scheme. All plot specific properties are included in the whole-plot error $\eta_{k(i)}$. The fact that all subplots on the same plot share the same whole-plot error has the side-effect that observations from the same plot are modeled as correlated data. The part $\beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}$ is the "reaction" of the subplot on the j th variety (including a potential interaction with the i th fertilization scheme). All subplot specific properties can now be found in the split-plot error ϵ_{ijk}. If we only consider fertilization scheme, we do a completely randomized design here (with plots as experimental units). The first part of the model formula is actually the corresponding model equation. On the other side, if we only consider variety, we could treat the plots as blocks and would have a randomized complete block design on this "level" including an interaction term (this is what we see in the second part of the model formula).

In R, we use a mixed model. The whole-plot error (acting on plots) can easily be incorporated with *(1 | plot)*. The split-plot error (acting on the subplot level) is automatically included as it is on the level of individual observations. We therefore have

```
fit <- lmer(mass ~ fertil * variety + (1 | plot),
    data = d)
```

With *anova(fit)*, we can look at the F-tests:

```
##       Sum  Sq. Mean  Sq. NumDF DenDF F. value  Pr(>F)
## f     137.413  137.413    1     6    68.240 0.0001702
## v     96.431   32.144     3     18   15.963 2.594e-05
## f:v   4.173    1.391      3     18   0.691  0.5695061
```

The interaction is not significant while the two main effects are. We have 6 denominator degrees of freedom for fertilizer because we basically performed a completely randomized design with eight experimental units and a treatment factor having two levels, i.e. $df = 8 - 1 - 1$.

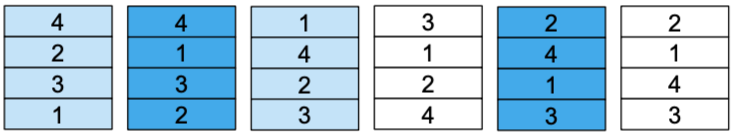
8.2 Example: Irrigation and Corn Variety:

Consider the following factorial problem:

- 4 different corn varieties
- 3 different irrigation levels
- Response: Biomass
- Available resources: 6 plots of land

We cannot vary the irrigation level on a too small scale. We are "forced" to use "large" experimental units for the factor irrigation level. Assume that we can use one specific irrigation level on each of the 6 plots.

We randomly assign each irrigation level to 2 of the plots, the so-called whole plots or main plots. In each of the plots, we randomly assign the 4 different corn varieties to the so- called split plots (or sub plots).



Two independent randomizations are being performed!
Note We also call irrigation level the whole-plot factor and corn variety the split- plot factor.

- Whole plots (plots of land) are the experimental units for the whole-plot factor (irrigation level).
- Split plots (subplots of land) are the experimental units for the split-plot factor (corn variety).
- In the split-plot "world", whole plots act as blocks

Basically, we are performing two different experiments in one: each experiment has its own randomization, each experiment has its own idea of experimental unit.
We use a mixed model formulation with two different errors:

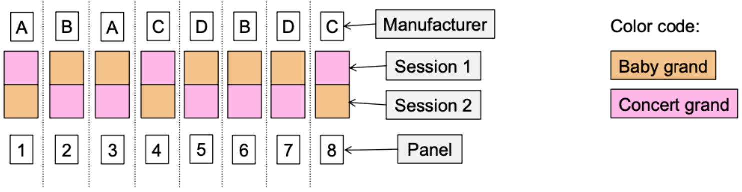
$$Y_{ijk} = \mu + \alpha_i + \eta_{k(i)} + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{k(ij)}$$

where α_i is fixed effect of irrigation, $\eta_{k(i)} \sim N\left(0, \sigma_\eta^2\right)$ is the whole-plot error, β_j is the fixed effect of corn variety, $(\alpha\beta)_{ij}$ is the (fixed) interaction between irrigation and corn variety, and $\epsilon_{k(ij)} \sim N(0, \sigma^2)$ is the split-plot error.
This means: Observations in the same whole plot share the same whole-plot error $\eta_{k(i)}$ and are therefore not independent.
In R, this model can be easily fitted using lmer with a random effect (better terminology here: error) of the form $(1|whole.plot)$.

8.3 Example: Pianos:

Setup:

- 2 piano types from each of 4 manufacturers (A, B, C, D).
- 40 music students are divided at random into 8 groups ("panels") of 5 students each.
- 2 panels are assigned at random to each manufacturer (= 2 panels per manufacturer).
- Each panel goes to the concert hall and hears (blindfolded) the sound of both pianos (in random order).
- Response: Average rating of the 5 students in the panel (hence, a student is "only" a measurement unit here).



The whole plots are the 8 panels. The whole-plot factor is the manufacturer. The split plots are the 2 session time-slots. The split-plot factor is the piano type.
We use the model

$$Y_{ijk} = \mu + \alpha_i + \eta_{k(i)} + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{k(ij)}$$

where α_i is fixed effect of manufacturer, $\eta_{k(i)} \sim N\left(0, \sigma_\eta^2\right)$ is the whole-plot error, β_j is the fixed effect of piano type, $(\alpha\beta)_{ij}$ is the (fixed) interaction between manufacturer and piano type, and $\epsilon_{k(ij)} \sim N(0, \sigma^2)$ is the split-plot error.
Again: This means that observations in the same whole-plot share the same whole-plot error $\eta_{k(i)}$ and are therefore not independent.ì

8.4 Properties:

Typically, split-plot designs are suitable for situations where one of the factors can only be varied on a "large" scale. E.g., fertilizer or irrigation on (large) plots of land. While "large" was literally large in the previous example, this is not always the case. Let us consider an example with a machine running under different settings using different source material. While it is easy to change the source material it is much more tedious to change the machine settings. Hence, we don't want to change the machine settings too often. We could think of an experimental design where we change the machine setting and keep using the same setting for different source materials. This means we are not completely randomizing machine setting and source material. This would be another example of a split-plot design where machine settings is the whole-plot factor and source material is the split-plot factor.

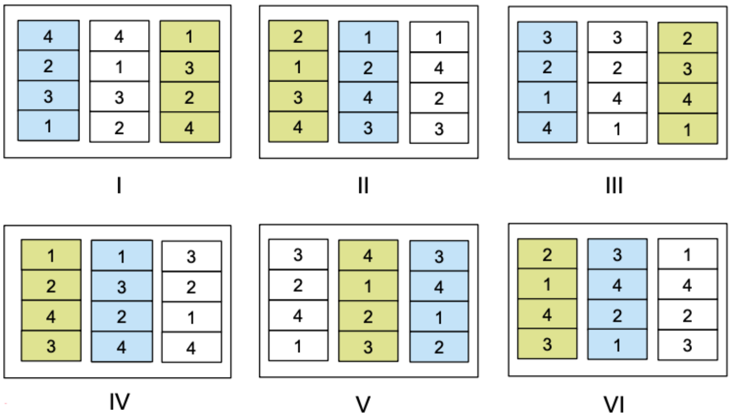
The price on the whole-plot level is less precision (or less power) because we have much less observations on this level.

8.5 Split-Split Plot Design: Oats example:

If we have more than two factors, a split-split plot design can be performed, where we have an additional "layer" and therefore three "sizes" of experimental units: Whole plots, split plots and split-split plots.

For example, consider the following experiment design:

- 6 different blocks (B)
- 3 different varieties (V)
- 4 different nitrogen treatments (N)
- Response (Y)



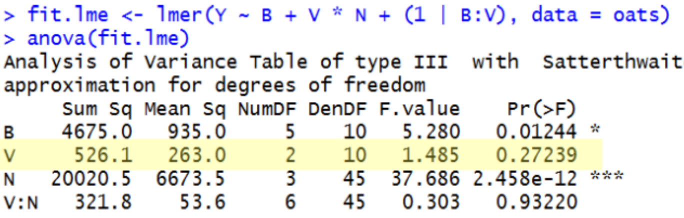
The varieties were applied to the main plots and the nitrogen treatments to the sub-plots. An experimental unit on the whole-plot level is given by the combination of block and variety.

A whole plot is given by a plot of land in a block (B), the whole-plot factor is variety (V). A block design (RCBD) was used at the whole-plot level. A split plot is given by a subplot of land, the split-plot factor is given by nitrogen treatment (N). The mathematical model is

$$Y_{ijk} = \mu + \alpha_i + \gamma_k + \eta_{ik} + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}$$

Where α_i is the fixed effect of variety, γ_k the fixed effect of block, η_{ik} the whole-plot error $\left(\mathcal{N}\left(0, \sigma_\eta^2\right)\right)$, β_j the fixed effect of nitrogen treatment, $(\alpha\beta)_{ij}$ the interaction between variety and nitrogen treatment and ϵ_{ijk} the split-plot error. In R, we would model this as

```
fit <- lmer(Y ~ B + V * N + (1 | B:V), data=oats)
```



```
> fit.lme <- lmer(Y ~ B + V * N + (1 | B:V), data = oats)
> anova(fit.lme)
Analysis of Variance Table of type III  with  Satterthwaite
approximation for degrees of freedom
     Sum Sq Mean Sq NumDF DenDF F.value    Pr(>F)    
B    4675.0   935.0     5    10   5.280   0.01244 *  
V     526.1   263.0     2    10   1.485   0.27239    
N   20020.5  6673.5     3    45  37.686 2.458e-12 ***
V:N   321.8    53.6     6    45   0.303   0.93220    
```

Observe that the test for variety uses 2 and 10 degrees of freedom, respectively. Indeed on the whole-plot level we have the following ANOVA table:

Source	df
Block	5
Variety	2
Error (whole-plot)	10(= 17 − 7)
Total	17(= 18 − 1)

Think of "averaging away" the nitrogen factor, hence we have one observation per combination of block and variety. Technically speaking, variety is tested against the interaction of block and variety.

We test everything involving the split-plot factor against the residual error, which has 45 df's. Hence, the main effect of the whole-plot factor is estimated less precisely, and the test is less powerful (compared to the split-plot level).

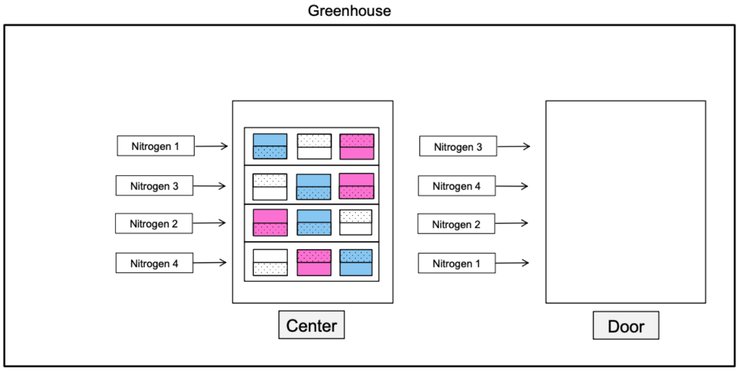
8.5.1 Example: Weed Biomass in Wetlands:

The experiment studied the effect of

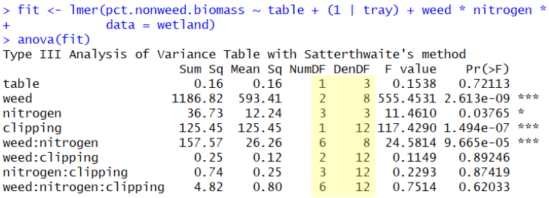
- nitrogen (4 levels of nitrogen)
- weed (3 levels)
- clipping treatments (2 levels: clipping / no clipping)

on plant growth in wetlands. The experiment was performed as follows:

- 8 trays, whereof each holds 3 artificial wetlands (rectangular wire baskets)
  - 4 of the trays were placed on a table near the door of the greenhouse,
  - 4 of the trays were placed on a table in the center of the greenhouse.
- On each table, we randomly assign one of the trays to each of the 4 nitrogen treatments.
- Within each tray, we randomly assign the 3 weed treatments.
- In addition, each wetland is split in half. One half is chosen at random and will be clipped, the other half is not clipped.
- After 8 weeks: Measure fraction of biomass that is non-weed.



Position in the greenhouse is a block factor (with levels center / door). Trays are whole plots, and nitrogen level is the whole-plot factor. Wetlands are split plots and weed treatment is the split-plot factor. Wetland halves are so-called split-split plots and clipping is the split-split-plot factor.
We use the following model


```
> fit <- lmer(pct.nonweed.biomass ~ table + (1 | tray) + weed * nitrogen * clipping + (1 | wetland),
+             data = wetland)
> anova(fit)
Type III Analysis of Variance Table with Satterthwaite's method
                       Sum Sq Mean Sq NumDF DenDF  F value    Pr(>F)    
table                    0.16    0.16     1     3   0.1538   0.72113    
weed                  1186.82  593.41     2     8 555.4531 2.613e-09 ***
nitrogen                36.73   12.24     3     3  11.4610   0.03765 *  
clipping               125.45  125.45     1    12 117.4290 1.494e-07 ***
weed:nitrogen          157.57   26.26     6     8  24.5814 9.665e-05 ***
weed:clipping            0.25    0.12     2    12   0.1149   0.89246    
nitrogen:clipping        0.74    0.25     3    12   0.2293   0.87419    
weed:nitrogen:clipping   4.82    0.80     6    12   0.7514   0.62033    
```

Note that all main effects and the nitrogen×weed interaction are significant.

We are here performing three experiments in one. On the whole-plot level we have the "experiment":

Source	df
Table (block)	1
Nitrogen	3
Error (per tray)	3(= 7 − 4)
Total	7(= 8 − 1)

On the split-plot level we have the "experiment":

Source	df
Tray (block)	7
Weed	2
Weed × Nitrogen	6
Error (per wetland)	**8**(= 23 − 15)
Total	23(= 24 − 1)

On the split-split-plot level we have the "experiment":

Source	df
Wetland (block)	23
Clipping	1
Weed × Clipping	2
Nitrogen × Clipping	3
Nitrogen × Weed × Clipping	6
Error (per wetland half)	**12**(= 47 − 35)
Total	47(= 48 − 1)

8.5.2 Example: Hardening temperature:

A material scientist wanted to analyze the influence of hardening temperature and type on breaking strength of different metal alloys using a split-plot design. Four ovens were available for experimen- tation, each of which had three trays. He chose the temperatures 675 ◦F, 700 ◦F, 725 ◦F and 750 ◦F. Each day, he randomly assigned the four temperatures to the four ovens and the three types of al- loys to the three oven trays. This is a block design with three blocks (days).

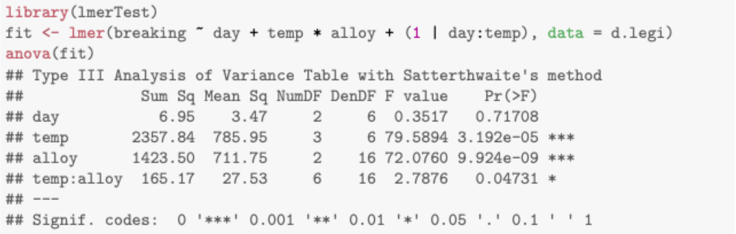
Each day, the four temperature levels have been randomly assigned to the four ovens. Hence, ovens are the whole plots (= experimental units for the whole-plot factor temperature). Per oven, the three alloys have been randomly assigned to the three oven trays. Hence, oven trays are split plots (= the experimental units for the split-plot factor alloy).

Note: all factors are crossed with each other.
The model is

$$Y_{ijkl} = \mu + \gamma_i + \alpha_j + \eta_{l(ij)} + \beta_k + (\alpha\beta)_{jk} + \varepsilon_{l(ijk)},$$

where $l = 1$ (there are no whole-plot replicates per block)

Y_{ijkl}	breaking strength (random)
μ	overall mean (fixed)
γ_i	effect of day (block effect), $i = 1, 2, 3$ (fixed, but can also be regarded as random)
α_j	main effect of temperature, $j = 1, 2, 3, 4$ (fixed)
β_k	main effect of alloy, $k = 1, 2, 3$ (fixed)
$(\alpha\beta)_{jk}$	interaction of temperature and alloy (fixed)
$\eta_{l(ij)}$	whole-plot error (random): deviation in the oven with the j-th temperature on the i-th day
$\varepsilon_{l(ijk)}$	split-plot error (random)


```
library(lmerTest)
fit <- lmer(breaking ~ day + temp * alloy + (1 | day:temp), data = d.legi)
anova(fit)
## Type III Analysis of Variance Table with Satterthwaite's method
##              Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)    
## day            6.95    3.47     2     6  0.3517   0.71708    
## temp        2357.84  785.95     3     6 79.5894 3.192e-05 ***
## alloy       1423.50  711.75     2    16 72.0760 9.924e-09 ***
## temp:alloy   165.17   27.53     6    16  2.7876   0.04731 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

8.5.3 Mixed Example: Pizza optimization:

Three new types of pizzas in six different packings are investigated by 90 consumers on a 0-10 scale.

1) **Each person rates the six packings of just one type of pizza**, that is pizzas are randomized to persons and each person tastes the different packings in random order.

This is a split-plot design with persons as whole plots and rating orders (or time slots) as split plots. Pizza type is the whole-plot factor, packing the split-plot factor. We have:

$$Y_{ijk} = \mu + \beta_i + \eta_{k(i)} + \alpha_j + (\alpha\beta)_{ij} + \epsilon_{k(ij)}$$

Where $\eta_{k(i)}$ is the whole-plot error (per person). The ANOVA skeleton is given by:

Plot level	Source	df	MS	F
Whole plots	pizza	2	MS_B	$\frac{MS_B}{MS_\eta}$
	residual	87	MS_η	
Split plots	packing	5	MS_A	$\frac{MS_A}{MS_E}$
	packing:pizza	10	MS_{AB}	$\frac{MS_{AB}}{MS_E}$
	residual	435	MS_E	
	total	539		

2) **Each person rates exactly one pizza in one packing, where the combination of pizza and packing is randomized to persons.**

This is a completely randomized factorial design (two-way ANOVA) with the model

$$Y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}$$

and the ANOVA skeleton

Source	df	MS	F
A	5	MS_A	MS_A/MS_E
B	2	MS_B	MS_B/MS_E
AB	10	MS_{AB}	$\text{MS}_{AB}/\text{MS}_E$
Residual	72	MS_E	
Total	89		

3) **Each person rates every pizza in every packing, where the combination of pizza and packing is randomized within person (with respect to tasting order).**

Here we have a randomized complete block design with persons as blocks (γ_i), where the model is given by

$$Y_{ijk} = \mu + \gamma_i + \alpha_j + \beta_k + (\alpha\beta)_{jk} + \epsilon_{ijk}$$

and the skeleton by

Source	df	MS	F
Blocks	89	$\text{MS}_{\text{blocks}}$	
A	5	MS_A	MS_A/MS_E
B	2	MS_B	MS_B/MS_E
AB	10	MS_{AB}	$\text{MS}_{AB}/\text{MS}_E$
Residual	1513	MS_E	
Total	1619		

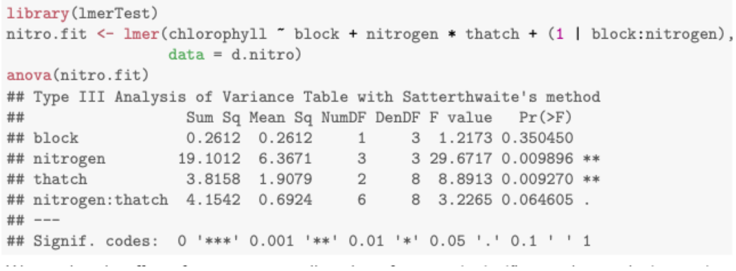
8.5.4 Example: Nitrogen in chemical form:

A soil scientist wanted to investigate the effects of nitrogen supplied in four different forms and later evaluate those effects combined with those of thatch accumulation (two, five or eight years of accumulation) on the quality of an established turf. A golf green had been constructed and seeded with grass on the experimental plots. The nitrogen treatment plots were arranged on the golf green in a randomized complete block design with two block levels. Each of the eight experimental plots was split into three subplots to which the levels of the second treatment factor were randomly assigned.

This is a split-plot design with whole-plot factor nitrogen, split-plot factor thatch and a block factor block.

$$Y_{ijkl} = \mu + \gamma_i + \alpha_j + \beta_k + (\alpha\beta)_{jk} + \eta_{l(ij)} + \epsilon_{l(ijk)}$$

Where $l = 1, \gamma_i$ fixed effect of block $(i = 1, 2), \alpha_j$ fixed main effect of nitrogen $(j = 1, 2, 3, 4), \beta_k$ fixed main effect of thatch $(k = 1, 2, 3), (\alpha\beta)_{jk}$ interaction, $\eta_{l(ij)}$ the error on the whole-plot level and $\epsilon_{l(ijk)}$ the error on the split-plot level.



```
library(lmerTest)
nitro.fit <- lmer(chlorophyll ~ block + nitrogen * thatch + (1 | block:nitrogen),
                  data = d.nitro)
anova(nitro.fit)
## Type III Analysis of Variance Table with Satterthwaite's method
##                 Sum Sq Mean Sq NumDF DenDF F value   Pr(>F)
## block           0.2612  0.2612     1     3  1.2173 0.350450
## nitrogen       19.1012  6.3671     3     3 29.6717 0.009896 **
## thatch          3.8158  1.9079     2     8  8.8913 0.009270 **
## nitrogen:thatch 4.1542  0.6924     6     8  3.2265 0.064605 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

9 Incomplete Block Designs

The block designs we have seen so far were complete, meaning that every block contained all treatments. In practice, this is not always possible. For example, the physical size of a block might be too small. There are also situations where it is not advisable to have too many treatments in each block. If we do a food tasting experiment, we typically want to restrict the number of different "recipes" (treatments) we want to show to an individual rater. In an incomplete block design (IBD), we have to decide what subset of treatments we use on an individual block.

We call a design disconnected if we can build two groups of treatments such that it never happens that we see members of both groups together in the same block. In such designs, certain quantities can't be estimated (e.g. comparing treatment in different groups). If we fit a standard main effect model, two of the treatment coefficients will be set to zero (not only one as usual when using *contr.treatment* and looking at the coefficients with *dummy.coef*), e.g.:

```
## (Intercept):        0.307025
## block:              1            2            ...
##                     0.000000     -0.2788532   ...
## treat:              A            B            ...
##                     0.000000     0.99894512   ...
```

R automatically drops one of the treatment levels out of the model (which might be dangerous if you are not carefully inspecting the output) in these cases.

9.1 Balanced Incomplete Block Designs:

A possible optimality criterion is that we can estimate all treatment differences with the same precision, i.e. all confidence intervals for $\alpha_i - \alpha_j$ have the same width (for any pair i, j). This is fulfilled by a balanced incomplete block design (BIBD). A BIBD is an incomplete block design where all pairs of treatment occur together in the same block equally often ($= \lambda$). The following notation is used:

- g: Number of treatments ("number of different chocolate chip cookie brands")
- b: Number of blocks ("number of raters")
- k: Number of units per block ($k < g$) ("number of cookie brands each raters gets to see")
- r: Number of replicates per treatment ("how often do we see a certain cookie brand across all raters?")
- N: Total number of units

We have $N = b * k = g * r$. For every setting $k < g$, it's possible to find a BIBD by taking all possible $\binom{g}{k}$ (choose in R) subsets. The corresponding design is called an unreduced balanced incomplete block design. In R, the combinations are retrieved with combn, e.g. combn$(x = 4, m = 3)$ which would generate 4 blocks because $\binom{4}{3} = 4$ which would give the following outputs (where the columns are blocks):

```
##         [,1] [,2] [,3] [,4]
## [1,]    1    1    1    2
## [2,]    2    2    3    3
## [3,]    3    4    4    4
```

Typically, we would randomize the order or the placement of the treatments within a block. In practice we cannot always afford to do an unreduced BIBD as the required number of blocks might be too large. Whether a BIBD exists for a certain desired setting of number of blocks b, block size k and number of treatments g, is a combinatorial problem. A necessary (but not sufficient, i.e. even if the condition is fulfilled, it might be the case that you can't find a BIBD) condition for a BIBD to exist is

$$\frac{r \cdot (k-1)}{g-1} = \lambda$$

where λ is the number of occurence of two factors toghether in the same block among all blocks.
The intuition behind the formula is that a treatment appears in r different blocks. In each block, there are $k-1$ available other units, leading to a total of $r*(k-1)$ available units. There are $g-1$ available treatments that must be divided among them in order to have a balanced design. Hence, $r \cdot (k-1) = \lambda \cdot (g-1)$ must hold. In R, the package *ibd* provides some functionality to find (B)IBDs. E.g. the function *bibd* can be used to find a BIBD:

```
des.bibd <- bibd(v = 6, b = 10,
                 r = 5, k = 3, lambda = 2)
```

where v is the number of treatments and the rest as defined above. In *des.bibd\$design*, each row corresponds to a block:

```
##         [,1] [,2] [,3]
##   [1,]    4    5    6
##   [2,]    1    4    5
##   [3,]    1    3    4
##   [4,]    1    3    6
##   [5,]    3    5    6
##   [6,]    1    2    6
##   [7,]    2    4    6
##   [8,]    2    3    5
##   [9,]    1    2    5
## [10,]    2    3    4
```

With *des.bibd\$NNP*, we can view the concurrency matrix. When no BIBD is possible, ibd can be used to find a nearly balanced design: *des.ibd ¡- ibd(v = 6, b = 9, k = 3)* In the concurrency matrix (*des.ibd\$conc.mat*), we see how often any pair of treatments appear together in the same block, e.g.:

```
##         [,1] [,2] [,3] [,4]
## [1,]    1    0    1    0
## [2,]    0    1    1    0
## [3,]    1    1    3    1
## [4,]    0    0    1    1
```

isGYD(d) can be used to check if an incomplete design is balanced, where d is a matrix with blocks as rows (i.e. the same output format as *des.bibd*) In a partially balanced incomplete block design, some treatment pairs occur together more often than other pairs. **Row-Column Incomplete Block Designs** In these designs, we have two block factors (rows and columns) and one or both of them are incomplete blocks.

Youden Squares A Youden Square is rectangular such that the columns (or rows; can be flipped) form a BIBD and for the rows (or columns), each treatment appears equally often in each row (column). The columns therefore form a BIBD, the rows an RCB.

9.2 Analysis:

The analysis of an incomplete block design is "as usual". We use a block factor and a treatment factor leading to

$$Y_{ij} = \mu + \alpha_i + \beta_j + \epsilon_{ij}.$$

As we are faced with an unbalanced design we typically use sum of squares for treatment effects that are adjusted for block effects (i.e. *drop*1), e.g.:

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

Where we would get the p-values for session and detergent (but would only be interested in detergent).

In this approach, we analyze the treatment effects while controlling for the block effects, which is a so called intrablock analysis of the (B)IBD. It is also possible to recover some information by comparing different blocks, which would be called an interblock analysis. If we don't use adjusted sum of squares (*drop*1), it's important to first list the block factor in the model and then the treatment, as we're interested in seeing if the treatment has any influence on the response, after controlling for the variation between blocks.

10 Power

A statistical test controls by construction the type I error rate with the significance level α. This means the probability that we falsely reject the null hypothesis H_0 is less than α. The type II error occurs if we fail to reject the null hypothesis even though the alternative hypothesis H_A holds. The probability of a type II error is denoted by β (and we aren't controlling it). There is no universal β, it depends on the specific alternative H_A that is assumed. The power of a statistical test is

$$P\,(\text{ reject } H_0 \mid \text{ a certain setting in } H_A \text{ holds }) = 1 - \beta.$$

For calculating power, no data is needed but a precise specification of the parameter setting under the alternative that we believe in ("what would happen if..."). If we plan an experiment with low power, it means that we waste time and money because with high probability we are not getting a significant result. A "rule of thumb" is that power should be larger than 80%. Power depends on

- Design of the experiment (balanced, unbalanced, without blocking, with blocking, ...)
- Significance level α
- Parameter setting under the alternative (incl. error variance σ^2
- Sample size n

We can mainly maximize power using the first (design) and last item (sample size) from above.

For some designs (like a completely randomized design), there are closed-form formulas to calculate power. When the design is getting more complex, simulations are usually used.

For example, in a simple one-way ANOVA model with five groups and the null hypothesis:

$$H_0 : \mu_1 = \ldots = \mu_5$$

And the alternative hypothesis:

$$H_A : \mu_k \neq \mu_l \text{ for at least one pair } k \neq l$$

One possible assumption for the alternative would be:

$$\mu_1 = 57, \mu_2 = 63, \mu_3 = 60, \mu_4 = 60, \mu_5 = 60.$$

In addition, the error variance has to be specified, which is assumed to be $\sigma^2 = 7.5$. For this design, the *power.anova.test* function can be used. It needs the number of groups, the variance between the group means μ_i, the error variance σ^2 and the sample size within each group (n). It is then called like this

```
mu     <- c(57, 63, rep(60, 3))
sigma2 <- 7.5
power.anova.test(groups = length(mu), n = 4,
    between.var = var(mu), within.var = sigma2)
```

And a possible output would be:

```
##  power = 0.5452079
```

Which would mean we have a 54% chance to get a significant result under the above setting. We can also leave away n and use the argument *power* to get the required sample size (per group) for a certain power.

We can also simply simulate the data many times, do the corresponding test and measure the proportion of simulation runs that rejected H_0 (which is then the power). Because this is simply a binomial distribution (sum of independent Bernoulli trials), the function $binom.test(x, n)$ can be used to get a confidence interval on the estimated power, where x is the number of times H_0 was rejected (the number of "successes") and n the number of overall simulations. If this confidence interval is too wide, n should be increased, i.e. one should do more simulation runs. For simulation, the function

```
rnorm(n * g, mean = rep(means, each = n), sd = sigma)
```

can be helpful, which (in the above case) generates a vector of normally distributed variables with mean *mean* and standard deviation *sigma*.

11 Notation

$$y_{i\cdot} = \sum_{j=1}^{n_i} y_{ij} \qquad \text{sum of group } i$$

$$y_{\cdot\cdot} = \sum_{i=1}^{g} \sum_{j=1}^{n_i} y_{ij} \qquad \text{sum of all observations}$$

$$\bar{y}_{i\cdot} = \frac{1}{n_i} \sum_{j=1}^{n_i} y_{ij} \qquad \text{mean of group } i$$

$$\bar{y}_{\cdot\cdot} = \frac{1}{N} \sum_{i=1}^{g} \sum_{j=1}^{n_i} y_{ij} \qquad \text{overall (or total) mean}$$

12 Design Examples

12.1 Latin Square Design Example:

	C_1	C_2	C_3	C_4
R_1	A	B	C	D
R_2	B	C	D	A
R_3	C	D	A	B
R_4	D	A	B	C

12.2 Graeco-Latin Square Design Example:

	C_1	C_2	C_3	C_4
R_1	$A\alpha$	$B\gamma$	$C\delta$	$D\beta$
R_2	$B\beta$	$A\delta$	$D\gamma$	$C\alpha$
R_3	$C\gamma$	$D\alpha$	$A\beta$	$B\delta$
R_4	$D\delta$	$C\beta$	$B\alpha$	$A\gamma$

12.3 Row-Column Incomplete Block Design:

	C_1	C_2	C_3	C_4	C_5	C_6	C_7
R_1	3	4	5	6	7	1	2
R_2	5	6	7	1	2	3	4
R_3	6	7	1	2	3	4	5
R_4	7	1	2	3	4	5	6

The rows are complete blocks (there are 7 treatments), the columns form a BIBD, which is a so called row-orthogonal design.

13 R Details

13.1 Factors:

Categorical variables in R are also called factors and the different values levels. With *str*, we can check the structure of data (if it's encoded as a factor). If it's not, we can use *factor*, e.g.:

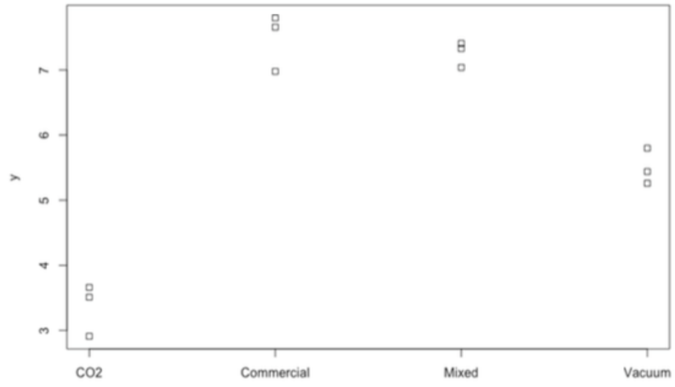
```
data$column = factor(data$column)
```

With *levels*, we can check the levels of a factor.

13.2 Visualization:

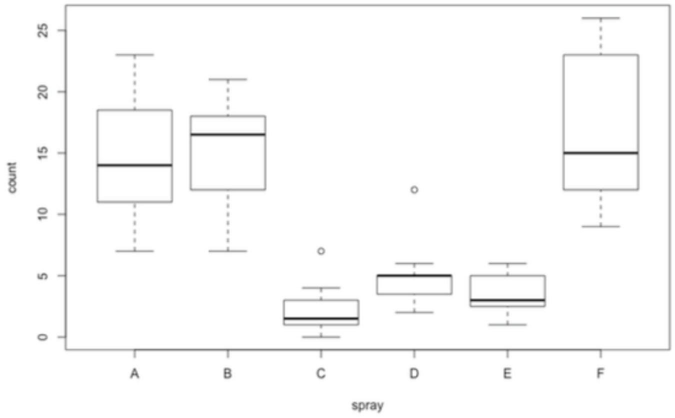
The *stripchart* function can be used to generate a stripchart

```
stripchart(y ~ treatment, data = meat,
    vertical = TRUE)
```



And *boxplot* for boxplots:

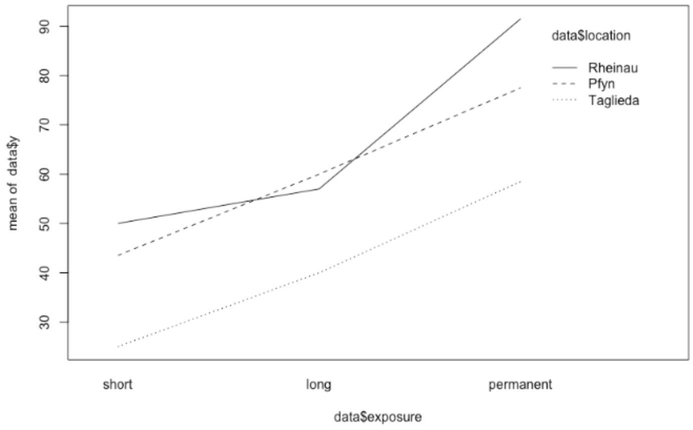
```
boxplot(count ~ spray, data = InsectSprays)
```



The thick line in the middle is the median value, the rectangle the interquartile range (IQR), i.e. the lower value is the 25th percentile and the higher value the 75th percentile. The whiskers (top / bottom lines) are usually bounded by 1.5 * IQR, values outside are displayed as outliers.

An interaction plot can be used to visualize interactions between two factors

```
with(data, interaction.plot(x.factor = exposure,
           trace.factor = location, response = y))
```

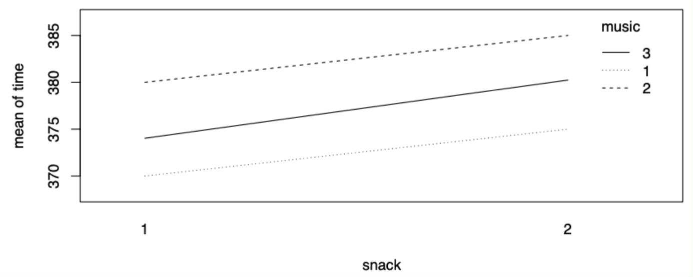


Example: Time and Music

Given balanced two-way ANOVA study with factors music (levels: 1, 2, 3) and snack (levels: 1,2). Response $time_{ijk}, i = 1, \ldots, 3, j = 1, 2, k = 1, \ldots, 5$

We can read off the average change in time spent working when switching from snack 1 to snack 2 while listening to music 3 . That is, we can read off the snack effect when listening to music 3 . This average change in time spent working can be calculated from the data as:

$$\frac{1}{K}\sum_{k=1}^{K} \text{time }_{32k} - \frac{1}{K}\sum_{k=1}^{K} \text{time }_{31k}$$



13.3 Data Generation:

Generate 10 "A"'s, followed by 10 "B"'s (two methods):

```
c(rep("A", times = 10), rep("B", times = 10))
rep(c("A", "B"), each = 10)
```

Alternate "A", "B" 10 times:

```
rep(c("A", "B"), times = 10)
```

Toss a coin 20 times (0.5 prob. for "A", "B"):

```
sample(c("A", "B"), 20, replace = TRUE)
```

Choose 10 "A" at random, the rest "B":

```
sample(rep(c("A", "B"),times = 10),20, replace =FALSE)
```

13.4 Calculations:

Overall mean of column "blood" (two methods):

```
mean(b.data$blood)
aggregate(blood ~ 1, data = b.data, mean)
```

Group means per treatment:

```
aggregate(blood ~ treat, data = b.data, mean)
```

13.5 Tests:

If we want to check if we can simultaneously drop A and B from the model:

```
fit.null <- aov(Y ~ 1, data = dat)
fit.main <- aov(Y ~ A + B, data = dat)
anova(fit.null, fit.main)
```

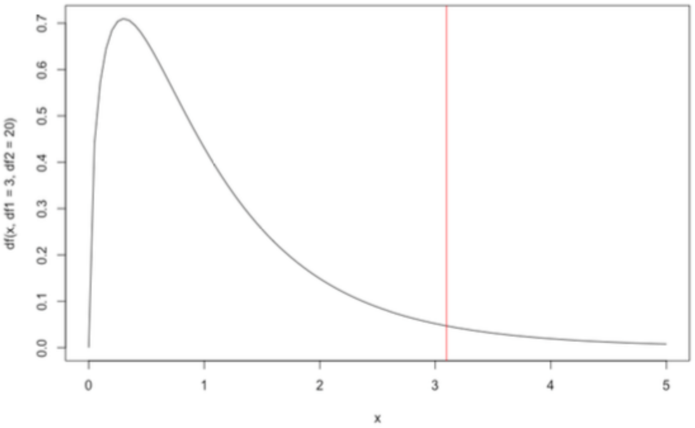
14 Various distributions

14.1 F Distribution:

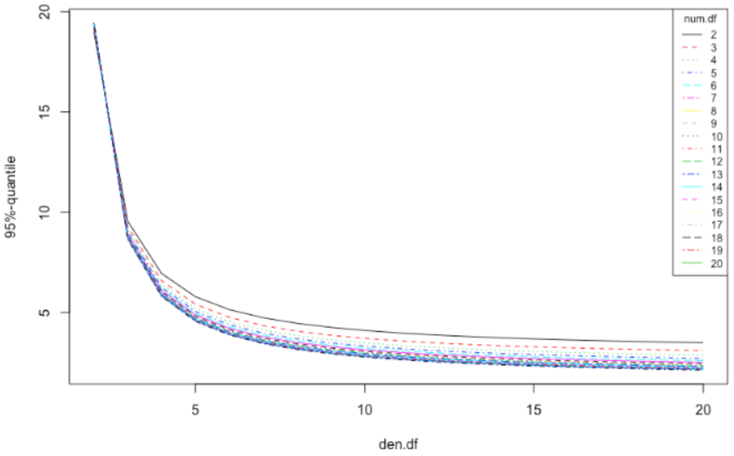
We have:

$$F_{n,m} = \frac{\frac{1}{n}\left(X_1^2 + \ldots + X_n^2\right)}{\frac{1}{m}\left(Y_1^2 + \ldots + Y_m^2\right)}$$

Note $N \uparrow$, F-quantile $\downarrow$
Where X_i, Y_j are i.i.d. $\mathcal{N}(0,1)$. The $F_{1,m}$-distribution is a special case, it's the square of a t_m-distribution. The F distribution with 3 denominator degrees of freedom and 20 numerator degrees of freedom looks like this (where the red line is the 95%-quantile):



The 95%-quantile behaves like this, depending on the denominator / numerator degrees of freedom:



The degrees of freedom of the numerator are determined by the number of levels of each factor. The degrees of freedom of the denominator are determined by the number of observations and the sum of the degrees of freedom for each factor.

14.2 Two Sample t-Test for Unpaired Data:

We have X_i i.i.d. $\sim \mathcal{N}\left(\mu_X, \sigma^2\right)$ and Y_j i.i.d. $\sim \mathcal{N}\left(\mu_Y, \sigma^2\right)$ with X_i, Y_j independent. For the t-Test, $H_0 : \mu_X = \mu_Y$ and $H_A : \mu_X \neq \mu_Y$ (or one-sided). Then:

$$T = \frac{\left(\bar{X}_n - \bar{Y}_m\right)}{s_{\text{pool}}\sqrt{\frac{1}{n} + \frac{1}{m}}} \sim t_{n+m-2} \text{ under } H_0$$

14.3 Two Sample t-Test for Paired Data:

We have independent $D_i = X_i - Y_i$ and:

$$\bar{D} = \frac{1}{n}\sum_{i=1}^{n} D_i \sim \mathcal{N}\left(\mu_D; \sigma_D/\sqrt{n}\right)$$

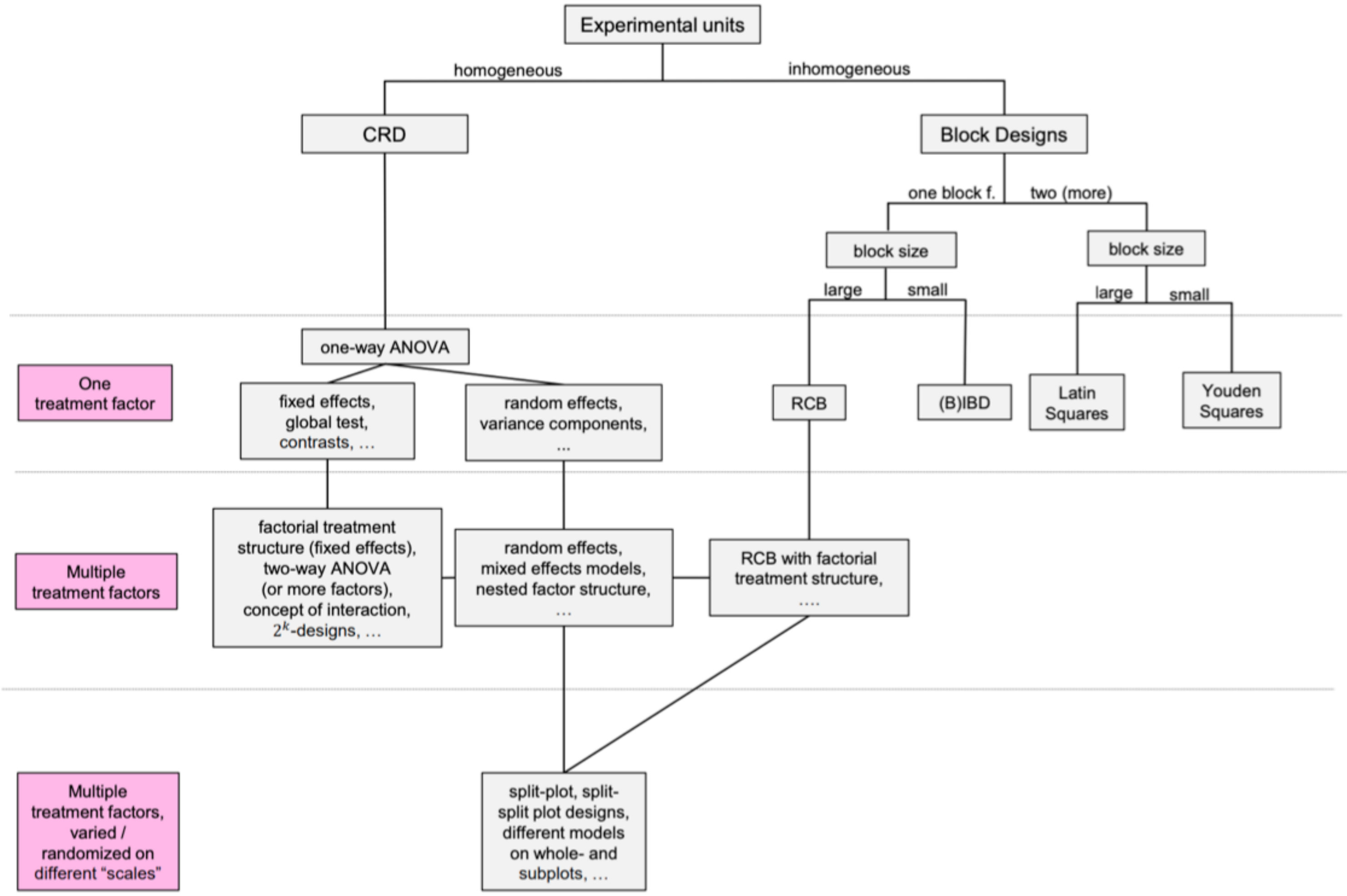
H_0/H_A as before and:

$$T = \sqrt{n}\frac{\bar{D}}{S_D} \sim t_{n-1} \text{ under } H_0$$

14.4 Binomial Coefficient:

$$\binom{n}{k} = \frac{n!}{k! * (n-k)!}$$

15 Appendix


Experimental units
homogeneous
inhomogeneous
CRD
Block Designs
one block f.
two (more)
block size
block size
large
small
large
small
One treatment factor
one-way ANOVA
fixed effects, global test, contrasts, ...
random effects, variance components, ...
RCB
(B)IBD
Latin Squares
Youden Squares
Multiple treatment factors
factorial treatment structure (fixed effects), two-way ANOVA (or more factors), concept of interaction, 2^k-designs, ...
random effects, mixed effects models, nested factor structure, ...
RCB with factorial treatment structure, ...
Multiple treatment factors, varied / randomized on different "scales"
split-plot, split-split plot designs, different models on whole- and subplots, ...

15.1 Three way ANOVA:

We can easily extend the model to more than two factors. If we have three factors A, B and C (with a, b and c levels, respectively), we have 3 main effects, $3 \cdot 2/2 = 3$ two-way interactions (the "usual" interaction so far) and one so-called three-way interaction. We omit the mathematical model formulation and work directly with the corresponding R code. In R, we would write y ∼ A⋆B⋆C for a three-way ANOVA model including all main effects, all two-way interactions and a three-way interaction. An equivalent formulation would be $y \sim A + B + C + A : B + A : C + B : C + A : B : C$. Main effects are interpreted as average effects, two-way interaction effects are interpreted as deviations from the main effects model, i.e., the correction for an effect that depends on the level of the other factor, and the three-way interaction is an adjustment of the two-way interaction depending on the third factor. Or in other words, if there is a three-way interaction it means that the effect of factor A depends on the level combination of the factors B and C, i.e., each level combination of B and C has its own effect of A. This typically makes interpretation difficult. $n =$ **replicates**

Source	df	F-ratio
A	$a-1$	$\frac{MS_A}{MS_E}$
B	$b-1$	$\frac{MS_B}{MS_E}$
C	$c-1$	$\frac{MS_C}{MS_E}$
AB	$(a-1)(b-1)$	$\frac{MS_{AB}}{MS_E}$
AC	$(a-1)(c-1)$	$\frac{MS_{AC}}{MS_E}$
BC	$(b-1)(c-1)$	$\frac{MS_{BC}}{MS_E}$
ABC	$(a-1)(b-1)(c-1)$	$\frac{MS_{ABC}}{MS_E}$
Error	$abc(n-1)$	

The model with a three-way interaction is flexible enough to model abc different cell means! We also observe that the three-way interaction typically has large degrees of freedom (meaning, it needs a lot of parameters!). We can only do statistical inference about the three-way interaction if we have multiple observations in the individual cells. If we would only have one observation in each cell ($n = 1$), we could still fit the model $y \sim A+B+C+A : B+A : C+B : C$. This means that we drop the highest-order interaction and "pool it" into the error term. This is quite a common strategy, especially if we have more than three factors. Most often, the effect size of the most complex interaction is assumed to be small or zero. Hence, the corresponding term can be dropped from the model to save degrees of freedom. Ideally, these decisions are made before looking at the data. Otherwise, dropping all the insignificant terms from the model and putting them into the error term will typically lead to biased results. This will make the remaining model terms look too significant, i.e., we declare too many effects as significant although they are actually not. This means that the type I error rate is not being controlled anymore. In addition, the corresponding confidence intervals will be too narrow.