11

Analysis of Variance

Analysis of variance is the technique we use when all the explanatory variables are categorical. The explanatory variables are called **factors**, and each factor has two or more **levels**. When there is a single factor with three or more levels we use one-way ANOVA. If we had a single factor with just two levels, we would use Student's *t* test (see p. 294), and this would give us exactly the same answer that we would have obtained by ANOVA (remember the rule that $F = t^2$). Where there are two or more factors, then we use two-way or three-way ANOVA, depending on the number of explanatory variables. When there is replication at each combination of levels in a multi-way ANOVA, the experiment is called a **factorial** design, and this allows us to study **interactions** between variables, in which we test whether the response to one factor depends on the level of another factor.

One-Way ANOVA

There is a real paradox about analysis of variance, which often stands in the way of a clear understanding of exactly what is going on. The idea of analysis of variance is to compare two or more means, but it does this by comparing variances. How can that work?

The best way to see what is happening is to work through a simple example. We have an experiment in which crop yields per unit area were measured from 10 randomly selected fields on each of three soil types. All fields were sown with the same variety of seed and provided with the same fertilizer and pest control inputs. The question is whether soil type significantly affects crop yield, and if so, to what extent.

```
results<-read.table("c:\\temp\\yields.txt",header=T)
attach(results)
names(results)
```

```
[1] "sand"  "clay"  "loam"
```

To see the data just type `results` followed by the Return key:

```
   sand clay loam
1     6   17   13
2    10   15   16
3     8    3    9
4     6   11   12
5    14   14   15
```

The R Book   Michael J. Crawley
© 2007 John Wiley & Sons, Ltd. ISBN: 978-0-470-51024-7

```
 6   17   12   16
 7    9   12   17
 8   11    8   13
 9    7   10   18
10   11   13   14
```

The function sapply is used to calculate the mean yields for the three soils:

```
sapply(list(sand,clay,loam),mean)
```

```
[1]  9.9  11.5  14.3
```

Mean yield was highest on loam (14.3) and lowest on sand (9.9).

It will be useful to have all of the yield data in a single vector called y:

```
y<-c(sand,clay,loam)
```

and to have a single vector called soil to contain the factor levels for soil type:

```
soil<-factor(rep(1:3,c(10,10,10)))
```

Before carrying out analysis of variance, we should check for constancy of variance (see Chapter 8) across the three soil types:

```
sapply(list(sand,clay,loam),var)
```

```
[1]  12.544444  15.388889  7.122222
```

The variances differ by more than a factor of 2. But is this significant? We test for heteroscedasticity using the Fligner–Killeen test of homogeneity of variances:

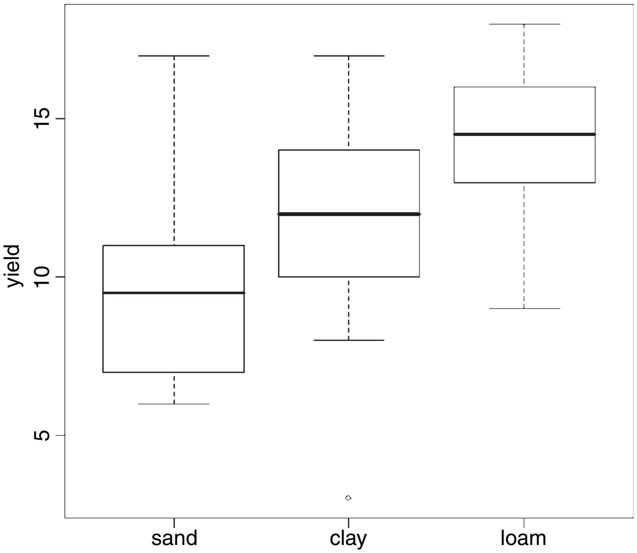
```
fligner.test(y~soil)
```

```
        Fligner-Killeen test of homogeneity of variances

data: y by soil
Fligner-Killeen:med chi-squared = 0.3651, df = 2, p-value = 0.8332
```

We could have used bartlett.test(y~soil), which gives $p = 0.5283$ (but this is more a test of non-normality than of equality of variances). Either way, there is no evidence of any significant difference in variance across the three samples, so it is legitimate to continue with our one-way analysis of variance.

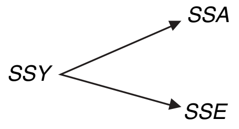
Because the explanatory variable is categorical (three levels of soil type), initial data inspection involves a box-and-whisker plot of y against soil like this:

```
plot(soil,y,names=c("sand","clay","loam"),ylab="yield")
```



Median yield is lowest on sand and highest on loam but there is considerable variation from replicate to replicate within each soil type (there is even an outlier on clay). It looks as if yield on loam will turn out to be significantly higher than on sand (their boxes do not overlap) but it is not clear whether yield on clay is significantly greater than on sand or significantly lower than on loam. The analysis of variance will answer these questions.

The analysis of variance involves calculating the total variation in the response variable (yield in this case) and partitioning it ('analysing it') into informative components. In the simplest case, we partition the total variation into just two components: explained variation and unexplained variation:



Explained variation is called the treatment sum of squares (*SSA*) and unexplained variation is called the error sum of squares (*SSE*, also known as the residual sum of squares). The unexplained variation is defined as the sum of the squares of the differences between the individual *y* values and the relevant treatment mean:

$$SSE = \sum_{i=1}^{k} \sum (y - \bar{y}_i)^2.$$

We compute the mean for the *i*th level of the factor in advance, and then add up the squares of the differences. Given that we worked it out this way, can you see how many degrees of freedom should be associated with *SSE*? Suppose that there were n replicates in each treatment ($n = 10$ in our example). And suppose that there are k levels of the factor ($k = 3$ in our example). If you estimate k parameters from the data before you can work out *SSE*, then you must have lost k degrees of freedom in the process. Since each of the k levels of the factor has n replicates, there must be $k \times n$ numbers in the whole experiment ($3 \times 10 = 30$ in our example). So the degrees of freedom associated with *SSE* are $kn - k = k(n - 1)$. Another way of seeing this is to say that there are n replicates in each treatment, and hence

$n-1$ degrees of freedom for error in each treatment (because 1 d.f. is lost in estimating each treatment mean). There are k treatments (i.e. k levels of the factor) and hence there are $k \times (n-1)$ d.f. for error in the experiment as a whole.

The component of the variation that is explained by differences between the treatment means, the treatment sum of squares, is traditionally denoted by *SSA*. This is because in two-way analysis of variance, with two different categorical explanatory variables, *SSB* is used to denote the sum of squares attributable to differences between the means of the second factor, *SSC* to denote the sum of squares attributable to differences between the means of the third factor, and so on.

Typically, we compute all but one of the components of the total variance, then find the value of the last component by subtraction of the others from *SSY*. We already have a formula for *SSE*, so we could obtain *SSA* by difference: $SSA = SSY - SSE$. Box 11.1 looks at the formula for *SSA* in more detail.

Box 11.1 Corrected sums of squares in one-way ANOVA

The definition of the total sum of squares, *SSY*, is the sum of the squares of the differences between the data points, y_{ij}, and the overall mean, $\bar{\bar{y}}$:

$$SSY = \sum_{i=1}^{k} \sum_{j=1}^{n} (y_{ij} - \bar{\bar{y}})^2,$$

where $\sum_{j=1}^{n} y_{ij}$ means the sum over the n replicates within each of the k factor levels. The error sum of squares, *SSE*, is the sum of the squares of the differences between the data points, y_{ij}, and their individual treatment means, $\bar{y}_i$:

$$SSE = \sum_{i=1}^{k} \sum_{j=1}^{n} (y_{ij} - \bar{y}_i)^2.$$

The treatment sum of squares, *SSA*, is the sum of the squares of the differences between the individual treatment means, $\bar{y}_i$, and the overall mean, $\bar{\bar{y}}$:

$$SSA = \sum_{i=1}^{k} \sum_{j=1}^{n} (\bar{y}_i - \bar{\bar{y}})^2 = n \sum_{i=1}^{k} (\bar{y}_i - \bar{\bar{y}})^2.$$

Squaring the bracketed term and applying summation gives

$$\sum \bar{y}_i^2 - 2\bar{\bar{y}} \sum \bar{y}_i + k\bar{\bar{y}}^2.$$

Let the grand total of all the values of the response variable $\sum_{i=1}^{k} \sum_{j=1}^{n} y_{ij}$ be shown as $\sum y$. Now replace $\bar{y}_i$ by T_i/n (where T is our conventional name for the k individual treatment totals) and replace $\bar{\bar{y}}$ by $\sum y/kn$ to get

$$\frac{\sum_{i=1}^{k} T_i^2}{n^2} - 2\frac{\sum y \sum_{i=1}^{k} T_i}{nkn} + k\frac{\sum y \sum y}{knkn}.$$

Note that $\sum_{i=1}^{k} T_i = \sum_{i=1}^{k} \sum_{j=1}^{n} y_{ij}$, so the right-hand positive and negative terms both have the form $(\sum y)^2/kn^2$. Finally, multiplying through by n gives

$$SSA = \frac{\sum T^2}{n} - \frac{(\sum y)^2}{kn}.$$

As an exercise, you should prove that $SSY = SSA + SSE$.

Let's work through the numbers in R. From the formula for *SSY*, we can obtain the total sum of squares by finding the differences between the data and the overall mean:

```
sum((y-mean(y))^2)
```

```
[1] 414.7
```

The unexplained variation, *SSE*, is calculated from the differences between the yields and the mean yields *for that soil type*:

```
sand-mean(sand)
```

```
[1] -3.9 0.1 -1.9 -3.9 4.1 7.1 -0.9 1.1 -2.9 1.1
```

```
clay-mean(clay)
```

```
[1] 5.5 3.5 -8.5 -0.5 2.5 0.5 0.5 -3.5 -1.5 1.5
```

```
loam-mean(loam)
```

```
[1] -1.3 1.7 -5.3 -2.3 0.7 1.7 2.7 -1.3 3.7 -0.3
```

We need the sums of the squares of these differences:

```
sum((sand-mean(sand))^2)
```

```
[1] 112.9
```

```
sum((clay-mean(clay))^2)
```

```
[1] 138.5
```

```
sum((loam-mean(loam))^2)
```

```
[1] 64.1
```

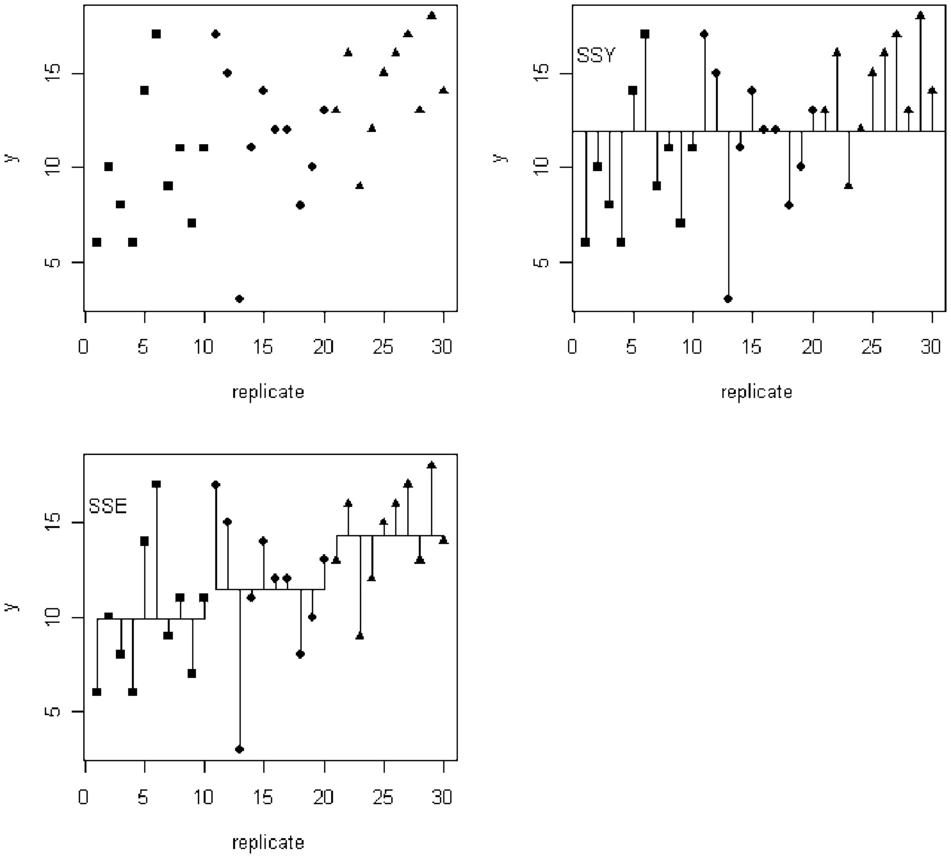
To get the sum of these totals across all soil types, we can use sapply like this:

```
sum(sapply(list(sand,clay,loam),function (x) sum((x-mean(x))^2) ))
```

```
[1] 315.5
```

So *SSE*, the unexplained (or residual, or error) sum of squares, is 315.5.

The extent to which *SSE* is less than *SSY* is a reflection of the magnitude of the differences between the means. The greater the difference between the mean yields on the different soil types, the greater will be the difference between *SSE* and *SSY*. This is the basis of analysis of variance. We can make inferences about differences between means by looking at differences between variances (or between sums or squares, to be more precise at this stage).



At top left we have an ‘index plot’ of the yields with different symbols for the different soil types: square = sand, diamond = clay, triangle = loam. At top right is a picture of the total sum of squares: *SSY* is the sum of the squares of the lengths of the lines joining each data point to the overall mean, $\bar{\bar{y}}$. At bottom left is a picture of the error sum of squares: *SSE* is the sum of the squares of the lengths of the lines joining each data point to its particular treatment mean, $\bar{y}_i$. The extent to which the lines are shorter in *SSE* than in *SSY* is a measure of the significance of the difference between the mean yields on the different soils. In the extreme case in which there was *no* variation between the replicates, then *SSY* is large, but *SSE* is zero:

This picture was created with the following code, where the x values, xvc, are

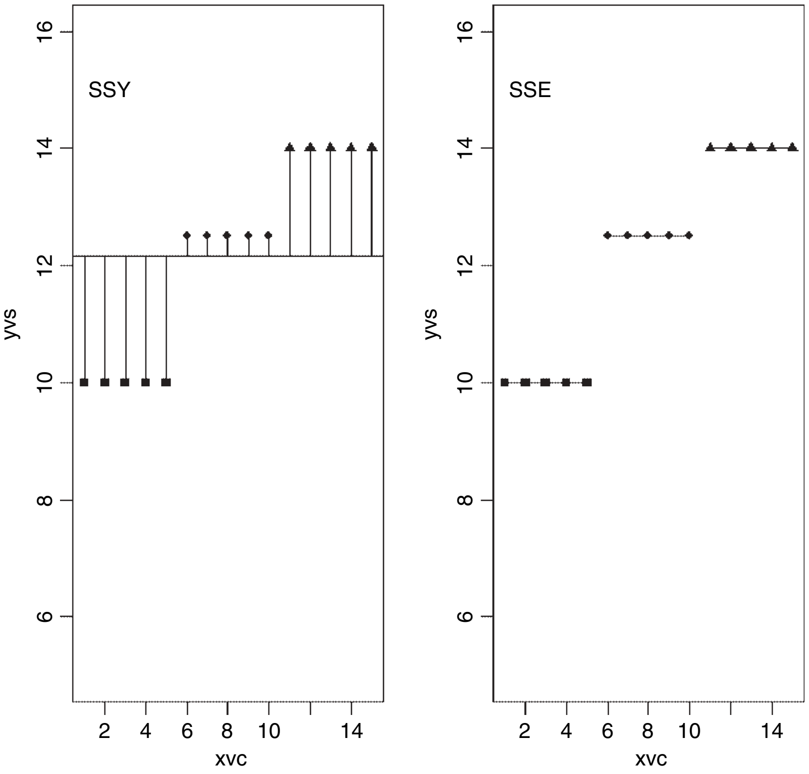
```
xvc<-1:15
```

and the y values, yvs, are

```
yvs<-rep(c(10,12,14),each=5)
```

To produce the two plots side by side, we write:

```
par(mfrow=c(1,2))
plot(xvc,yvs,ylim=c(5,16),pch=(15+(xvc>5)+(xvc>10)))
for (i in 1:15) lines(c(i,i),c(yvs[i],mean(yvs)))
```



```
abline(h=mean(yvs))
text(3,15,"SSY")
plot(xvc,yvs, ylim=c(5,16),pch=(15+(xvc(>)5)+(xvc(>)10)))
lines(c(1,5),c(10,10))
lines(c(6,10),c(12,12))
lines(c(11,15),c(14,14))
text(3,15,"SSE")
```

The difference between *SSY* and *SSE* is called the treatment sum of squares, *SSA*: this is the amount of the variation in yield that is explained by differences between the treatment means. In our example,

$$SSA = SSY - SSE = 414.7 - 315.5 = 99.2.$$

Now we can draw up the ANOVA table. There are six columns indicating, from left to right, the source of variation, the sum of squares attributable to that source, the degrees of freedom for that source, the variance for that source (traditionally called the mean square rather than the variance), the F ratio (testing the null hypothesis that this source of variation is not significantly different from zero) and the p value associated with that F value (if $p < 0.05$ then we reject the null hypothesis). We can fill in the sums of squares just calculated, then think about the degrees of freedom:

There are 30 data points in all, so the total degrees of freedom are $30 - 1 = 29$. We lose 1 d.f. because in calculating *SSY* we had to estimate one parameter from the data in advance,

Source	Sum of squares	Degrees of freedom	Mean square	F ratio	p value
Soil type	99.2	2	49.6	4.24	0.025
Error	315.5	27	$s^2 = 11.685$		
Total	414.7	29			

namely the overall mean, $\bar{\bar{y}}$. Each soil type has $n = 10$ replications, so each soil type has $10 - 1 = 9$ d.f. for error, because we estimated one parameter from the data *for each soil type*, namely the treatment means $\bar{y}_i$ in calculating *SSE*. Overall, therefore, the error has $3 \times 9 = 27$ d.f. There were 3 soil types, so there are $3 - 1 = 2$ d.f. for soil type.

The mean squares are obtained simply by dividing each sum of squares by its respective degrees of freedom (in the same row). The error variance, s^2, is the residual mean square (the mean square for the unexplained variation); this is sometimes called the 'pooled error variance' because it is calculated across all the treatments; the alternative would be to have three separate variances, one for each treatment:

```
sapply(list(sand,clay,loam),var)
```

```
[1] 12.544444 15.388889 7.122222
```

You will see that the pooled error variance $s^2 = 11.685$ is simply the mean of the three separate variances, because there is equal replication in each soil type ($n = 10$):

```
mean(sapply(list(sand,clay,loam),var))
```

```
[1] 11.68519
```

By tradition, we do not calculate the total mean square, so the bottom cell of the fourth column of the ANOVA table is empty. The F ratio is the treatment variance divided by the error variance, testing the null hypothesis that the treatment means are all the same. If we reject this null hypothesis, we accept the alternative hypothesis that *at least one of the means is significantly different from the others*. The question naturally arises at this point as to whether 4.24 is a big number or not. If it is a big number then we reject the null hypothesis. If it is not a big number, then we accept the null hypothesis. As ever, we decide whether the test statistic $F = 4.24$ is big or small by comparing it with the *critical value* of F, given that there are 2 d.f. in the numerator and 27 d.f. in the denominator. Critical values in R are found from the function qf which gives us quantiles of the F distribution:

```
qf(.95,2,27)
```

```
[1] 3.354131
```

Our calculated test statistic of 4.24 is larger than the critical value of 3.35, so we reject the null hypothesis. At least one of the soils has a mean yield that is significantly different from the others. The modern approach is not to work slavishly at the 5% level but rather to calculate the p-value associated with our test statistic of 4.24. Instead of using the function for quantiles of the F distribution, we use the function pf for cumulative probabilities of the F distribution like this:

```
1-pf(4.24,2,27)
```

```
[1] 0.02503987
```

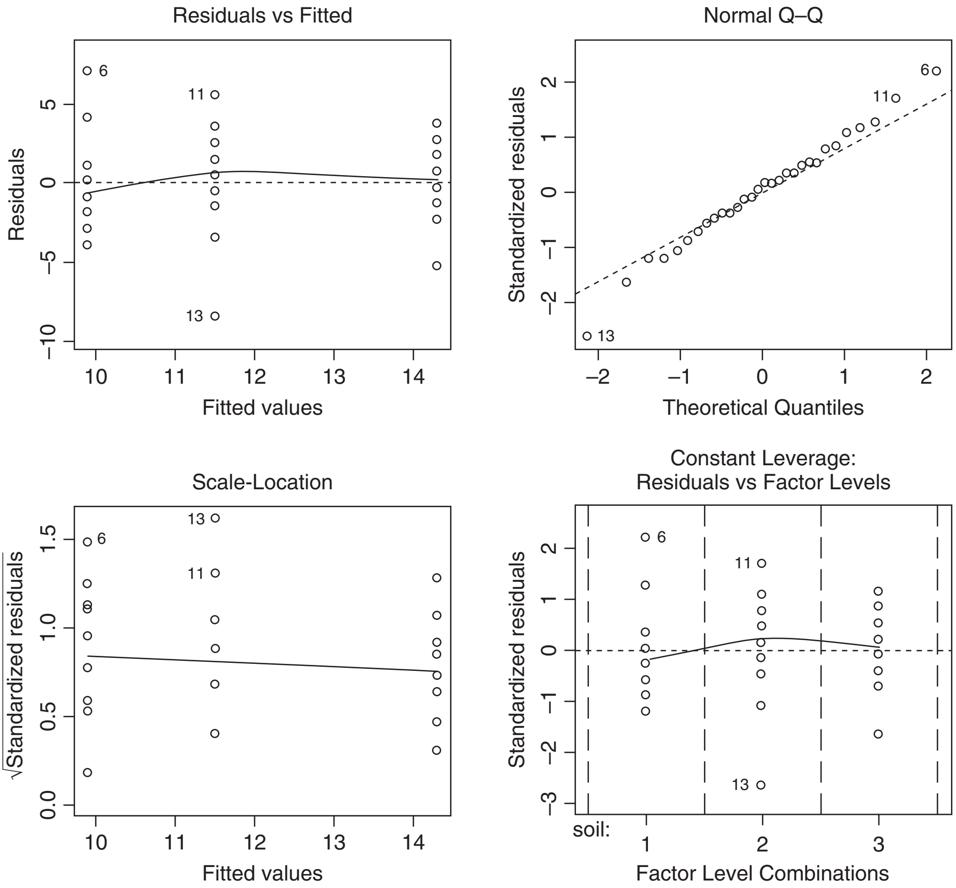
The p-value is 0.025, which means that a value of $F = 4.24$ or bigger would arise by chance alone when the null hypothesis was true about 25 times in 1000. This is a sufficiently small probability (i.e. it is less than 5%) for us to conclude that there is a significant difference between the mean yields (i.e. we reject the null hypothesis).

That was a lot of work. R can do the whole thing in a single line:

```
summary(aov(y~soil))
```

```
            Df   Sum Sq  Mean Sq  F value   Pr(>F)
soil         2   99.200   49.600   4.2447  0.02495  *
Residuals   27  315.500   11.685
```

Here you see all the values that we calculated long-hand. The error row is labelled `Residuals`. In the second and subsequent columns you see the degrees of freedom for treatment and error (2 and 27), the treatment and error sums of squares (99.2 and 315.5), the treatment mean square of 49.6, the error variance $s^2 = 11.685$, the F ratio and the p-value (labelled `Pr(> F)`). The single asterisk next to the p value indicates that the difference



between the soil means is significant at 5% (but not at 1%, which would have merited two asterisks). Notice that R does not print the bottom row of the ANOVA table showing the total sum of squares and total degrees of freedom.

The next thing we would do is to check the assumptions of the aov model. This is done using plot like this (see Chapter 10):

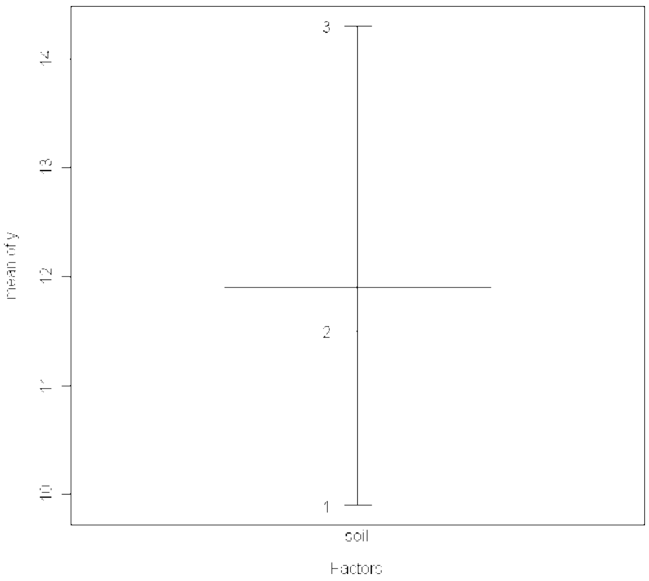
```
plot(aov(y~soil))
```

The first plot (top left) checks the most important assumption of constancy of variance; there should be no pattern in the residuals against the fitted values (the three treatment means) – and, indeed, there is none. The second plot (top right) tests the assumption of normality of errors: there should be a straight-line relationship between our standardized residuals and theoretical quantiles derived from a normal distribution. Points 6, 11 and 13 lie a little off the straight line, but this is nothing to worry about (see p. 339). The residuals are well behaved (bottom left) and there are no highly influential values that might be distorting the parameter estimates (bottom right).

Effect sizes

The best way to view the effect sizes graphically is to use plot.design (which takes a formula rather than a model object):

```
plot.design(y~soil)
```



For more complicated models, you might want to use the effects library to get more attractive plots (p. 178). To see the effect sizes in tabular form use model.tables (which takes a model object as its argument) like this:

```
model<-aov(y~soil);model.tables(model,se=T)
```

```
Tables of effects

  soil
soil
   1     2   3
-2.0  -0.4  2.4

Standard errors of effects
          soil
         1.081
replic.    10
```

The effects are shown as departures from the overall mean: soil 1 (sand) has a mean yield that is 2.0 below the overall mean, and soil 3 (loam) has a mean that is 2.4 above the overall mean. The standard error of effects is 1.081 on a replication of $n = 10$ (this is the standard error of a mean). You should note that this is *not* the appropriate standard error for comparing two means (see below). If you specify "means" you get:

```
model.tables(model,"means",se=T)
```

```
Tables of means
Grand mean

11.9

  soil
soil
    1     2     3
  9.9  11.5  14.3

Standard errors for differences of means
            soil
           1.529
replic.      10
```

Now the three means are printed (rather than the effects) and the standard error of the difference of means is given (this *is* what you need for doing a t test to compare any two means).

Another way of looking at effect sizes is to use the summary.lm option for viewing the model, rather than summary.aov (as we used above):

```
summary.lm(aov(y~soil))
```

```
Coefficients:
              Estimate  Std. Error  t value  Pr(>|t|)
(Intercept)      9.900       1.081    9.158  9.04e-10 ***
soil2            1.600       1.529    1.047   0.30456
soil3            4.400       1.529    2.878   0.00773  **

Residual standard error: 3.418 on 27 degrees of freedom
Multiple R-Squared: 0.2392,  Adjusted R-squared: 0.1829
F-statistic: 4.245 on 2 and 27 DF, p-value: 0.02495
```

In regression analysis (p. 399) the summary.lm output was easy to understand because it gave us the intercept and the slope (the two parameters estimated by the model) and their standard errors. But this table has three rows. Why is that? What is an intercept in the context of analysis of variance? And why are the standard errors different for the intercept and for soil 2 and soil 3? Come to that, what are soil 2 and soil 3?

It will take a while before you feel at ease with summary.lm tables for analysis of variance. The details are explained on p. 365, but the central point is that all summary.lm tables have as many rows as there are parameters estimated from the data. There are three rows in this case because our aov model estimates three parameters; a mean yield for each of the three soil types. In the context of aov, an intercept is a mean value; in this case it is the mean yield for sand because we gave that factor level 1 when we computed the vales for the factor called soil, earlier. In general, the intercept would be the factor level whose name came lowest in alphabetical order (see p. 366). So if Intercept is the mean yield for sand, what are the other two rows labelled soil2 and soil3. This is the hardest thing to understand. All other rows in the summary.lm table for aov are differences between means. Thus row 2, labelled soil2, is the difference between the mean yields on sand and clay, and row 3, labelled soil3, is the difference between the mean yields of sand and loam:

```
tapply(y,soil,mean)-mean(sand)
```

```
  1    2    3
0.0  1.6  4.4
```

The first row (Intercept) is a mean, so the standard error column in row 1 contains the standard error of a mean. Rows 2 and 3 are differences between means, so their standard error columns contain the standard error of the difference between two means (and this is a bigger number; see p. 367). The standard error of a mean is

$$se_{\text{mean}} = \sqrt{\frac{s^2}{n}} = \sqrt{\frac{11.685}{10}} = 1.081,$$

whereas the standard error of the difference between two means is

$$se_{\text{diff}} = \sqrt{2\frac{s^2}{n}} = \sqrt{2 \times \frac{11.685}{10}} = 1.529.$$

The summary.lm table shows that soil 3 produces significantly bigger yields than soil 1 (the intercept) with a p-value of 0.007 73. The difference between the two means was 4.400 and the standard error of the difference was 1.529. This difference is of two-star significance, meaning $0.001 < p < 0.01$. In contrast, soil 2 does not produce a significantly greater yield than soil 1; the difference is 1.600 and the standard error of the difference was 1.529 ($p = 0.304\,56$). The only remaining issue is whether soil 2 yielded significantly less than soil 3. We need to do some mental arithmetic to work this out: the difference between these two means was $4.4 - 1.6 = 2.8$ and so the t value is $2.8/1.529 = 1.83$. This is less than 2 (the rule of thumb for t) so the mean yields of soils 2 and 3 are not significantly different. To find the precise value with 10 replicates, the critical value of t is given by the function qt with 18 d.f.:

```
qt(0.975,18)
```

```
[1] 2.100922
```

Alternatively we can work out the p value associated with our calculated $t = 1.83$:

```
2*(1 - pt(1.83, df = 18))
```

```
[1] 0.0838617
```

giving $p = 0.084$. We multiply by 2 because this is a two-tailed test (see p. 208); we did not know in advance that loam would out-yield clay under the particular circumstances of this experiment.

The residual standard error in the summary.lm output is the square root of the error variance from the ANOVA table: $\sqrt{11.685} = 3.418$. R-Squared and Adjusted R-Squared are explained on p. 399. The F-statistic and the p-value come from the last two columns of the ANOVA table.

So there it is. That is how analysis of variance works. When the means are significantly different, then the sum of squares computed from the individual treatment means will be significantly smaller than the sum of squares computed from the overall mean. We judge the significance of the difference between the two sums of squares using analysis of variance.

Plots for interpreting one-way ANOVA

There are two traditional ways of plotting the results of ANOVA:

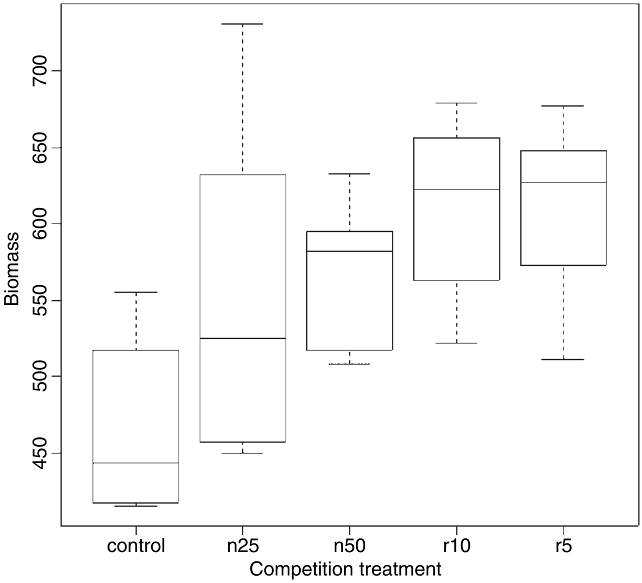
- box-and-whisker plots;
- barplots with error bars.

Here is an example to compare the two approaches. We have an experiment on plant competition with one factor and five levels. The factor is called clipping and the five levels consist of control (i.e. unclipped), two intensities of shoot pruning and two intensities of root pruning:

```
comp<-read.table("c:\\temp\\competition.txt",header=T);attach(comp);names(comp)
```

```
[1] "biomass" "clipping"
```

```
plot(clipping,biomass,xlab="Competition treatment",ylab="Biomass")
```



The box-and-whisker plot is good at showing the nature of the variation within each treatment, and also whether there is skew within each treatment (e.g. for the control plots, there is a wider range of values between the median and third quartile than between the median and first quartile). No outliers are shown above the whiskers, so the tops and bottoms of the bars are the maxima and minima within each treatment. The medians for the competition treatments are all higher than the third quartile of the controls, suggesting that they may be significantly different from the controls, but there is little to suggest that any of the competition treatments are significantly different from one another (see below for the analysis). We could use the notch=T option to get a visual impression of the significance of differences between the means; all the treatment medians fall outside the notch of the controls, but no other comparisons appear to be significant.

Barplots with error bars are preferred by many journal editors, and some people think that they make hypothesis testing easier. We shall see. Unlike S-PLUS, R does not have a built-in function called error.bar so we shall have to write our own. Here is a very simple version without any bells or whistles. We shall call it error.bars to distinguish it from the much more general S-PLUS function.

```
error.bars<-function(yv,z,nn){
xv<-
barplot(yv,ylim=c(0,(max(yv)+max(z))),names=nn,ylab=deparse(substitute(yv)
))
g=(max(xv)-min(xv))/50
for (i in 1:length(xv)) {
lines(c(xv[i],xv[i]),c(yv[i]+z[i],yv[i]-z[i]))
lines(c(xv[i]-g,xv[i]+g),c(yv[i]+z[i], yv[i]+z[i]))
lines(c(xv[i]-g,xv[i]+g),c(yv[i]-z[i], yv[i]-z[i]))
}}
```

To use this function we need to decide what kind of values (z) to use for the lengths of the bars. Let’s use the standard error of a mean based on the pooled error variance from the ANOVA, then return to a discussion of the pros and cons of different kinds of error bars later. Here is the one-way analysis of variance:

```
model<-aov(biomass~clipping)
summary(model)
```

```
            Df   Sum Sq   Mean Sq   F value     Pr(>F)
clipping     4    85356     21339    4.3015   0.008752   **
Residuals   25   124020      4961
```

From the ANOVA table we learn that the pooled error variance $s^2 = 4961.0$. Now we need to know how many numbers were used in the calculation of each of the five means:

```
table(clipping)
```

```
clipping
control   n25   n50   r10   r5
      6     6     6     6    6
```

There was equal replication (which makes life easier), and each mean was based on six replicates, so the standard error of a mean is $\sqrt{s^2/n} = \sqrt{4961/6} = 28.75$. We shall draw an error bar up 28.75 from each mean and down by the same distance, so we need 5 values for z, one for each bar, each of 28.75:

```
se<-rep(28.75,5)
```

We need to provide labels for the five different bars: the factor levels should be good for this:

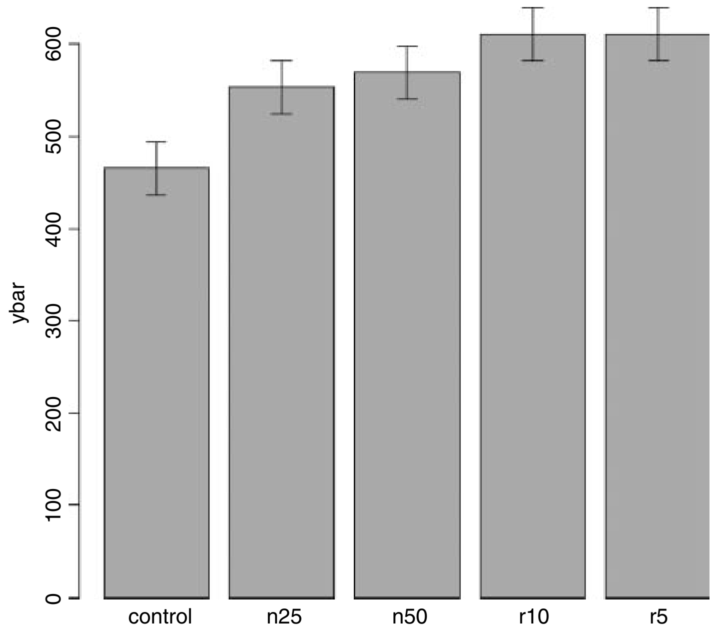
```
labels<-as.character(levels(clipping))
```

Now we work out the five mean values which will be the heights of the bars, and save them as a vector called ybar:

```
ybar<-as.vector(tapply(biomass,clipping,mean))
```

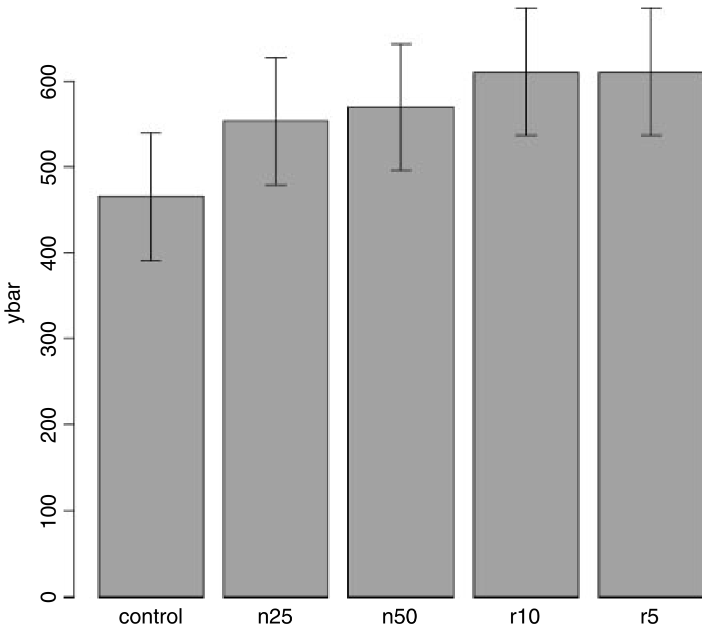
Finally, we can create the barplot with error bars (the function is defined on p. 462):

```
error.bars(ybar,se,labels)
```



We do not get the same feel for the distribution of the values *within* each treatment as was obtained by the box-and-whisker plot, but we can certainly see clearly which means are not significantly different. If, as here, we use ±1 standard error as the length of the error bars, then *when the bars overlap this implies that the two means are not significantly different*. Remember the rule of thumb for *t*: significance requires 2 or more standard errors, and if the bars overlap it means that the difference between the means is less than 2 standard errors. There is another issue, too. For comparing means, we should use the standard error of the difference between two means (not the standard error of one mean) in our tests (see p. 294); these bars would be about 1.4 times as long as the bars we have drawn here. So while we can be sure that the two root-pruning treatments are not significantly different from one another, and that the two shoot-pruning treatments are not significantly different from one another (because their bars overlap), we cannot conclude from this plot that the controls have significantly lower biomass than the rest (because the error bars are not the correct length for testing differences between means).

An alternative graphical method is to use 95% confidence intervals for the lengths of the bars, rather than standard errors of means. This is easy to do: we multiply our standard errors by Student's t, qt(.975,5) = 2.570 582, to get the lengths of the confidence intervals:



Now, all of the error bars overlap, implying visually that there are no significant differences between the means. But we know that this is not true from our analysis of variance, in which we rejected the null hypothesis that all the means were the same at $p = 0.008\,75$. If it were the case that the bars did not overlap when we are using confidence intervals (as here), then that would imply that the means differed by more than 4 standard errors, and this is a much greater difference than is required to conclude that the means are significantly different. So this is not perfect either. With standard errors we could be sure that the means were *not* significantly different when the bars *did* overlap. And with confidence intervals we can be sure that the means *are* significantly different when the bars *do not* overlap. But the alternative cases are not clear-cut for either type of bar. Can we somehow get the best of both worlds, so that the means *are* significantly different when the bars *do not* overlap, and the means are *not* significantly different when the bars *do* overlap?

The answer is yes, we can, if we use least significant difference (LSD) bars. Let's revisit the formula for Student's t test:

$$t = \frac{\text{a difference}}{\text{standard error of the difference}}$$

We say that the difference is significant when $t > 2$ (by the rule of thumb, or $t >$ qt(0.975,df) if we want to be more precise). We can rearrange this formula to find the smallest difference that we would regard as being significant. We can call this the least significant difference:

$$LSD = \text{qt(0.975,df)} \times \text{standard error of a difference} \approx 2 \times se_{\text{diff}}.$$

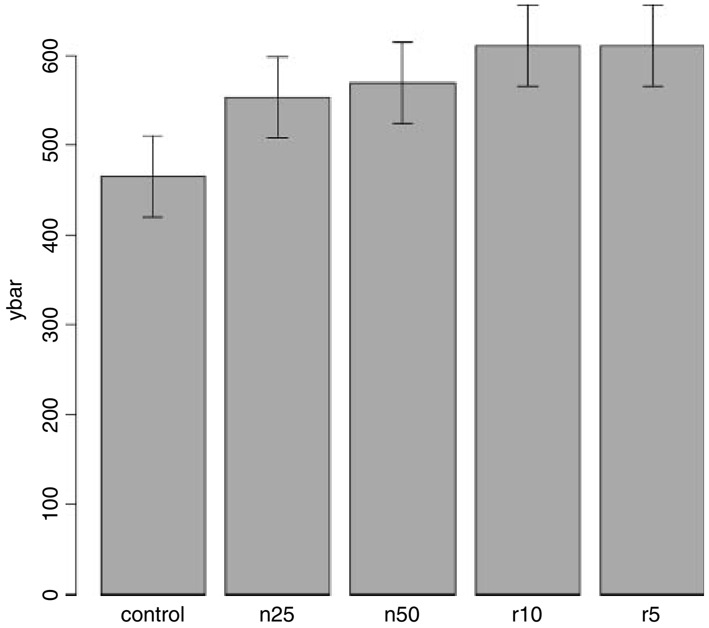
In our present example this is

```
qt(0.975,10)*sqrt(2*4961/6)
```

```
[1] 90.60794
```

because a difference is based on $12 - 2 = 10$ degrees of freedom. What we are saying is the two means would be significantly different if they differed by 90.61 or more. How can we show this graphically? We want overlapping bars to indicate a difference less than 90.61, and non-overlapping bars to represent a difference greater than 90.61. With a bit of thought you will realize that we need to draw bars that are *LSD*/2 in length, up and down from each mean. Let's try it with our current example:

```
lsd<-qt(0.975,10)*sqrt(2*4961/6)
lsdbars<-rep(lsd,5)/2
error.bars(ybar,lsdbars,labels)
```



Now we can interpret the significant differences visually. The control biomass is significantly lower than any of the four treatments, but none of the four treatments is significantly different from any other. The statistical analysis of this contrast is explained in detail in Chapter 9. Sadly, most journal editors insist on error bars of 1 standard error. It is true that there are complicating issues to do with LSD bars (not least the vexed question of multiple comparisons; see p. 483), but at least they do what was intended by the error plot (i.e. overlapping bars means non-significance and non-overlapping bars means significance); neither standard errors nor confidence intervals can say that. A better option might be to use box-and-whisker plots with the notch=T option to indicate significance (see p. 159).

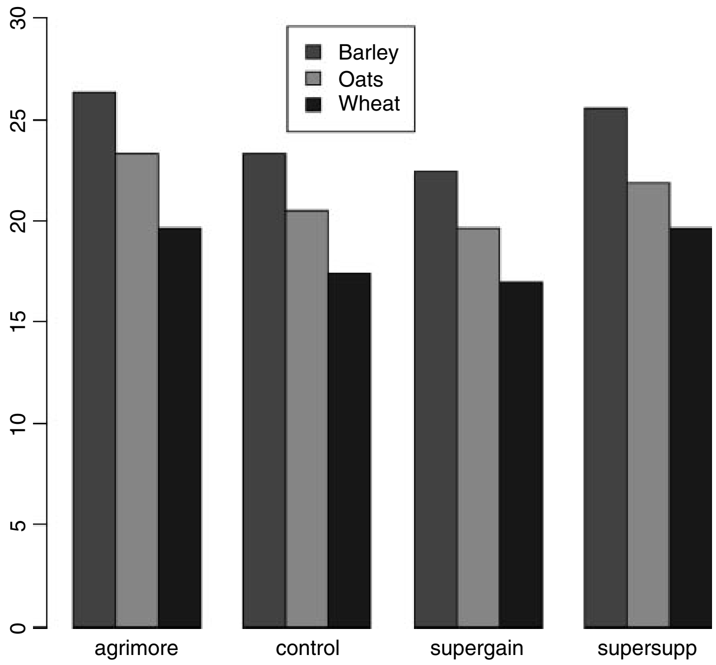
Factorial Experiments

A factorial experiment has two or more factors, each with two or more levels, plus replication for each combination of factors levels. This means that we can investigate statistical interactions, in which the response to one factor depends on the level of another factor. Our example comes from a farm-scale trial of animal diets. There are two factors: diet and supplement. Diet is a factor with three levels: barley, oats and wheat. Supplement is a factor with four levels: agrimore, control, supergain and supersupp. The response variable is weight gain after 6 weeks.

```
weights<-read.table("c:\\temp\\growth.txt",header=T)
attach(weights)
```

Data inspection is carried out using `barplot` (note the use of `beside=T` to get the bars in adjacent clusters rather than vertical stacks):

```
barplot(tapply(gain,list(diet,supplement),mean),
                    beside=T,ylim=c(0,30),col=rainbow(3))
```



Note that the second factor in the list (supplement) appears as groups of bars from left to right in alphabetical order by factor level, from agrimore to supersupp. The second factor (diet) appears as three levels within each group of bars: red = barley, green = oats, blue = wheat, again in alphabetical order by factor level. We should really add a key to explain the levels of diet. Use `locator`(1) to find the coordinates for the *top left* corner of the box around the legend. You need to increase the default scale on the *y* axis to make enough room for the legend box.

```
labs<-c("Barley","Oats","Wheat")
legend(locator(1),labs,fill=rainbow(3))
```

We inspect the mean values using tapply as usual:

```
tapply(gain,list(diet,supplement),mean)
```

```
        agrimore   control  supergain  supersupp
barley  26.34848  23.29665   22.46612   25.57530
oats    23.29838  20.49366   19.66300   21.86023
wheat   19.63907  17.40552   17.01243   19.66834
```

Now we use aov or lm to fit a factorial analysis of variance (the choice affects whether we get an ANOVA table or a list of parameters estimates as the default output from summary). We estimate parameters for the main effects of each level of diet and each level of supplement, plus terms for the interaction between diet and supplement. Interaction degrees of freedom are the product of the degrees of freedom of the component terms (i.e. $(3-1)\times(4-1)=6$). The model is gain~diet + supplement + diet:supplement, but this can be simplified using the asterisk notation like this:

```
model<-aov(gain~diet*supplement)
summary(model)
```

```
                  Df   Sum Sq  Mean Sq  F value     Pr(>F)
diet               2  287.171  143.586  83.5201  2.998e-14  ***
supplement         3   91.881   30.627  17.8150  2.952e-07  **
diet:supplement    6    3.406    0.568   0.3302     0.9166
Residuals         36   61.890    1.719
```

The ANOVA table shows that there is no hint of any interaction between the two explanatory variables ($p=0.9166$); evidently the effects of diet and supplement are additive. The disadvantage of the ANOVA table is that it does not show us the effect sizes, and does not allow us to work out how many levels of each of the two factors are significantly different. As a preliminary to model simplification, summary.lm is often more useful than summary.aov:

```
summary.lm(model)
```

```
Coefficients:
                              Estimate  Std. Error  t value  Pr(>|t|)
(Intercept)                    26.3485      0.6556   40.191   < 2e-16  ***
dietoats                       -3.0501      0.9271   -3.290  0.002248  **
dietwheat                      -6.7094      0.9271   -7.237  1.61e-08  ***
supplementcontrol              -3.0518      0.9271   -3.292  0.002237  **
supplementsupergain            -3.8824      0.9271   -4.187  0.000174  ***
supplementsupersupp            -0.7732      0.9271   -0.834  0.409816
dietoats:supplementcontrol      0.2471      1.3112    0.188  0.851571
dietwheat:supplementcontrol     0.8183      1.3112    0.624  0.536512
dietoats:supplementsupergain    0.2470      1.3112    0.188  0.851652
dietwheat:supplementsupergain   1.2557      1.3112    0.958  0.344601
dietoats:supplementsupersupp   -0.6650      1.3112   -0.507  0.615135
dietwheat:supplementsupersupp   0.8024      1.3112    0.612  0.544381

Residual standard error: 1.311 on 36 degrees of freedom
Multiple R-Squared: 0.8607, Adjusted R-squared: 0.8182
F-statistic: 20.22 on 11 and 36 DF, p-value: 3.295e-012
```

This is a rather complex model, because there are 12 estimated parameters (the number of rows in the table): six main effects and six interactions. The output re-emphasizes that none of the interaction terms is significant, but it suggests that the minimal adequate model will require five parameters: an intercept, a difference due to oats, a difference due to wheat, a difference due to control and difference due to supergain (these are the five rows with significance stars). This draws attention to the main shortcoming of using treatment contrasts as the default. If you look carefully at the table, you will see that the effect sizes of two of the supplements, control and supergain, are not significantly different from one another. You need lots of practice at doing *t* tests in your head, to be able to do this quickly. Ignoring the signs (because the signs are negative for both of them), we have 3.05 vs. 3.88, a difference of 0.83. But look at the associated standard errors (both 0.927); the difference is less than 1 standard error of a difference between two means. For significance, we would need roughly 2 standard errors (remember the rule of thumb, in which $t \geq 2$ is significant; see p. 228). The rows get starred in the significance column because treatments contrasts compare all the main effects in the rows with the intercept (where each factor is set to its first level in the alphabet, namely agrimore and barley in this case). When, as here, several factor levels are different from the intercept, but not different from one another, they all get significance stars. This means that you cannot count up the number of rows with stars in order to determine the number of significantly different factor levels.

We first simplify the model by leaving out the interaction terms:

```
model<-aov(gain~diet+supplement)
summary.lm(model)
```

```
Coefficients:
                    Estimate  Std. Error  t value  Pr(>|t|)
(Intercept)          26.1230      0.4408   59.258   < 2e-16  ***
dietoats             -3.0928      0.4408   -7.016  1.38e-08  ***
dietwheat            -5.9903      0.4408  -13.589   < 2e-16  ***
supplementcontrol    -2.6967      0.5090   -5.298  4.03e-06  ***
supplementsupergain  -3.3815      0.5090   -6.643  4.72e-08  ***
supplementsupersupp  -0.7274      0.5090   -1.429     0.160
```

It is clear that we need to retain all three levels of diet (oats differ from wheat by $5.99 - 3.09 = 2.90$ with a standard error of 0.44). It is not clear that we need four levels of supplement, however. Supersupp is not obviously different from agrimore (0.727 with standard error 0.509). Nor is supergain obviously different from the unsupplemented control animals ($3.38 - 2.70 = 0.68$). We shall try a new two-level factor to replace the four-level supplement, and see if this significantly reduces the model's explanatory power. Agrimore and supersupp are recoded as best and control and supergain as worst:

```
supp2<-factor(supplement)
levels(supp2)
```

```
[1] "agrimore" "control" "supergain" "supersupp"
```

```
levels(supp2)[c(1,4)]<-"best"
levels(supp2)[c(2,3)]<-"worst"
levels(supp2)
```

```
[1]  "best"  "worst"
```

Now we can compare the two models:

```
model2<-aov(gain~diet+supp2)
anova(model,model2)
```

```
Analysis of Variance Table

Model 1: gain ~ diet + supplement
Model 2: gain ~ diet + supp2

Res.Df         RSS  Df  Sum of Sq       F  Pr((>)F)
1       42  65.296
2       44  71.284  –2    –5.988  1.9257    0.1584
```

The simpler model2 has saved two degrees of freedom and is not significantly worse than the more complex model ($p = 0.158$). This is the minimal adequate model: all of the parameters are significantly different from zero and from one another:

```
summary.lm(model2)
```

```
Coefficients:
                Estimate  Std. Error  t value  Pr(>|t|)
(Intercept)      25.7593      0.3674   70.106   < 2e-16  ***
dietoats         –3.0928      0.4500   –6.873  1.76e-08  ***
dietwheat        –5.9903      0.4500  –13.311   < 2e-16  ***
supp2worst       –2.6754      0.3674   –7.281  4.43e-09  ***

Residual standard error: 1.273 on 44 degrees of freedom
Multiple R-Squared: 0.8396, Adjusted R-squared: 0.8286
F-statistic: 76.76 on 3 and 44 DF, p-value: 0
```

Model simplification has reduced our initial 12-parameter model to a four-parameter model.

Pseudoreplication: Nested Designs and Split Plots

The model-fitting functions aov and lmer have the facility to deal with complicated error structures, and it is important that you can recognize them, and hence avoid the pitfalls of pseudoreplication. There are two general cases:

- nested sampling, as when repeated measurements are taken from the same individual, or observational studies are conduced at several different spatial scales (mostly random effects);
- split-plot analysis, as when designed experiments have different treatments applied to plots of different sizes (mostly fixed effects).

Split-plot experiments

In a split-plot experiment, different treatments are applied to plots of different sizes. Each different plot size is associated with its own error variance, so instead of having one error variance (as in all the ANOVA tables up to this point), we have as many error terms as there are different plot sizes. The analysis is presented as a series of component ANOVA tables, one for each plot size, in a hierarchy from the largest plot size with the lowest replication at the top, down to the smallest plot size with the greatest replication at the bottom.

The following example refers to a designed field experiment on crop yield with three treatments: irrigation (with two levels, irrigated or not), sowing density (with three levels, low, medium and high), and fertilizer application (with three levels, low, medium and high).

```
yields<-read.table("c:\\temp\\splityield.txt",header=T)
attach(yields)
names(yields)
```

```
[1] "yield"  "block"  "irrigation"  "density"  "fertilizer"
```

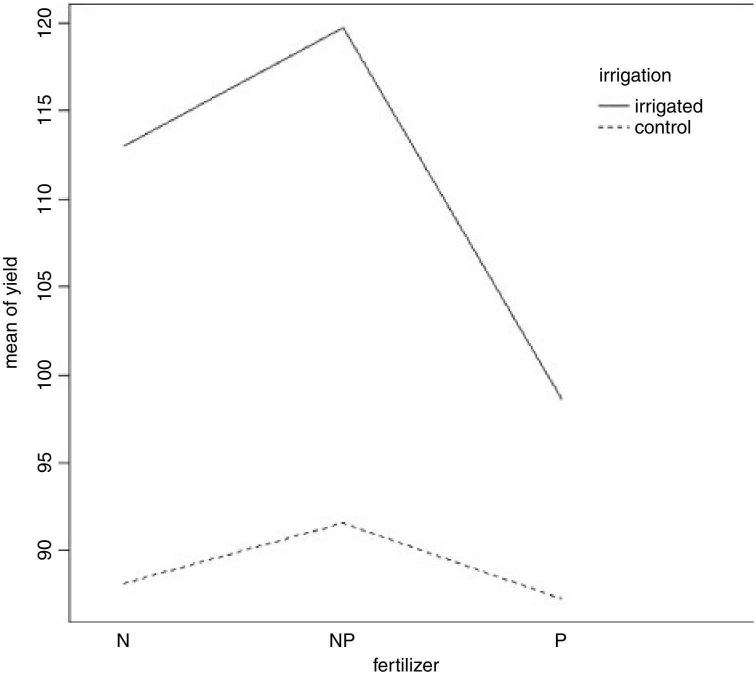
The largest plots were the four whole fields (block), each of which was split in half, and irrigation was allocated at random to one half of the field. Each irrigation plot was split into three, and one of three different seed-sowing densities (low, medium or high) was allocated at random (independently for each level of irrigation and each block). Finally, each density plot was divided into three, and one of three fertilizer nutrient treatments (N, P, or N and P together) was allocated at random. The model formula is specified as a factorial, using the asterisk notation. The error structure is defined in the Error term, with the plot sizes listed from left to right, from largest to smallest, with each variable separated by the slash operator /. Note that the smallest plot size, fertilizer, does not need to appear in the Error term:

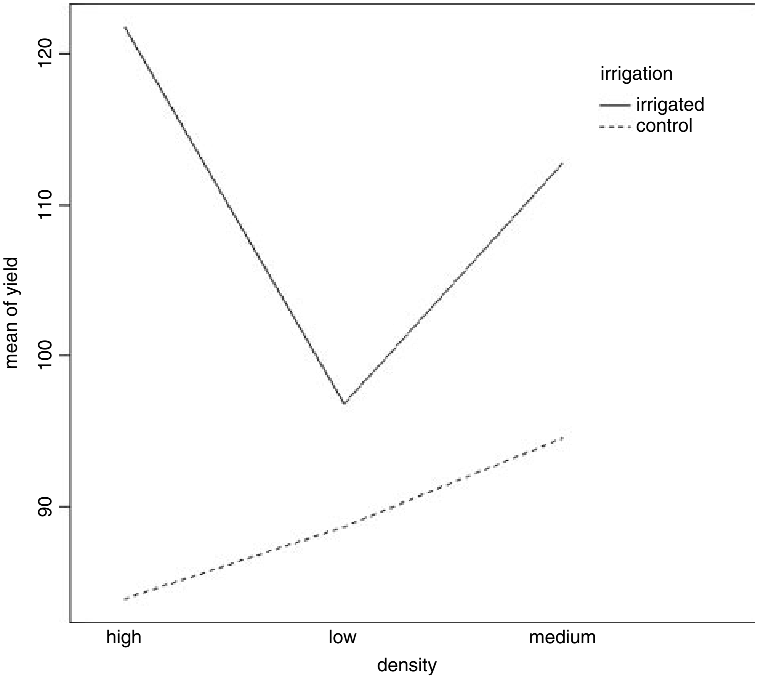
```
model<-aov(yield~irrigation*density*fertilizer+Error(block/irrigation/density))
summary(model)
```

```
Error:  block
           Df   Sum Sq  Mean Sq  F value  Pr(>F)
Residuals   3  194.444   64.815

Error: block:irrigation

            Df  Sum Sq  Mean Sq  F value   Pr(>F)
irrigation   1  8277.6   8277.6   17.590  0.02473  *
Residuals    3  1411.8    470.6

Error: block:irrigation:density

                    Df   Sum Sq   Mean Sq  F value   Pr(>F)
density              2  1758.36    879.18   3.7842  0.05318  .
irrigation:density   2  2747.03   1373.51   5.9119  0.01633  *
Residuals           12  2787.94    232.33

Error: Within
                              Df   Sum Sq  Mean Sq  F value     Pr(>F)
fertilizer                     2  1977.44   988.72  11.4493  0.0001418  ***
irrigation:fertilizer          2   953.44   476.72   5.5204  0.0081078  **
density:fertilizer             4   304.89    76.22   0.8826  0.4840526
irrigation:density:fertilizer  4   234.72    58.68   0.6795  0.6106672
Residuals                     36  3108.83    86.36
```

Here you see the four ANOVA tables, one for each plot size: blocks are the biggest plots, half blocks get the irrigation treatment, one third of each half block gets a sowing density treatment, and one third of a sowing density treatment gets each fertilizer treatment. Note that the non-significant main effect for density ($p = 0.053$) does *not* mean that density is unimportant, because density appears in a significant interaction with irrigation (the density terms cancel out, when averaged over the two irrigation treatments; see below). The best way to understand the two significant interaction terms is to plot them using interaction.plot like this:

```
interaction.plot(fertilizer,irrigation,yield)
```





Irrigation increases yield proportionately more on the N-fertilized plots than on the P-fertilized plots. The irrigation–density interaction is more complicated:

```
interaction.plot(density,irrigation,yield)
```

On the irrigated plots, yield is minimal on the low-density plots, but on control plots yield is minimal on the high-density plots. Alternatively, you could use the effects package which takes a model object (a linear model or a generalized linear model) and provides attractive trellis plots of specified interaction effects (p. 178).

Missing values in a split-plot design

When there are missing values, then factors have effects in more than one stratum and the same main effect turns up in more than one ANOVA table. Suppose that the 69th yield value was missing:

```
yield[69]<-NA
```

Now the summary table looks very different:

```
model<-aov(yield~irrigation*density*fertilizer+Error(block/irrigation/density))
summary(model)
```

```
Error: block
           Df   Sum Sq  Mean Sq  F value  Pr(>F)
irrigation  1    0.075    0.075    9e-04  0.9788
Residuals   2  167.704   83.852

Error: block:irrigation
           Df  Sum Sq  Mean Sq  F value   Pr(>F)
irrigation  1  7829.9   7829.9  21.9075  0.04274  *
density     1   564.4    564.4   1.5792  0.33576
Residuals   2   714.8    357.4

Error: block:irrigation:density
                    Df   Sum Sq  Mean Sq    F value   Pr(>F)
density              2  1696.47   848.24     3.4044  0.07066  .
fertilizer           1     0.01     0.01  2.774e-05  0.99589
irrigation:density   2  2786.75  1393.37     5.5924  0.02110  *
Residuals           11  2740.72   249.16

Error: Within
                               Df  Sum Sq Mean Sq F value    Pr(>F)
fertilizer                      2 1959.36  979.68 11.1171 0.0001829 ***
irrigation:fertilizer           2  993.59  496.79  5.6375 0.0075447 **
density:fertilizer              4  273.56   68.39  0.7761 0.5482571
irrigation:density:fertilizer   4  244.49   61.12  0.6936 0.6014280
Residuals                      35 3084.33   88.12
```

Notice that with just one missing value, each main effect appears in two tables (not one, as above). It is recommended that when there are missing values in a split-plot experiment you use lmer or lme instead of aov to fit the model.

Random effects and nested designs

Mixed-effects models are so called because the explanatory variables are a mixture of fixed effects and random effects:

- fixed effects influence only the *mean* of *y*;
- random effects influence only the *variance* of *y*.

A random effect should be thought of as coming from a population of effects: the existence of this population is an extra assumption. We speak of **prediction** of random effects, rather than estimation: we **estimate** fixed effects from data, but we intend to make predictions about the population from which our random effects were sampled. Fixed effects are unknown constants to be estimated from the data. Random effects govern the variance – covariance structure of the response variable. The fixed effects are often experimental treatments that were applied under our direction, and the random effects are either categorical or continuous variables that are distinguished by the fact that we are typically not interested in the parameter values, but only in the variance they explain.

One of more of the explanatory variables represents **grouping** in time or in space. Random effects that come from the same group will be correlated, and this contravenes one of the fundamental assumptions of standard statistical models: **independence of errors**. Mixed-effects models take care of this non-independence of errors by modelling the covariance structure introduced by the grouping of the data. A major benefit of random-effects models is that they economize on the number of degrees of freedom used up by the factor levels. Instead of estimating a mean for every single factor level, the random-effects model estimates the distribution of the means (usually as the standard deviation of the differences of the factor-level means around an overall mean). Mixed-effects models are particularly useful in cases where there is temporal pseudoreplication (repeated measurements) and/or spatial pseudoreplication (e.g. nested designs or split-plot experiments). These models can allow for

- spatial autocorrelation between neighbours;
- temporal autocorrelation across repeated measures on the same individuals;
- differences in the mean response between blocks in a field experiment;
- differences between subjects in a medical trial involving repeated measures.

The point is that we really do not want to waste precious degrees of freedom in estimating parameters for each of the separate levels of the categorical random variables. On the other hand, we do want to make use of the all measurements we have taken, but because of the pseudoreplication we want to take account of both the

- correlation structure, used to model within-group correlation associated with temporal and spatial dependencies, using **correlation**, and
- variance function, used to model non-constant variance in the within-group errors using **weights**.

Fixed or random effects?

It is difficult without lots of experience to know when to use categorical explanatory variables as fixed effects and when as random effects. Some guidelines are given below.

- Am I interested in the effect sizes? Yes means fixed effects.
- Is it reasonable to suppose that the factor levels come from a population of levels? Yes means random effects.
- Are there enough levels of the factor in the dataframe on which to base an estimate of the variance of the population of effects? No means fixed effects.

- Are the factor levels informative? Yes means fixed effects.
- Are the factor levels just numeric labels? Yes means random effects.
- Am I mostly interested in making inferences about the distribution of effects, based on the random sample of effects represented in the dataframe? Yes means random effects.
- Is there hierarchical structure? Yes means you need to ask whether the data are experimental or observations.
- Is it a hierarchical experiment, where the factor levels are experimental manipulations? Yes means fixed effects in a split-plot design (see p. 469)
- Is it a hierarchical observational study? Yes means random effects, perhaps in a variance components analysis (see p. 475).
- When your model contains both fixed and random effects, use mixed-effects models.
- If your model structure is linear, use linear mixed effects, lmer.
- Otherwise, specify the model equation and use non-linear mixed effects, nlme.

Removing the pseudoreplication

The extreme response to pseudoreplication in a data set is simply to eliminate it. Spatial pseudoreplication can be averaged away and temporal pseudoreplication can be dealt with by carrying out carrying out separate ANOVAs, one at each time. This approach has two major weaknesses:

- It cannot address questions about treatment effects that relate to the longitudinal development of the mean response profiles (e.g. differences in growth rates between successive times).
- Inferences made with each of the separate analyses are not independent, and it is not always clear how they should be combined.

Analysis of longitudinal data

The key feature of longitudinal data is that the same individuals are measured repeatedly through time. This would represent temporal pseudoreplication if the data were used uncritically in regression or ANOVA. The set of observations on one individual subject will tend to be positively correlated, and this correlation needs to be taken into account in carrying out the analysis. The alternative is a cross-sectional study, with all the data gathered at a single point in time, in which each individual contributes a single data point. The advantage of longitudinal studies is that they are capable of separating *age effects* from *cohort effects*; these are inextricably confounded in cross-sectional studies. This is particularly important when differences between years mean that cohorts originating at different times experience different conditions, so that individuals of the same age in different cohorts would be expected to differ.

There are two extreme cases in longitudinal studies:

- a few measurements on a large number of individuals;
- a large number of measurements on a few individuals.

In the first case it is difficult to fit an accurate model for change within individuals, but treatment effects are likely to be tested effectively. In the second case, it is possible to get an accurate model of the way that individuals change though time, but there is less power for testing the significance of treatment effects, especially if variation from individual to individual is large. In the first case, less attention will be paid to estimating the correlation structure, while in the second case the covariance model will be the principal focus of attention. The aims are:

- to estimate the average time course of a process;
- to characterize the degree of heterogeneity from individual to individual in the rate of the process;
- to identify the factors associated with both of these, including possible cohort effects.

The response is not the individual measurement, but the *sequence of measurements* on an individual subject. This enables us to distinguish between age effects and year effects; see Diggle *et al.* (1994) for details.

Derived variable analysis

The idea here is to get rid of the pseudoreplication by reducing the repeated measures into a set of summary statistics (slopes, intercepts or means), then *analyse these summary statistics* using standard parametric techniques such as ANOVA or regression. The technique is weak when the values of the explanatory variables change through time. Derived variable analysis makes most sense when it is based on the parameters of scientifically interpretable non-linear models from each time sequence. However, the best model from a theoretical perspective may not be the best model from the statistical point of view.

There are three qualitatively different sources of random variation:

- **random effects**, where experimental units differ (e.g. genotype, history, size, physiological condition) so that there are intrinsically high responders and other low responders;
- **serial correlation**, where there may be time-varying stochastic variation within a unit (e.g. market forces, physiology, ecological succession, immunity) so that correlation depends on the time separation of pairs of measurements on the same individual, with correlation weakening with the passage of time;
- **measurement error**, where the assay technique may introduce an element of correlation (e.g. shared bioassay of closely spaced samples; different assay of later specimens).

Variance components analysis

For random effects we are often more interested in the question of how much of the variation in the response variable can be attributed to a given factor, than we are in estimating means or assessing the significance of differences between means. This procedure is called variance components analysis.

The following classic example of pseudoreplication comes from Snedecor Cochran (1980):

```
rats<-read.table("c:\\temp\\rats.txt",header=T)
attach(rats)
names(rats)
```

```
[1] "Glycogen" "Treatment" "Rat" "Liver"
```

Three experimental treatments were administered to rats, and the glycogen content of the rats' livers was analysed as the response variable. There were two rats per treatment, so the total sample was $n = 3 \times 2 = 6$. The tricky bit was that after each rat was killed, its liver was cut up into three pieces: a left-hand bit, a central bit and a right-hand bit. So now there are six rats each producing three bits of liver, for a total of $6 \times 3 = 18$ numbers. Finally, two separate preparations were made from each macerated bit of liver, to assess the measurement error associated with the analytical machinery. At this point there are $2 \times 18 = 36$ numbers in the data frame as a whole. The factor levels are numbers, so we need to declare the explanatory variables to be categorical before we begin:

```
Treatment<-factor(Treatment)
Rat<-factor(Rat)
Liver<-factor(Liver)
```

Here is the analysis done the *wrong* way:

```
model<-aov(Glycogen~Treatment)
summary(model)
```

```
           Df   Sum Sq  Mean Sq  F value     Pr(>F)
Treatment   2  1557.56   778.78   14.498  3.031e–05  ***
Residuals  33  1772.67    53.72
```

Treatment has a highly significant effect on liver glycogen content ($p = 0.000\,03$). This is wrong! We have committed a classic error of pseudoreplication. Look at the error line in the ANOVA table: it says the residuals have 33 degrees of freedom. But there were only 6 rats in the whole experiment, so the error d.f. has to be $6 - 1 - 2 = 3$ (not 33)!

Here is the analysis of variance done properly, averaging away the pseudoreplication:

```
tt<-as.numeric(Treatment)
yv<-tapply(Glycogen,list(Treatment,Rat),mean)

tv<-tapply(tt,list(Treatment,Rat),mean)
model<-aov(as.vector(yv)~factor(as.vector(tv)))
summary(model)
```

```
                       Df   Sum Sq   Mean Sq  F value   Pr(>F)
factor(as.vector(tv))   2  259.593   129.796    2.929   0.1971
Residuals               3  132.944    44.315
```

Now the error degrees of freedom are correct (d.f. = 3, not 33), and the interpretation is completely different: there is no significant differences in liver glycogen under the three experimental treatments ($p = 0.1971$).

There are two different ways of doing the analysis properly in R: ANOVA with multiple error terms (aov) or linear mixed-effects models (lmer). The problem is that the bits of the same liver are pseudoreplicates because they are spatially correlated (they come from the same rat); they are not independent, as required if they are to be true replicates. Likewise, the two preparations from each liver bit are very highly correlated (the livers were macerated before the preparations were taken, so they are essentially the same sample (certainly not independent replicates of the experimental treatments).

Here is the correct analysis using aov with multiple error terms. In the Error term we start with the largest scale (treatment), then rats within treatments, then liver bits within rats within treatments. Finally, there were replicated measurements (two preparations) made for each bit of liver.

```
model2<-aov(Glycogen~Treatment+Error(Treatment/Rat/Liver))
summary(model2)
```

```
Error: Treatment
          Df   Sum Sq  Mean Sq
Treatment  2  1557.56   778.78

Error: Treatment:Rat
          Df  Sum Sq  Mean Sq  F value  Pr(>F)
Residuals  3  797.67   265.89

Error: Treatment:Rat:Liver
          Df  Sum Sq  Mean Sq  F value  Pr(>F)
Residuals 12   594.0     49.5

Error: Within
          Df  Sum Sq  Mean Sq  F value  Pr(>F)
Residuals 18  381.00    21.17
```

You can do the correct, non-pseudoreplicated analysis of variance from this output (Box 11.2).

Box 11.2 Sums of squares in hierarchical designs

The trick to understanding these sums of squares is to appreciate that with nested categorical explanatory variables (random effects) the correction factor, which is subtracted from the sum of squared subtotals, is *not* the conventional $(\sum y)^2/kn$. Instead, the correction factor is the uncorrected sum of squared subtotals from the level in the hierarchy immediately above the level in question. This is very hard to see without lots of practice. The total sum of squares, *SSY*, and the treatment sum of squares, *SSA*, are computed in the usual way (see Box 11.1):

$$SSY = \sum y^2 - \frac{(\sum y)^2}{n},$$

$$SSA = \frac{\sum_{i=1}^{k} C_i^2}{n} - \frac{(\sum y)^2}{kn}.$$

The analysis is easiest to understand in the context of an example. For the rats data, the treatment totals were based on 12 numbers (two rats, three liver bits per rat and two preparations per liver bit). In this case, in the formula for *SSA*, above, $n = 12$ and $kn = 36$. We need to calculate sums of squares for rats within treatments, SS_{Rats}, liver bits within rats within treatments, $SS_{\text{Liverbits}}$, and preparations within liver bits within rats within treatments, $SS_{\text{Preparations}}$:

$$SS_{\text{Rats}} = \frac{\sum R^2}{6} - \frac{\sum C^2}{12},$$

$$SS_{\text{Liverbits}} = \frac{\sum L^2}{2} - \frac{\sum R^2}{6},$$

$$SS_{\text{Preparations}} = \frac{\sum y^2}{1} - \frac{\sum L^2}{2}.$$

The correction factor at any level is the *uncorrected sum of squares from the level above*. The last sum of squares could have been computed by difference:

$$SS_{\text{Preparations}} = SSY - SSA - SS_{\text{Rats}} - SS_{\text{Liverbits}}.$$

The F test for equality of the treatment means is the treatment variance divided by the 'rats within treatment variance' from the row immediately beneath: $F = 778.78/265.89 = 2.928\,956$, with 2 d.f. in the numerator and 3 d.f. in the denominator (as we obtained in the correct ANOVA, above).

To turn this into a variance components analysis we need to do a little work. The mean squares are converted into variance components like this:

Residuals = preparations within liver bits: unchanged = 21.17,

Liver bits within rats within treatments: $(49.5 - 21.17)/2 = 14.165$,

Rats within treatments: $(265.89 - 49.5)/6 = 36.065$.

You divide the difference in variance by the number of numbers in the level below (i.e. two preparations per liver bit, and six preparations per rat, in this case).

Analysis of the rats data using lmer is explained on p. 648.

What is the difference between split-plot and hierarchical samples?

Split-plot experiments have informative factor levels. Hierarchical samples have uninformative factor levels. That's the distinction. In the irrigation experiment, the factor levels were as follows:

```
levels(density)
```

```
[1] "high" "low" "medium"
```

```
levels(fertilizer)
```

```
[1] "N" "NP" "P"
```

They show the density of seed sown, and the kind of fertilizer applied: they are informative. Here are the factor levels from the rats experiment:

```
levels(Rat)
```

```
[1] "1" "2"
```

```
levels(Liver)
```

```
[1] "1" "2" "3"
```

These factor levels are uninformative, because rat number 2 in treatment 1 has nothing in common with rat number 2 in treatment 2, or with rat number 2 in treatment 3. Liver bit number 3 from rat 1 has nothing in common with liver bit number 3 from rat 2. Note, however, that numbered factor levels are *not* always uninformative: treatment levels 1, 2, and 3 are informative: 1 is the control, 2 is a diet supplement, and 3 is a combination of two supplements.

When the factor levels are informative, the variable is known as a *fixed effect*. When the factor levels are uninformative, the variable is known as a *random effect*. Generally, we are interested in fixed effects as they influence the mean, and in random effects as they influence the variance. We tend not to speak of effect sizes attributable to random effects, but effect sizes and their standard errors are often the principal focus when we have fixed effects. Thus, irrigation, density and fertilizer are fixed effects, and rat and liver bit are random effects.

ANOVA with aov or lm

The difference between lm and aov is mainly in the form of the output: the summary table with aov is in the traditional form for analysis of variance, with one row for each categorical variable and each interaction term. On the other hand, the summary table for lm produces one row per estimated parameter (i.e. one row for each factor level and one row for each interaction level). If you have multiple error terms then you must use aov because lm does not support the Error term. Here is the same two-way analysis of variance fitted using aov first then using lm:

```
daphnia<-read.table("c:\\temp\\Daphnia.txt",header=T)
attach(daphnia)
names(daphnia)
```

```
[1] "Growth.rate" "Water" "Detergent" "Daphnia"
```

```
model1<-aov(Growth.rate~Water*Detergent*Daphnia)
summary(model1)
```

```
                         Df  Sum Sq  Mean Sq  F value     Pr(>F)
Water                     1   1.985    1.985   2.8504  0.0978380    .
Detergent                 3   2.212    0.737   1.0586  0.3754783
Daphnia                   2  39.178   19.589  28.1283  8.228e-09  ***
Water:Detergent           3   0.175    0.058   0.0837  0.9686075
Water:Daphnia             2  13.732    6.866   9.8591  0.0002587  ***
Detergent:Daphnia         6  20.601    3.433   4.9302  0.0005323  ***
Water:Detergent:Daphnia   6   5.848    0.975   1.3995  0.2343235
Residuals                48  33.428    0.696
```

```
model2<-lm(Growth.rate~Water*Detergent*Daphnia)
summary(model2)
```

```
Coefficients:
                                          Estimate  Std. Error  t value  Pr (>|t|)
(Intercept)                                2.81126     0.48181    5.835   4.48e-07
WaterWear                                 −0.15808     0.68138   −0.232    0.81753
DetergentBrandB                           −0.03536     0.68138   −0.052    0.95883
DetergentBrandC                            0.47626     0.68138    0.699    0.48794
DetergentBrandD                           −0.21407     0.68138   −0.314    0.75475
DaphniaClone2                              0.49637     0.68138    0.728    0.46986
DaphniaClone3                              2.05526     0.68138    3.016    0.00408
WaterWear:DetergentBrandB                  0.46455     0.96361    0.482    0.63193
WaterWear:DetergentBrandC                 −0.27431     0.96361   −0.285    0.77712
WaterWear:DetergentBrandD                  0.21729     0.96361    0.225    0.82255
WaterWear:DaphniaClone2                    1.38081     0.96361    1.433    0.15835
WaterWear:DaphniaClone3                    0.43156     0.96361    0.448    0.65627
DetergentBrandB:DaphniaClone2              0.91892     0.96361    0.954    0.34506
DetergentBrandC:DaphniaClone2             −0.16337     0.96361   −0.170    0.86609
DetergentBrandD:DaphniaClone2              1.01209     0.96361    1.050    0.29884
DetergentBrandB:DaphniaClone3             −0.06490     0.96361   −0.067    0.94658
DetergentBrandC:DaphniaClone3             −0.80789     0.96361   −0.838    0.40597
DetergentBrandD:DaphniaClone3             −1.28669     0.96361   −1.335    0.18809
WaterWear:DetergentBrandB:DaphniaClone2   −1.26380     1.36275   −0.927    0.35837
WaterWear:DetergentBrandC:DaphniaClone2    1.35612     1.36275    0.995    0.32466
WaterWear:DetergentBrandD:DaphniaClone2    0.77616     1.36275    0.570    0.57164
WaterWear:DetergentBrandB:DaphniaClone3   −0.87443     1.36275   −0.642    0.52414
WaterWear:DetergentBrandC:DaphniaClone3   −1.03019     1.36275   −0.756    0.45337
WaterWear:DetergentBrandD:DaphniaClone3   −1.55400     1.36275   −1.140    0.25980

Residual standard error: 0.8345 on 48 degrees of freedom
Multiple R-Squared: 0.7147, Adjusted R-squared: 0.578
F-statistic: 5.227 on 23 and 48 DF, p-value: 7.019e-07
```

Note that two significant interactions, Water–Daphnia and Detergent–Daphnia, show up in the aov table but not in the lm summary (this is often due to the fact that the lm summary shows treatment contrasts rather than Helmert contrasts). This draws attention to the importance of model simplification rather than per-row *t* tests (i.e. removing the non-significant three-way interaction term in this case). In the aov table, the *p* values are 'on deletion' *p* values, which is a big advantage.

The main difference is that there are eight rows in the aov summary table (three main effects, three two-way interactions, one three-way interaction and an error term) but there are 24 rows in the lm summary table (four levels of detergent by three levels of daphnia clone by two levels of water). You can easily view the output of model1 in linear model layout, or model2 as an ANOVA table using the summary options .lm or .aov:

```
summary.lm(model1)
summary.aov(model2)
```

Effect Sizes

In complicated designed experiments, it is easiest to summarize the effect sizes with the model.tables function. This takes the name of the fitted model object as its first argument, and you can specify whether you want the standard errors (as you typically would):

```
model.tables(model1, "means", se = TRUE)
```

```
Tables of means
Grand mean

3.851905

  Water

Water
 Tyne   Wear
3.686  4.018

  Detergent

Detergent
BrandA  BrandB  BrandC  BrandD
 3.885   4.010   3.955   3.558

  Daphnia
Daphnia
Clone1  Clone2  Clone3
 2.840   4.577   4.139

Water:Detergent

      Detergent
Water  BrandA  BrandB  BrandC  BrandD
 Tyne   3.662   3.911   3.814   3.356
 Wear   4.108   4.109   4.095   3.760

Water:Daphnia

      Daphnia
Water  Clone1  Clone2  Clone3
 Tyne   2.868   3.806   4.383
 Wear   2.812   5.348   3.894

Detergent:Daphnia

Daphnia
Detergent  Clone1  Clone2  Clone3
   BrandA   2.732   3.919   5.003
   BrandB   2.929   4.403   4.698
   BrandC   3.071   4.773   4.019
   BrandD   2.627   5.214   2.834

Water:Detergent:Daphnia
, , Daphnia = Clone1

Detergent
Water  BrandA  BrandB  BrandC  BrandD
 Tyne   2.811   2.776   3.288   2.597
 Wear   2.653   3.082   2.855   2.656

, , Daphnia = Clone2

Detergent
Water  BrandA  BrandB  BrandC  BrandD
 Tyne   3.308   4.191   3.621   4.106
 Wear   4.530   4.615   5.925   6.322

, , Daphnia = Clone3
```

```
Detergent
Water  BrandA  BrandB  BrandC  BrandD
 Tyne   4.867   4.766   4.535   3.366
 Wear   5.140   4.630   3.504   2.303

Standard errors for differences of means
          Water  Detergent  Daphnia  Water:Detergent  Water:Daphnia
         0.1967     0.2782   0.2409           0.3934         0.3407
replic.      36         18       24                9             12
         Detergent:Daphnia  Water:Detergent:Daphnia
                    0.4818                   0.6814
replic.                  6                        3
```

Note that the standard errors are standard errors of differences, and they are different in each of the different strata because the replication differs. All standard errors use the same pooled error variance $s^2 = 0.696$ (see above). For instance, the three-way interactions have $se = \sqrt{2 \times 0.696/3} = 0.681$ and the daphnia main effects have $se = \sqrt{2 \times 0.696/24} = 0.2409$.

Attractive plots of effect sizes can be obtained using the effects library (p. 178).

Replications

The replications function allows you to check the number of replicates at each level in an experimental design:

```
replications(Growth.rate~Daphnia*Water*Detergent,daphnia)
```

```
                               Daphnia              Water
Detergent
                                    24                 36
18
                         Daphnia:Water  Daphnia:Detergent
Water:Detergent
                                    12                  6
9
Daphnia:Water:Detergent
                                     3
```

There are three replicates for the three-way interaction and for all of the two-way interactions (you need to remember the number of levels for each factor to see this: there are two water types, three daphnia clones and four detergents (see above).

Multiple Comparisons

When comparing the means for the levels of a factor in an analysis of variance, a simple comparison using multiple t tests will inflate the probability of declaring a significant difference when there is none. This because the intervals are calculated with a given coverage probability for each *interval* but the interpretation of the coverage is usually with respect to *the entire family of intervals* (i.e. for the factor as a whole).

If you follow the protocol of model simplification recommended in this book, then issues of multiple comparisons will not arise very often. An occasional significant t test amongst a bunch of non-significant interaction terms is not likely to survive a deletion test (see p. 325). Again, if you have factors with large numbers of levels you might consider using mixed-effects models rather than ANOVA (i.e. treating the factors as random effects rather than fixed effects; see p. 627).

John Tukey introduced intervals based on the range of the sample means rather than the individual differences; nowadays, these are called Tukey's honest significant differences. The intervals returned by the TukeyHSD function are based on Studentized range statistics. Technically the intervals constructed in this way would only apply to balanced designs where the same number of observations is made at each level of the factor. This function incorporates an adjustment for sample size that produces sensible intervals for mildly unbalanced designs.

The following example concerns the yield of fungi gathered from 16 different habitats:

```
data<-read.table("c:\\temp\\Fungi.txt",header=T)
attach(data)
names(data)
```

First we establish whether there is any variation in fungus yield to explain:

```
model<-aov(Fungus.yield~Habitat)
summary(model)
```

```
           Df  Sum Sq  Mean Sq  F value     Pr(>F)
Habitat    15  7527.4    501.8   72.141  < 2.2e-16  ***
Residuals 144  1001.7      7.0
```

Yes, there is ($p < 0.000\,001$). But this is not of much real interest, because it just shows that some habitats produce more fungi than others. We are likely to be interested in *which* habitats produce significantly more fungi than others. Multiple comparisons are an issue because there are 16 habitats and so there are $(16 \times 15)/2 = 120$ possible pairwise comparisons. There are two options:

- apply the function TukeyHSD to the model to get Tukey's honest significant differences;
- use the function pairwise.t.test to get adjusted p values for all comparisons.

Here is Tukey's test in action: it produces a table of p values by default:

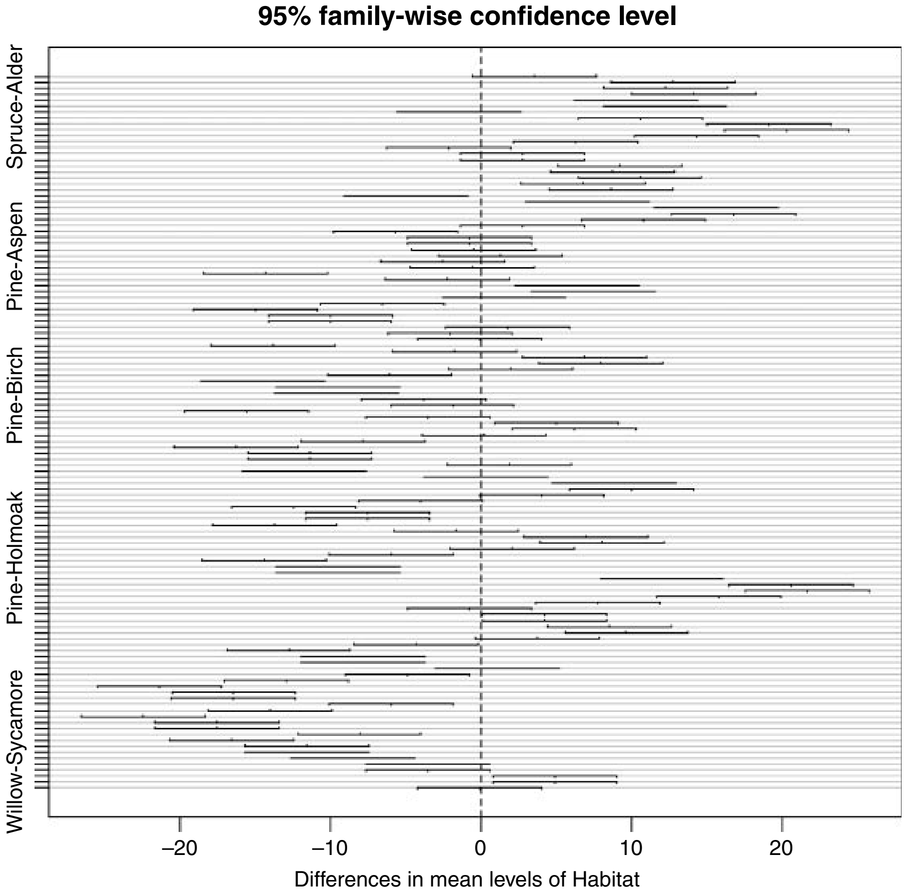
```
TukeyHSD(model)
```

```
Tukey multiple comparisons of means
 95% family-wise confidence level

Fit: aov(formula = Fungus.yield ~ Habitat)
$Habitat
                     diff         lwr        upr      p adj
Ash-Alder        3.53292777  -0.5808096   7.6466651  0.1844088
Aspen-Alder     12.78574402   8.6720067  16.8994814  0.0000000
Beech-Alder     12.32365349   8.2099161  16.4373908  0.0000000
Birch-Alder     14.11348150   9.9997441  18.2272189  0.0000000
Cherry-Alder    10.29508769   6.1813503  14.4088250  0.0000000
Chestnut-Alder  12.24107899   8.1273416  16.3548163  0.0000000
Holmoak-Alder   -1.44360558  -5.5573429   2.6701318  0.9975654
Hornbeam-Alder  10.60271044   6.4889731  14.7164478  0.0000000
Lime-Alder      19.19458205  15.0808447  23.3083194  0.0000000
Oak-Alder       20.29457340  16.1808360  24.4083108  0.0000000
Pine-Alder      14.34084715  10.2271098  18.4545845  0.0000000
```

```
Rowan-Alder       6.29495226   2.1812149  10.4086896  0.0000410
Spruce-Alder     −2.15119456  −6.2649319   1.9625428  0.9036592
Sycamore-Alder    2.80900108  −1.3047363   6.9227384  0.5644643
...
Spruce-Rowan     −8.44614681 −12.5598842  −4.3324095  0.0000000
Sycamore-Rowan   −3.48595118  −7.5996885   0.6277862  0.2019434
Willow-Rowan     −3.51860059  −7.6323379   0.5951368  0.1896363
Sycamore-Spruce   4.96019563   0.8464583   9.0739330  0.0044944
Willow-Spruce     4.92754623   0.8138089   9.0412836  0.0049788
Willow-Sycamore  −0.03264941  −4.1463868   4.0810879  1.0000000
```

You can plot the confidence intervals if you prefer (or do both, of course):

```
plot(TukeyHSD(model))
```



Habitats on opposite sides of the dotted line and not overlapping it are significantly different from one another.

Alternatively, you can use the pairwise.t.test function in which you specify the response variable, and then the categorical explanatory variable containing the factor levels you want to be compared, separated by a comma (not a tilde):

```
pairwise.t.test(Fungus.yield,Habitat)
```

```
         Pairwise comparisons using t tests with pooled SD
data: Fungus.yield and Habitat
         Alder    Ash      Aspen    Beech    Birch    Cherry   Chestnut Holmoak
Ash      0.10011  -        -        -        -        -        -        -
Aspen    < 2e-16  6.3e-11  -        -        -        -        -        -
Beech    < 2e-16  5.4e-10  1.00000  -        -        -        -        -
Birch    < 2e-16  1.2e-13  1.00000  1.00000  -        -        -        -
Cherry   4.7e-13  2.9e-06  0.87474  1.00000  0.04943  -        -        -
Chestnut < 2e-16  7.8e-10  1.00000  1.00000  1.00000  1.00000  -        -
Holmoak  1.00000  0.00181  < 2e-16  < 2e-16  < 2e-16  3.9e-16  < 2e-16  -
Hornbeam 1.1e-13  8.6e-07  1.00000  1.00000  0.10057  1.00000  1.00000  < 2e-16
Lime     < 2e-16  < 2e-16  1.1e-05  1.9e-06  0.00131  3.3e-10  1.4e-06  < 2e-16
Oak      < 2e-16  < 2e-16  1.4e-07  2.0e-08  2.7e-05  1.9e-12  1.5e-08  < 2e-16
Pine     < 2e-16  3.9e-14  1.00000  1.00000  1.00000  0.02757  1.00000  < 2e-16
Rowan    1.8e-05  0.51826  8.5e-06  4.7e-05  3.9e-08  0.03053  6.2e-05  5.3e-08
Spruce   1.00000  0.00016  < 2e-16  < 2e-16  < 2e-16  < 2e-16  < 2e-16  1.00000
Sycamore 0.50084  1.00000  2.1e-12  1.9e-11  3.3e-15  1.5e-07  2.7e-11  0.01586
Willow   0.51826  1.00000  1.9e-12  1.6e-11  2.8e-15  1.4e-07  2.4e-11  0.01702
         Hornbeam Lime     Oak      Pine     Rowan    Spruce   Sycamore
Ash      -        -        -        -        -        -        -
Aspen    -        -        -        -        -        -        -
Beech    -        -        -        -        -        -        -
Birch    -        -        -        -        -        -        -
Cherry   -        -        -        -        -        -        -
Chestnut -        -        -        -        -        -        -
Holmoak  -        -        -        -        -        -        -
Hornbeam -        -        -        -        -        -        -
Lime     1.3e-09  -        -        -        -        -        -
Oak      8.4e-12  1.00000  -        -        -        -        -
Pine     0.05975  0.00253  6.1e-05  -        -        -        -
Rowan    0.01380  < 2e-16  < 2e-16  1.5e-08  -        -        -
Spruce   < 2e-16  < 2e-16  < 2e-16  < 2e-16  2.5e-09  -        -
Sycamore 4.2e-08  < 2e-16  < 2e-16  1.1e-15  0.10218  0.00187  -
Willow   3.8e-08  < 2e-16  < 2e-16  9.3e-16  0.10057  0.00203  1.00000
P value adjustment method: holm
```

As you see, the default method of adjustment of the *p*-values is holm, but other adjustment methods include hochberg, hommel, bonferroni, BH, BY, fdr and none. Without adjustment of the *p* values, the rowan–willow comparison looks highly significant ($p = 0.00335$), as you can see if you try

```
pairwise.t.test(Fungus.yield,Habitat,p.adjust.method="none")
```

I like TukeyHSD because it is conservative without being ridiculously so (in contrast to Bonferroni). For instance, Tukey gives the birch–cherry comparison as non-significant ($p = 0.1011027$) while Holm makes this difference significant ($p = 0.04943$). Tukey had Willow-Holm Oak as significant ($p = 0.0380910$), whereas Bonferroni throws this baby out with the bathwater ($p = 0.05672$). You need to decide how circumspect you want to be in the context of your particular question.

There is a useful package for multiple comparisons called multcomp:

```
install.packages("multcomp")
```

You can see at once how contentious the issue of multiple comparisons is, just by looking at the length of the list of different multiple comparisons methods supported in this package

- the many-to-one comparisons of Dunnett
- the all-pairwise comparisons of Tukey
- Sequen
- AVE
- changepoint
- Williams
- Marcus
- McDermott
- Tetrade
- Bonferroni correction
- Holm
- Hochberg
- Hommel
- Benjamini–Hochberg
- Benjamini–Yekutieli

The old-fashioned Bonferroni correction is highly conservative, because the *p* values are multiplied by the number of comparisons. Instead of using the usual Bonferroni and Holm procedures, the adjustment methods include less conservative corrections that take the exact correlations between the test statistics into account by use of the multivariate *t*-distribution. The resulting procedures are therefore substantially more powerful (the Bonferroni and Holm adjusted *p* values are reported for reference). There seems to be no reason to use the unmodified Bonferroni correction because it is dominated by Holm's method, which is valid under arbitrary assumptions.

The tests are designed to suit multiple comparisons within the general linear model, so they allow for covariates, nested effects, correlated means and missing values. The first four methods are designed to give strong control of the familywise error rate. The methods of Benjamini, Hochberg, and Yekutieli control the false discovery rate, which is the expected proportion of false discoveries amongst the rejected hypotheses. The false discovery rate is a less stringent condition than the familywise error rate, so these methods are more powerful than the others.

Projections of Models

If you want to see how the different factor levels contribute their additive effects to each of the observed values of the response, then use the proj function like this:

```
library(help="multcomp")
```

```
              (Intercept)       Water   Detergent     Daphnia  Water:Detergent
Water:Daphnia
1                3.851905  −0.1660431  0.03292724  −1.0120302      −0.05698158
0.1941404
2                3.851905  −0.1660431  0.03292724  −1.0120302      −0.05698158
0.1941404
3                3.851905  −0.1660431  0.03292724  −1.0120302      −0.05698158
0.1941404
...
```

The name proj comes from the fact that the function returns a matrix or list of matrices giving the 'projections of the data onto the terms of a linear model'.

Multivariate Analysis of Variance

Two or more response variables are sometimes measured in the same experiment. Of course you can analyse each response variable separately, and that is the typical way to proceed. But there are occasions where you want to treat the group of response variables as one multivariate response. The function for this is manova, the multivariate analysis of variance. Note that manova does not support multi-stratum analysis of variance, so the formula must not include an Error term.

```
data<-read.table("c:\\temp\\manova.txt",header=T)
attach(data)
names(data)
```

```
[1] "tear" "gloss" "opacity" "rate" "additive"
```

First, create a multivariate response variable, Y, by binding together the three separate response variables (tear, gloss and opacity), like this:

```
Y <- cbind(tear, gloss, opacity)
```

Then fit the multivariate analysis of variance using the manova function:

```
model<-manova(Y~rate*additive)
```

There are two ways to inspect the output. First, as a multivariate analysis of variance:

```
summary(model)
```

```
               Df  Pillai  approx F  num Df  den Df    Pr(>F)
rate            1  0.6181    7.5543       3      14  0.003034  **
additive        1  0.4770    4.2556       3      14  0.024745   *
rate:additive   1  0.2229    1.3385       3      14  0.301782
Residuals      16
```

This shows significant main effects for both rate and additive, but no interaction. Note that the F tests are based on 3 and 14 degrees of freedom (not 1 and 16). The default method in summary.manova is the Pillai–Bartlett statistic. Other options include Wilks, Hotelling–Lawley and Roy. Second, you will want to look at each of the three response variables separately:

```
summary.aov(model)
```

```
Response tear :
                 Df   Sum Sq  Mean Sq  F value    Pr(>F)
rate              1  1.74050  1.74050  15.7868  0.001092  **
additive          1  0.76050  0.76050   6.8980  0.018330   *
rate:additive     1  0.00050  0.00050   0.0045  0.947143
Residuals        16  1.76400  0.11025

Response gloss :
                 Df   Sum Sq  Mean Sq  F value   Pr(>F)
rate              1  1.30050  1.30050   7.9178  0.01248  *
additive          1  0.61250  0.61250   3.7291  0.07139  .
rate:additive     1  0.54450  0.54450   3.3151  0.08740  .
Residuals        16  2.62800  0.16425

Response opacity :
                 Df  Sum Sq  Mean Sq  F value  Pr(>F)
rate              1   0.421    0.421   0.1036  0.7517
additive          1   4.901    4.901   1.2077  0.2881
rate:additive     1   3.961    3.961   0.9760  0.3379
Residuals        16  64.924    4.058
```

Notice that one of the three response variables, opacity, is not significantly associated with either of the explanatory variables.