Design and Analysis of Experiments

Lecture notes

Part I

Christoph Kopp

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

November 5, 2018

Preface

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

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

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

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

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

Contents

1 Definitions and first examples

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

1.1 Basic definitions

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

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

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

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

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

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

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

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

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

1.2 Treatment structure

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

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

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

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

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

Not all experiments need a control treatment.

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

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

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

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

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

1.3 Plot structure

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

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

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

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

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

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

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

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

1.4 The design

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

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

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

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

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

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

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

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

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

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

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

2 Completely randomized designs without treatment structure

2.1 Introduction

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

In a CRD, plots have no structure. Treatments may or may not be structured in a CRD. In this chapter, we also assume unstructured treatments.
In the context of statistical models, the outcome is often called the *dependent variable* and the variable defining the treatments is called the *independent variable*. The terms *dependent* and *independent* are not a property of the variables themselves, they indicate how we intend to use them in the analysis.
In completely randomized designs, observational units were randomly allocated to the treatments, so the only systematic influence on the outcome should be due to the treatment. It is then justified to speak of the *effect* of the treatment on the outcome.
In general, not all plots will show precisely the same response to a treatment.[1] There is always some variablility, such that often one has to be content if a treatment shows some desired effect *on average*.
The first aim of the analysis is usually to investigate whether the mean of the numeric variable differs significantly between the treatments and to quantify the difference. Sometimes also other measures of location such as the median are studied.

2.1.1 `R` formulae

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

2.1.2 The insect sprays data

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

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

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

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

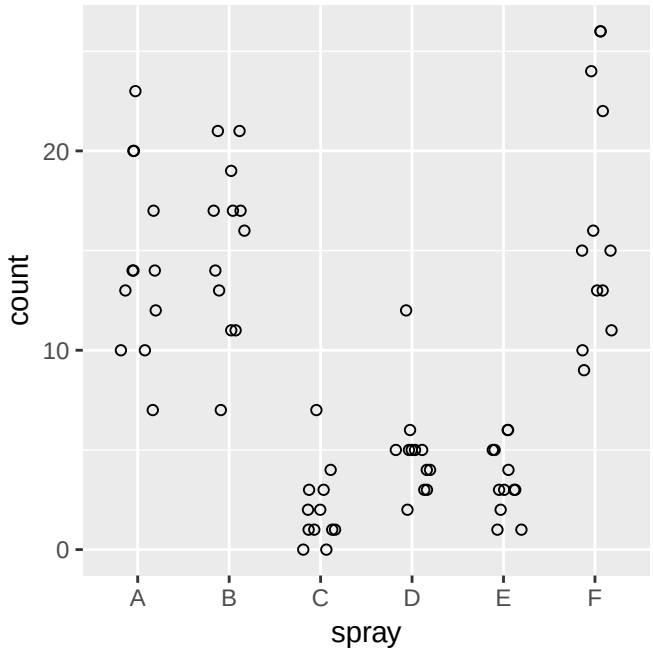
```
> with(InsectSprays, tapply(count, spray, length)) #12 obs. per trt
> with(InsectSprays, tapply(count, spray, summary)) #compare distr.
> with(InsectSprays, tapply(count, spray, sd)) #standard deviations
```

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

2.1.3 Visualization

To visualize the data, box plots or strip plots can be used.[3] Using `ggplot2`[4], a jittered strip plot can be produced by:

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

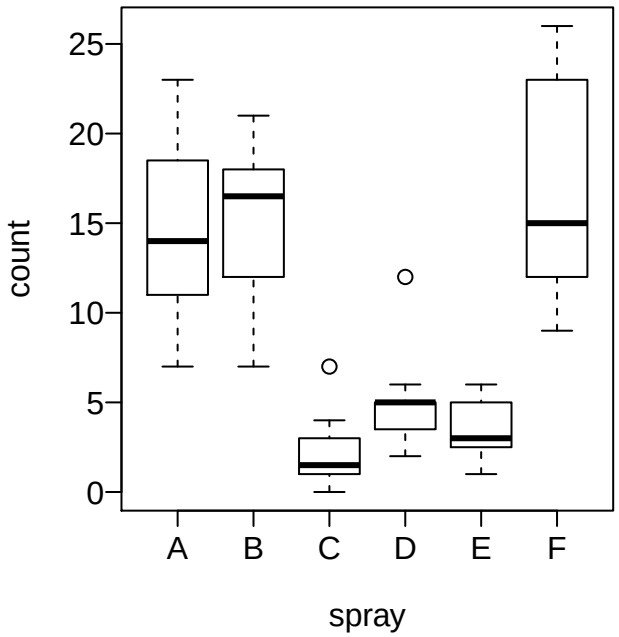


[3] In case less than 15 points per treatment are available, perhaps using a strip plot is better. Here, the `position` argument aids visualization by adding some random jitter in horizontal direction so that points do not overlap completely.

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

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

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



Look for

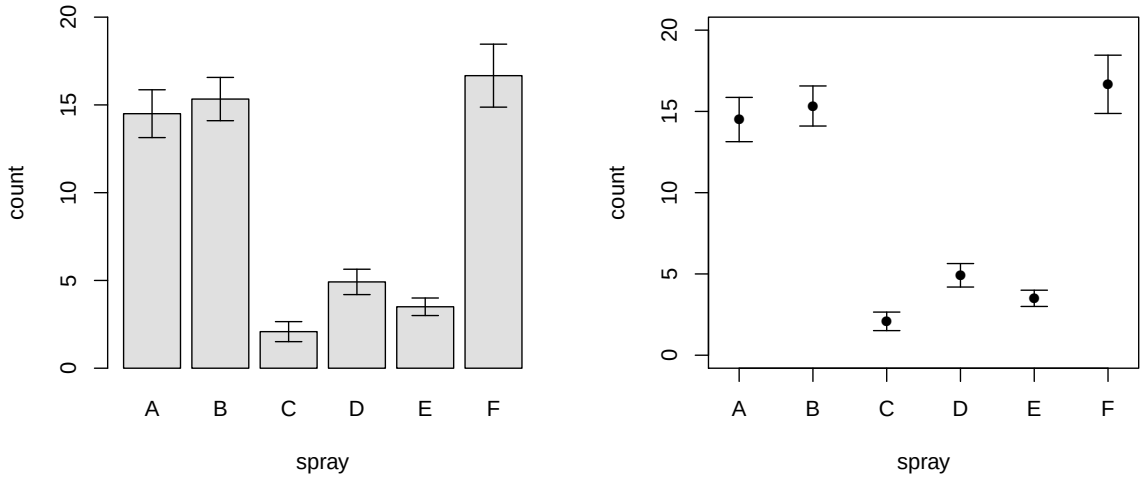
- location differences (compare the location of the bold lines representing the sample medians);
- variability differences (compare the box heights containing the central 50% of the data);
- skewness (look at the whiskers and at the location of the median in the box);
- outliers (look at single points).

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

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

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

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



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

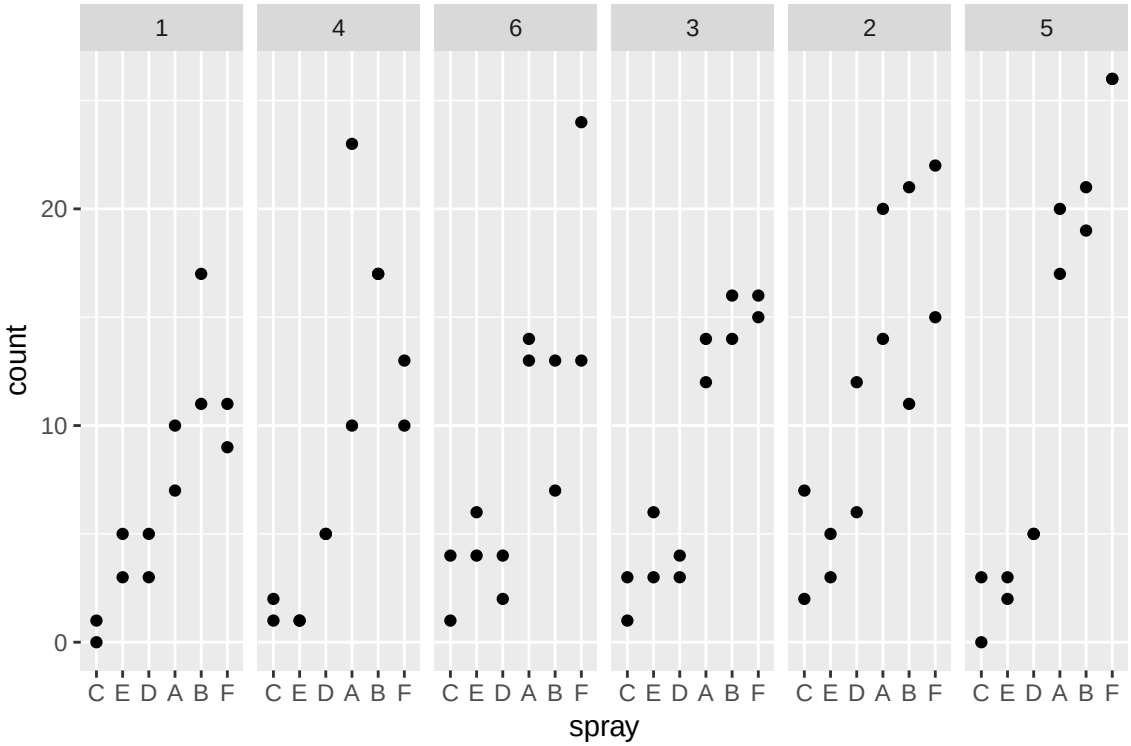
2.1.4 Block structure of the insect sprays data set

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

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

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

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



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

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

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

2.1.5 Notation and independence assumptions

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

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

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

If you are not willing to assume anything, no statistical analysis is possible because the model for the data is then too flexible. In other words, some things need to remain the same, such that you have more than one observation measured under comparable conditions and can start fitting a model.
We assume that observations are *mutually independent*. Remember that by definition, the random variables $Y_1, \ldots, Y_n$ are called mutually independent if for any $k \leq n$ and $y_1, \ldots, y_k \in \mathbb{R}$, we have that

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

For a simple example, set $k = 2$. Then, we require that for all (y_1, y_2), the probability of the event that $(Y_1 \leq y_1,\ Y_2 \leq y_2)$ must be the product of the probabilities of $(Y_1 \leq y_1)$ and $(Y_2 \leq y_2)$. In other words, Y_1 is not allowed to have any influence on Y_2, and vice versa.
It is impossible to establish independence with a statistical test, the assumption can only be justified with a good study design which produces observations that can be treated as independent.
We also assume that all the observations with the same treatment have the same distribution. If those two assumptions are not fulfilled (e. g. because of blocks), we need different models to account for it. The parametric and nonparametric methods discussed below differ with respect to the assumptions on the distribution of the outcome variable.

2.2 Parametric models: one-way ANOVA

2.2.1 Distributional assumptions

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

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

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

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

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

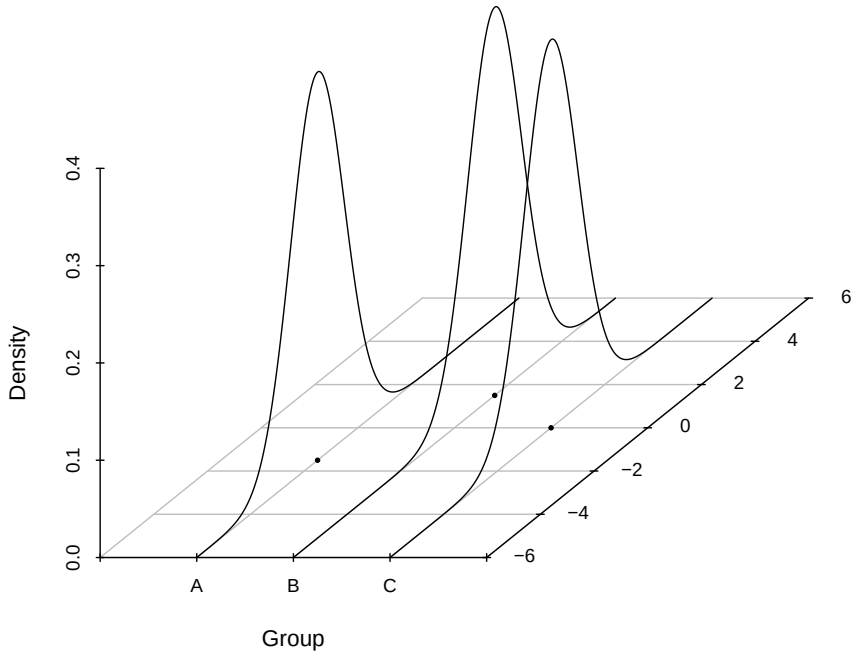


Figure 1: Visualizing the classical ANOVA assumptions

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

2.2.2 The ANOVA table and the overall F test

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

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

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

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

observations from that treatment only:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

2.2.3 Residuals, RMSE, and the R^2

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

The *residuals* are defined as

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

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

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

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

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

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

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

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

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

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

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

# [1] 3.921902
```

Divide (1) by SST, then

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

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

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

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

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

Technical remark

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

2.2.4 Estimating and testing group means

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

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

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

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

2.3 Pairwise treatment comparisons

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

2.3.1 The multiple testing problem

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

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

2.3.2 Directly adjusting a set of p values

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

2.3.3 Pairwise t and Wilcoxon tests

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

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

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

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

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

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

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

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

2.3.4 Tukey's honest significant difference (HSD)

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

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

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

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

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

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

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

  count groups
F 16.67      a
B 15.33      a
```

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

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

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

3 Completely randomized designs with control

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

3.1 Comparing all treatments with the control

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

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

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

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

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

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

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

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

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

```
> contrasts(InsectSprays$spray)

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

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

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

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

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

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

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

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

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

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

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

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

3.2 More about contrasts

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

```
> options("contrasts")

# $contrasts
# [1] "contr.treatment" "contr.poly"
```

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

3.2.1 Sum contrasts

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

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

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

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

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

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

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

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

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

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

Because we changed the contrasts, the interpretation of the coefficients changes. The `Intercept` is now an estimate of the overall mean (over all the treatments). On average (over all the treatments), 9.5 insects survive. With Spray A, 5.0 more insects survive

than the grand mean, etc. Spray E kills 6 more insects than the grand mean. Spray F is the reference and accordingly suppressed by `R` here (but see below). Other types of contrasts often used for ordinal factors are *Helmert* contrasts and *orthogonal polynomial* contrasts. We omit these here.

3.3 Specific comparisons

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

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

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

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

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

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

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

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

Compared to the `summary` in Section 3.1, the estimates and their standard errors are the same, but due to the adjustment for the multiple testing, the p values obtained now are higher. It is possible to disable the p value correction.
To test one or several specific hypotheses, the easiest way is to give a contrast matrix in which each row encodes one hypothesis. For example, we could ask whether the difference of treatments B and F to treatment A is the same, $\mu_B - \mu_A = \mu_F - \mu_A$. This amounts to fitting the model with treatment contrasts and leaving A as the reference level for the spray. Then, we simply need to test $\beta_B - \beta_F = 0$. We rewrite this such that each parameter gets a weight:

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

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

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

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

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

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

#
#   Simultaneous Tests for General Linear Hypotheses
#
# Multiple Comparisons of Means: GrandMean Contrasts
#
#
# Fit: lm(formula = count ~ spray, data = InsectSprays)
#
# Linear Hypotheses:
#          Estimate Std. Error t value Pr(>|t|)
# C 1 == 0     5.00       1.03    4.84  < 1e-04 ***
# C 2 == 0     5.83       1.03    5.64  < 1e-04 ***
# C 3 == 0    -7.42       1.03   -7.18  < 1e-04 ***
# C 4 == 0    -4.58       1.03   -4.43  0.00022 ***
# C 5 == 0    -6.00       1.03   -5.81  < 1e-04 ***
# C 6 == 0     7.17       1.03    6.93  < 1e-04 ***
# ---
# Signif. codes:
# 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# (Adjusted p values reported -- single-step method)
```

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

4 Model diagnostics and alternative analysis methods

4.1 The normality assumption

4.1.1 Graphical normality checking

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

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

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

The output is given in Figure 2.[13]

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

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

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

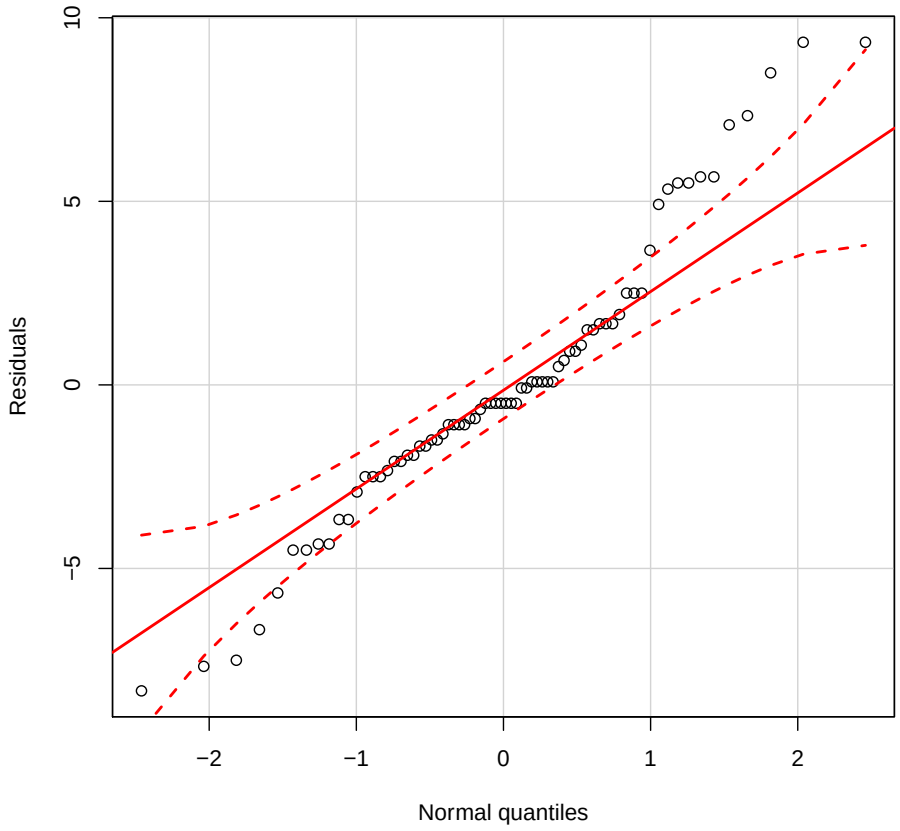


Figure 2: The QQ plot of the insect sprays ANOVA residuals

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

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

4.1.2 Testing for normality

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

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

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

yields a p value of 0.022.

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

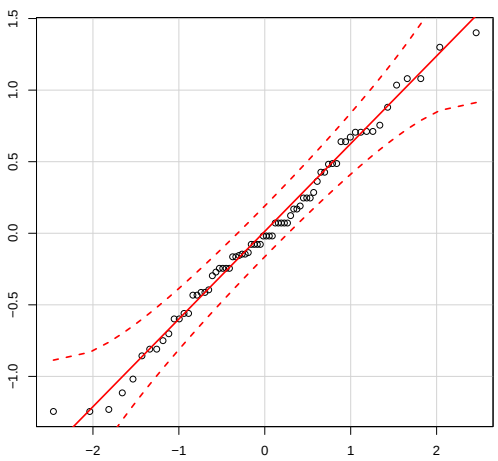
We also know what the problem is: the tails are too heavy.
It is very difficult to detect non-normality in small samples (any normality test has low power against many alternatives in small samples), so that some statisticians discourage the use of normality tests altogether. They argue that the danger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

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

4.1.3 Accounting for non-normality

We have at least three options now.

- The *transformation approach*:



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

The idea is to apply ANOVA to the transformed data $g(Y)$ and transform back

for interpretation. Any tried transformations should be stated in the final report. Incidentally, Beall 1942 used the insect sprays data set to illustrate a family of transformations.

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

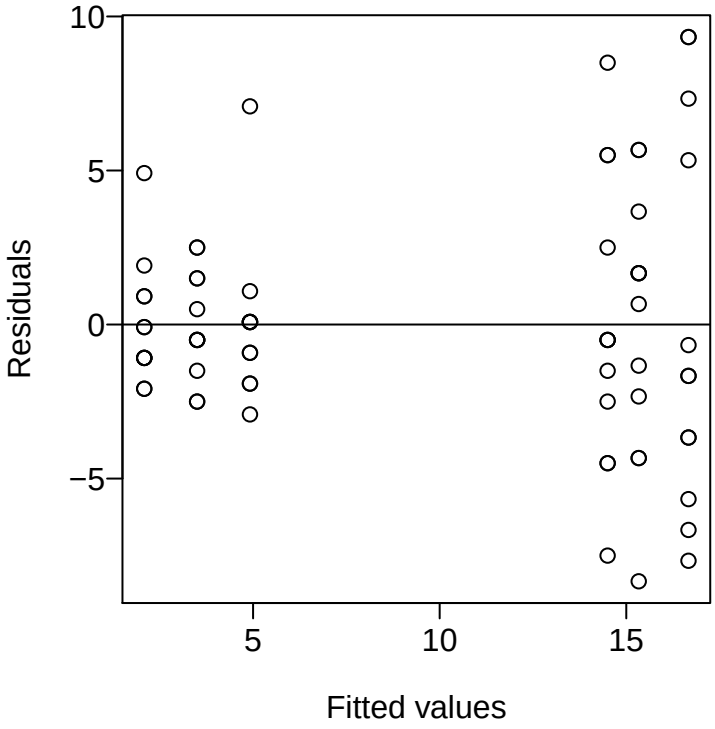
4.2 The homoskedasticity assumption

4.2.1 Residual plots

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

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

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



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

4.2.2 Testing homoskedasticity (equal variances)

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

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

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

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

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

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

4.2.3 Accounting for heteroskedasticity

We have three main alternatives.

1. Find a transformation g such that the variances of $g(Y)$ are more homogeneous. Sometimes a good g fixes both the normality and the variance problems.
2. Use robust methods, see Section 4.5.
3. Perform a one-way ANOVA with adjusted degrees of freedom. We show this next.

By calling

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

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

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

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

4.3 Consequences of model assumptions not holding

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

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

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

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

In large samples, the situation is generally better.

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

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

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

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

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

4.4 Nonparametric methods

4.4.1 Assumptions

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

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

1. The n random variables $\{Y_{ji}\}$ are mutually independent,
2. all observations $Y_{j1}, Y_{j2}, \cdots, Y_{jn_j}$ from treatment j come from the same continuous distribution F_j, for all $j \leq k$, and that
3. the treatment distribution functions F_j are *shifted* variants of some distribution F, i. e. that for all $t \in \mathbb{R}$
$$F_j(t) = F(t - \tau_j)$$
where τ_j is the shift (*effect*) for treatment j.

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

4.4.2 The Kruskal-Wallis test

The Kruskal-Wallis test is used to test the null hypothesis

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

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

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

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

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

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

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

4.5 Robust methods

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

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

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

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

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

[17] The situation is similar to Tukeys HSD improving upon pairwise t tests.
[18] Roughly a half hour on my 2009 MacBook Pro, some other software running.

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

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

The null hypothesis is rejected, and we conclude that the trimmed means differ between the insecticides. See Wilcox 2012 for further information.
Although nonparametric and robust methods get little room in this script, they are important for applications, because it is desirable to use statistical methods that do not rely on questionable assumptions.
A number of articles uses simulation (Monte Carlo) methods to study the consequences of non-normality and heteroskedasticity in the one-way setting. See e. g. Cribbie et al. 2007 for an overview. A good overview of robust methods is given in Wilcox 2012, see also the `R` vignette for robust methods.

4.6 Permutation methods

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

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

Bars separate the treatments. To test the null hypothesis that the three treatment means are the same, calculate the F statistic. (Permutation methods are also applicable to any other statistics, we just chose the F statistic to have something tangible.)
Now consider the following rearrangement (called a *permutation*) of the data:

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

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

1. Compute F for the original data.
2. Generate either all or a large number of randomly chosen permutations $\pi_b Y$ of the data. Call the total number of permutations B.
3. For each permutation $b = 1, \ldots, B$, calculate the F statistic for the permuted data $\pi_b Y$ and call it F_b.
4. Compute the p value
$$p = \frac{\#\{b : F_b \geq F\}}{B}$$

as the proportion of permutation test statistics F_b which are at least as extreme as the F statistic of the observed data.[19]
The reasoning is that if the null hypothesis is false, then the observed data should generate a large value of F, and only a small fraction of the F_b statistics obtained from the permuted data should be at least as extreme by chance.
This approach to hypothesis testing is called a *permutation test*. By simulation studies, it can be shown that permutation tests perform quite well for a large number of settings. They can be very flexibly adapted to a much wider set of test problems than ANOVA; one main disadvantage is that permutation tests only provide p values, but do not directly measure effect sizes.
In `R`, for example the libraries `lmPerm` and `coin` can perform the permutation test for one-way ANOVA; see the help files and vignettes for these two packages for further information.
Before running the analysis, let us think about the number of permutations briefly. There are $72! \approx 6 \cdot 10^{103}$ permutations of the insect sprays data. This astronomical number defies any modern computer. The packages implement different stopping rules to decide what constitutes a sufficiently big number B of randomly chosen permutations. Here, we show the use of `lmp` from `lmPerm`; `lmp` can be used like `lm`:

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

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

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

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

4.7 Exercises

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

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

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

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

Please follow the information provided on Moodle.