STATISTICS: AN INTRODUCTION USING R

By M.J. Crawley

Exercises

7. NESTED ANALYSIS & SPLIT PLOT DESIGNS

Up to this point, we have treated all categorical explanatory variables as if they were the same. This is certainly what R.A. Fisher had in mind when he invented the analysis of variance in the 1920’s and 30’s. It was Eisenhart (1947) who realised that there were actually 2 fundamentally different sorts of categorical explanatory variables: somewhat confusingly he called these **fixed effects** and **random effects**. The distinction is best seen by an example. In most mammal species the categorical variable ‘sex’ has two levels: “male” and “female”. For any individual that you find, the knowledge that it is, say, female conveys a great deal of information about the individual, and this information draws on experience gleaned from many other individuals that were female. A female will have a whole set of attributes (associated with her being female) no matter what population that individual was drawn from. Take a different categorical variable like genotype. If we have two genotypes in a population we might label them A and B. If we take two more genotypes from a *different* population we might label them A and B as well. In a case like this, the label A does not convey any information at all about the genotype, other than that it is different from Genotype B. In the case of sex, the factor level (male or female) is informative: sex is a fixed effect. In the case of genotype, the factor level (A or B) is uninformative: genotype is a random effect. Random effects have factor levels that are drawn from a large (potentially very large) population in which the individuals differ in many ways, but *we do not know exactly how or why they differ*. In the case of sex we know that males and females are likely to differ in characteristic and predictable ways. To get a feel for the difference between fixed effects and random effects here are some more examples:

Fixed Effects	**Random effects**
Drug administered or not	Genotype
Insecticide sprayed or not	Brood
Nutrient added or not	Block within a field
One country versus another	Split plot within a plot
Male or female	History of development
Wet versus dry	Untreated individuals
Light versus shade	Family
One age versus another	Parent

There are three main parts to this practical:

- Nested Designs
- Designed Split-Plot Experiments
- Mixed Effects Models

They are linked by two facts: (1) they involve categorical variables of two kinds (fixed effects and random effects); and (2) because their data frames all involve pseudoreplication, they offer great scope for getting the analysis wrong.

Model I and Model II anova

The distinction between the two models of anova was made by Eisenhart in 1947. In Model I, the differences between the means are ascribed entirely to the fixed treatment effects, so any data point can be decomposed like this:

$$y_{ij} = \mu + \alpha_i + \varepsilon_{ij}$$

It is the overall mean μ plus a treatment effect due to the *i*th level of the fixed effect α_i plus a deviation ε_{ij} that is unique to that individual and is drawn from a Normal distribution with mean 0 and variance σ^2.

Model II is subtly different. Here, the model is written

$$y_{ij} = \mu + U_i + \varepsilon_{ij}$$

which looks as if it is *exactly* the same as Model I, just replacing α_i by U_i. The distinction is this. In Model I the experimenter either *made* the treatments like they were (e.g. by the random allocation of treatments to individuals), or selected the categories because they were fixed and clearly distinctive (e.g. a comparison of individual fossils from the Jurassic and Carboniferous periods). In Model II the factor levels are different from one another, but the experimenter did not *make* them different. They were selected (perhaps *because* they were different, perhaps not), but they came from a much larger pool of factor levels that exhibits variation beyond the control of the experimenter and beyond their immediate interest (e.g. 6 genotypes were selected to work with, out of a pool of, who knows, perhaps many 1000's of genotypes). We call it random variation, and call such factors random effects.

The important point is that because the random effects U_i come from a large population, there is not much point in concentrating on estimating means of our small subset, *a*, of factor levels, and no point at all in comparing individual pairs of means for different factor levels. Much better to recognise them for what they are, random samples from a much larger population, and to concentrate on their variance σ_U^2. This is the *added* variation caused by differences between the levels of the random effects. Model II anova is all about estimating the size of this variance, and working out its percentage contribution to the overall variation.

The issues involved in Model II work fall into 2 broad categories

- questions about experimental design and the management of experimental error (e.g. where does most of the variation occur, and where would increased replication be most profitable?)

- questions about hierarchical structure, and the relative magnitude of variation at different levels within the hierarchy (e.g. studies on the genetics of individuals within families, families within parishes, and parishes with counties, to discover the relative importance of genetic and phenotypic variation).

Nested Analysis

Most anova models are based on the assumption that there is a single error term. But in nested experiments like split-plot designs, where the data are gathered at two or more different spatial scales, there is *a different error variance for each different plot size*.

We take an example from Sokal & Rohlf (1981). The experiment involved a simple one-factor anova with 3 treatments given to 6 rats. The analysis was complicated by the fact that three preparations were taken from the liver of each rat, and two readings of glycogen content were taken from each preparation. This generated 6 pseudoreplicates per rat to give a total of 36 readings in all. Clearly, it would be a mistake to analyse these data as if they were a straightforward one-way anova, because that would give us 33 degrees of freedom for error. In fact, since there are only two rats in each treatment, we have only one degree of freedom per treatment, giving a total of 3 d.f. for error.

The variance is likely to be different at each level of this nested analysis because:

- the readings differ because of variation in the glycogen detection method within each liver sample (measurement error)
- the pieces of liver may differ because of heterogeneity in the distribution of glycogen within the liver of a single rat
- the rats will differ from one another in their glycogen levels because of sex, age, size, genotype, etc.
- rats allocated different experimental treatments may differ as a result of the fixed effects of treatment.

If all we want to test is whether the experimental treatments have affected the glycogen levels, then we are not interested in liver bits within rat's livers, or in preparations within liver bits. We could add all the pseudoreplicates together, and analyse the 6 averages. This would have the virtue of showing what a tiny experiment this really was (we do this later; see below). But to analyse the full data set, we must proceed as follows.

The only trick is to ensure that the factor levels are set up properly. There were 3 treatments, so we make a treatment factor *T* with 3 levels. While there were 6 rats in total, there were only 2 in each treatment, so we declare rats as a factor *R* with 2 levels (not 6). There were 18 bits of liver in all, but only 3 per rat, so we declare liver-bits as a factor *L* with 3 levels (not 18).

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

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

```
tapply(Glycogen,Treatment,mean)
```

```
       1        2        3 
140.5000 151.0000 135.1667 
```

There are substantial differences between the treatment means, and our job is to say whether these differences are statistically significant or not. Because the 3 factors Treatment, Rat and Liver have numeric factor levels, we must declare them to be factors before beginning the modelling.

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

Because the sums of squares in nested designs are so confusing on first acquaintance, it is important to work through a simple example like this one by hand. The first two steps are easy, because they relate to the **fixed effect**, which is **Treatment**. We calculate SST and SSA in the usual way (see Practical 5, ANOVA). We need the sum of squares and the grand total of the response variable Glycogen

```
sum(Glycogen);sum(Glycogen^2)
```

```
[1] 5120
[1] 731508
```

$$SST = \sum y^2 - \frac{\left[\sum y\right]^2}{36}$$

SST = 731508-5120^2/36 = 3330.222

For the treatment sum of squares, SSA, we need the 3 treatment totals

```
tapply(Glycogen,Treatment,sum)
```

```
   1    2    3 
1686 1812 1622
```

Each of these was the sum of 12 numbers (2 preparations x 3 liver bits x 2 rats) so we divide the square of each subtotal by 12 before subtracting the correction factor

$$SSA = \frac{\sum T^2}{12} - \frac{\left[\sum y\right]^2}{36}$$

```
sum(tapply(Glycogen,Treatment,sum)^2)/12-5120^2/36
```

```
[1] 1557.556
```

So the Error sums of squares must be SST – SSA = 3330.222-1557.556 = 1772.666. If this is correct, then the ANOVA Table must be like this:

Source	SS	d.f.	MS	F	Critical F
Treatment	1557.556	2	778.778	14.497	3.28
Error	1772.66	33	53.717		
Total	3330.222	35			

```
qf(.95,2,33)
```

```
[1] 3.284918
```

The calculated value is much larger than the value in tables, so treatment has a highly significant effect on liver glycogen content. **Wrong !** We have made the classic mistake of pseudoreplication. We have counted all 36 data points as if they were replicates. The definition of replicates is that *replicates are independent of one another*. Two measurements from the same piece of rat's liver are clearly not independent. Nor are measures from three regions of the same rat liver. It is the rats that are the replicates in this experiment and there are only 6 of them in total ! So the correct total degrees of freedom is 5, not 35, and *there are 5-2 = 3 degrees of freedom for error, not 33*.

There are lots of ways of doing this analysis wrong, but only one way of doing it right. There are 3 spatial scales to the measurements (rats, liver bits from each rat and preparations from each liver bit) and hence there must be 3 different error variances in the analysis. Our first task is to compute the sum of squares for differences between the rats. It is easy to find the sums:

```
tapply(Glycogen,list(Treatment,Rat),sum)
```

```
    1   2
1 795 891
2 898 914
3 806 816
```

Note the use of Treatment and Rat to get the totals: it is **wrong** to do the following:

```
tapply(Glycogen,Rat,sum)
```

```
   1    2
2499 2621
```

because the rats are numbered 1 and 2 within each treatment. **This is very important**. There are 6 rats in the experiment, so we need 6 rat totals. Each rat total is the sum of 6 numbers (3 liver bits and 2 preparations per liver bit). So we square the 6 rat totals, add them up and divide by 6:

```
sum(tapply(Glycogen,list(Treatment,Rat),sum)^2)/6
```

```
[1] 730533
```

But what now? Our experience so far tells us simply to subtract the correction factor. But this is wrong. We use *the sum of squares from the spatial scale above* as the correction term at any given level. This may become clearer from the worked example.

What about the sum of squares for liver bits? There are 3 per rat, so there must be 18 liver-bit totals in all. We can inspect them like this:

```
tapply(Glycogen,list(Treatment,Rat,Liver),sum)
```

261	298	256	283	278	310
302	306	296	294	300	314
259	278	276	277	271	261

We need to square these, add them up and divide by 2. Why 2 ? Because each of these totals is the sum of the 2 measurements of each preparation from each liver bit.

```
sum(tapply(Glycogen,list(Treatment,Rat,Liver),sum)^2)/2
```

```
[1] 731127
```

So we have the following uncorrected sums of squares

Source	**Uncorrected SS**

Treatment	729735.3
Rats	730533
Liver bits	731127
Total	731508

The key to understanding nested designs is to understand the next step

Up to now, we have used the correction factor

$$CF = \frac{\left[\sum y\right]^2}{\sum n}$$

to determine the corrected sum of squares in all cases. We used it (correctly) to determine the treatment sum of squares, earlier in this example. With nested factors, however, we don't do this. **We use the uncorrected sum of squares from the next spatial scale larger than the scale in question**.

$$SS_{Rats} = \frac{\sum R^2}{6} - \frac{\sum T^2}{12}$$

Where R is a rat total and T is a treatment total. The sum of squares of rat totals is divided by 6 because each total was the sum of 6 numbers. The sum of squares of treatment totals is divided by 12 because each treatment total was the sum of 12 numbers (2 rats each generating 6 numbers). So, if L is the sum of the 2 liver-bit preparations, and y is an individual preparation (the "sum of 1 numbers" if you like), we get

$$SS_{Liver.Bits} = \frac{\sum L^2}{2} - \frac{\sum R^2}{6}$$

$$SS_{P.reparations} = \frac{\sum y^2}{1} - \frac{\sum L^2}{2}$$

We can now compute the numerical values of the nested sums of squares:

$$SS_{Rats} = 730533 - 729735.3 = 797.7$$

$$SS_{LiveBits} = 731127 - 730533 = 594.0$$

$$SS_{P.reparations} = 731508 - 731127 = 381.0$$

So now we can fill in the ANOVA table correctly, taking account of the nesting and the pseudoreplication.

Source	SS	d.f.	MS	F	Critical F
Treatment	1557.556	2	778.778	2.929	9.552094
Rats in Treatments	797.7	3	265.9	5.372	3.490295
Liver bits in Rats	594.0	12	49.5	2.339	2.342067
Readings in Liver bits	381.0	18	21.1666		
Total	3330.222	35			

There are several important things to see in this Anova table.

- First, you use the mean square from the spatial scale immediately below in testing significance. (You do **not** use 21.166 as we have done up to now). So the F ratio for treatment is 778.778/265.9 = 2.929 on 2 and 3 d.f. which is way short of significance (the critical value is 9.55; compare this with what we did wrong, earlier).
- Second, a different recipe is used in nested designs to compute the degrees of freedom. Treatment and Total degrees of freedom are the same as usual but the others are different
- There are 6 rats in total but there are not 5 d.f. of freedom for rats: there are two rats per treatment, so there is 1 d.f. for rats within each treatment. There are 3 treatments so there are 3 x 1 = 3 d.f. for rats within treatments.
- There are 18 liver bits in total but there are not 17 d.f. for liver bits: there are 3 bits in each liver, so there are 2 d.f. for liver bits within each liver. Since there are 6 livers in total (one for each rat) there are 6 x 2 = 12 d.f. for liver bits within rats.
- There are 36 preparations but there are not 35 d.f. for preparations; there are 2 preparations per liver bit, so there is 1 d.f. for preparation within each liver bit. There are 18 liver bits in all (3 from each of 6 rats) and so there are 1 x 18 = 18 d.f. for preparations within liver bits.
- Using the 3 different error variances to carry out the appropriate F tests we learn that there are no significant differences between treatments but there are significant differences between rats, and between parts of the liver within rats (i.e. small scale spatial heterogeneity in the distribution of glycogen within livers).

Now we carry out the analysis in SPlus. The new concept here is that we include an **Error** term in the model formula to show:

- How many error terms are required (answer: as many as there are plot sizes)
- What is the hierarchy of plot sizes (biggest on the left, smallest on the right)
- After the tilde ~ in the model formula we have Treatment as the only fixed effect
- The Error term follows a plus sign + and is enclosed in round brackets
- The plots, ranked by their relative sizes are separated by slash operators like this

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

```
summary(model)
```

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

Error: Treatment:Rat
              Df Sum Sq Mean Sq
Treatment:Rat  3 797.67  265.89

Error: Treatment:Rat:Liver
                    Df Sum Sq Mean Sq
Treatment:Rat:Liver 12  594.0    49.5

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

These are the correct mean squares, as we can see from our long-hand calculations earlier. The F test for the effect of treatment is 778.78/265.89 = 2.93 (n.s.), for differences between rats within treatments it is 265.89/49.5 = 5.37 (*p*<0.05) and for liver bits within rats F = 49.5/21.17 = 2.24 (*p* = 0.05).

In general, we use the slash operator in **model formulas** where the variables are random effects (i.e. where the factor levels are uninformative), and the asterisk operator in model formulas where the variables are fixed effects (i.e. the factor levels **are** informative). Knowing that a rat is number 2 tells us nothing about that rat. Knowing a rat is male tells us a lot about that rat. In **error formulas** we **always use the slash operator** (never the asterisk operator) to indicate the order of ‘plot-sizes’: the largest plots are on the left of the list and the smallest on the right (see below).

The Wrong Analysis

Here is what ***not*** to do.

```
model2<-aov(Glycogen~Treatment*Rat*Liver)
```

The model has been specified as if it were a full factorial with no nesting and no pseudoreplication. Note that the structure of the data allows this mistake to be made. It is a very common problem with data frames that include pseudoreplication. A summary of the model fit looks like this:

```
summary(model2)
```

```
                    Df  Sum Sq Mean Sq F value     Pr(>F)
Treatment            2 1557.56  778.78 36.7927  4.375e-07***
Rat                  1  413.44  413.44 19.5328  0.0003308***
Liver                2  113.56   56.78  2.6824  0.0955848 .
```

```
Treatment:Rat         2  384.22  192.11  9.0761  0.0018803 **
Treatment:Liver       4  328.11   82.03  3.8753  0.0192714 *
Rat:Liver             2   50.89   25.44  1.2021  0.3235761
Treatment:Rat:Liver   4  101.44   25.36  1.1982  0.3455924
Residuals            18  381.00   21.17
```

Signif. codes: 0 `***'  0.001  `**' 0.01 `*'  0.05  `.' 0.1 ` ' 1 summary(nested1)

This says that there was an enormously significant difference between the treatment means, rats were significantly different from one another and there was a significant interaction between treatments and rats. ***Wrong !*** The analysis is flawed because it is based on the assumption that there is only one error variance and that its value is 21.17. This value is actually the measurement error; that is to say the variation between one reading and another from the *same* piece of liver. For testing whether the treatment has had any effect, it is the rats that are the replicates, and there were only 6 of them in the whole experiment.

Here is a way to avoid making the mistake of pseudoreplication

The idea is to get rid of the pseudoreplication by averaging over the liver bits and preparations for each rat. We need to create a new vector, *ym*, of length 6 containing the mean glycogen levels of each rat. You can see how this works as follows

```
tapply(Glycogen,list(Treatment,Rat),mean)
```

```
         1        2
1 132.5000 148.5000
2 149.6667 152.3333
3 134.3333 136.0000
```

We make this into a vector for use in the model like this:

```
ym<-as.vector(tapply(Glycogen,list(Treatment,Rat),mean))
```

We also need a new vector, *tm*, of length 6 to contain a factor for the 3 treatment levels:

```
tm<-factor(as.vector(tapply(as.numeric(Treatment),list(Treatment,Rat),mean)))
```

Now we can do the anova:

```
summary(aov(ym~tm))
```

```
            Df  Sum Sq Mean Sq F value Pr(>F)
tm           2 259.593 129.796   2.929 0.1971
Residuals    3 132.944  44.315
```

This gives us the same (correct) value of the F test as when we did the full nested analysis. The sums of squares are different, of course, because we are using 6 mean values rather than 36 raw values of glycogen to carry out the analysis. We conclude (correctly) that treatment had no significant effect on liver glycogen content.

Analysis of split plot experiments

Split plot experiments are like nested designs in that they involve plots of different sizes and hence have multiple error terms (one error term for each plot size). They are also like nested designs in that they involve pseudoreplication: measurements made on the smaller plots are pseudoreplicates as far as the treatments applied to larger plots are concerned. This is spatial pseudoreplication, and arises because the smaller plots nested within the larger plots are not spatially independent of one another. The only real difference between nested analysis and split plot analysis is that other than blocks, all of the factors in a split plot experiment are typically fixed effects, whereas in most nested analyses most (or all) of the factors are random effects.

The only things to remember about split plot experiments are that

- we need to draw up as many anova tables as there are plot sizes
- the error term in each table is *the interaction between blocks and all factors applied at that plot size or larger*

This experiment involves the yield of cereals in a factorial experiment with 3 treatments, each applied to plots of different sizes. The largest plots (half of each block) were irrigated or not because of the practical difficulties of watering large numbers of small plots. Next, the irrigated plots were split into 3 smaller split-plots and seeds were sown at different densities. Again, because the seeds were machine sown, larger plots were preferred. Finally, each sowing density plot was split into 3 small split-split plots and fertilisers applied by hand (N alone, P alone and N+P together). The yield data look like this:

	Control			Irrigated		
	N	NP	P	N	NP	P
Block A	81	93	92	78	122	98
	90	107	95	80	100	87
	92	92	89	121	119	110
Block B	74	74	81	136	132	133
	83	95	80	102	105	109
	98	106	98	99	123	94
Block C	82	94	78	119	136	122
	85	88	88	60	114	104
	112	91	104	90	113	118
Block D	85	83	89	116	133	136
	86	89	78	73	114	114
	79	87	86	109	126	131

We begin by calculating SST in the usual way.

$$CF = \frac{\left[\sum y\right]^2}{abcd} = \frac{7180^2}{72} = 716005.6$$

$$SST = \sum y^2 - \frac{\left[\sum y\right]^2}{abcd} = 739762 - CF = 23756.44$$

The block sum of squares, SSB, is straightforward. All we need are the 4 block totals

$$SSB = \frac{\sum B^2}{acd} - CF = \frac{1746^2 + 1822^2 + 1798^2 + 1814^2}{18} - CF = 194.444$$

The irrigation main effect is calculated in a similar fashion

$$SSI = \frac{\sum I^2}{bcd} - CF = \frac{3204^2 + 3976^2}{4 \times 3 \times 3} - CF = 8277.556$$

This is where things get different. In a split plot experiment, *the error term is the interaction between blocks and all factors at that plot size or larger*. Because irrigation treatments are applied to the largest plots, the error term is just block:irrigation. We need to calculate an interaction sub-total table (see Practical 5 for an introduction to interaction sums of squares).

```
  control irrigated
A     831       915
B     789      1033
C     822       976
D     762      1052
```

The large plot error sum of squares SSBI is therefore

$$SSBI = \frac{\sum Q^2}{cd} - SSB - SSI - CF$$

$$SSBI = \frac{831^2 + 915^2 + \ldots + 1052^2}{3 \times 3} - SSB - SSI - CF = 1411.778$$

At this point we draw up the anova table for the largest plots. There were only 8 large plots in the whole experiment, so there are just 7 degrees of freedom in total. Block has 3 d.f., irrigation has 1 d.f., so the error variance has only 7-3-1 = 3 d.f.

Source	SS	d.f.	MS	F	p
Block	194.44	3			
Irrigation	8277.556	1	8277.556	17.59	0.025
Error	1411.778	3	470.593		

Block is a random effect so we are not interested in testing hypotheses about differences between block means. We have not written in a row for the totals, because we want the totals and the degrees of freedom to add up correctly across the 3 different anova tables.

Now we move on to consider the sowing density effects. At this split-plot scale we are interested in the main effects of sowing density, and the interaction between sowing density and irrigation. The error term will be the block:irrigation:density interaction.

$$SSD = \frac{2467^2 + 2226^2 + 2487^2}{4 \times 2 \times 3} - CF = 1758.361$$

For the irrigation:density interaction we need the table of sub totals

```
          high  low medium
  control 1006 1064   1134
irrigated 1461 1162   1353
```

$$SSID = \frac{1006^2 + \ldots + 1353^2}{4 \times 3} - SSI - SSD - CF = 2747.028$$

The error term for the split-plots is the block:irrigation:density interaction, and we need the table of sub totals:

```
      high                       low              medium
  control irrigated   control irrigated  control irrigated
A     266       298       292       267      273       350
B     229       401       258       316      302       316
C     254       377       261       278      307       321
D     257       385       253       301      252        36
```

$$SSBID = \frac{226^2 + 298^2 + \ldots + 252^2 + 36^2}{3} - SSBI - SSID - SSB - SSI - SSD - CF = 2787.944$$

At this point we draw up the second anova table for the split-plots:

Source	SS	d.f.	MS	F	p
Density	1758.361	2	879.181	3.784	0.053
Irrigation:Density	2747.028	2	1373.514	5.912	0.016
Error	2787.944	12	232.329		

There are 4 x 2 x 3 = 24 of these plots so there are 23 d.f. in total. We have used up 7 in the first anova table: there are 2 for density, 2 for irrigation:density and hence 23 – 7 – 2 – 2 = 12 d.f. for error. There is a significant interaction between irrigation and density, so we take no notice of the non significant main effect of density.

Finally, we move on to the smallest, split-split-plots. We first calculate the main effect of fertilizer in the familiar way:

$$SSF = \frac{\sum F^2}{abc} - CF = \frac{2230^2 + 2536^2 + 2414^2}{2 \times 4 \times 3} - CF = 1977.444$$

Now the irrigation:fertilizer interaction: the interaction sub totals are

```
             N   NP    P
  control 1047 1099 1058
irrigated 1183 1437 1356
```

$$SSIF = \frac{1047^2 + 1099^2 + \ldots + 1356^2}{4 \times 3} - SSI - SSF - CF = 953.444$$

and the density:fertilizer interaction

```
         N  NP   P
  high 771 867 829
   low 659 812 755
medium 800 857 830
```

$$SSDF = \frac{771^2 + 867^2 + ... + 830^2}{4 \times 2} - SSD - SSF - CF = 304.889$$

The final 3-way interaction is calculated next: irrigation:density:fertilizer

```
              N                    NP                  P
            high  low medium   high low  medium   high low medium
  control   322   344  381      344 379   376    340     341  377
irrigated   449   315  419      523 433   481    489     414  453
```

$$SSIDF = \frac{322^2 + 344^2 + ... + 414^2 + 453^2}{4} - SSID - SSIF - SSDF - SSI - SSD - SSF - CF = 234.722$$

The rest is easy. The error sum of squares is just the remainder when all the calculated sums of squares are subtracted from the total:

$$SSE = SST - SSB - SSI - SSD - SSF - SSIB - SSID - ... - SSIDF = 3108.833$$

Technically, this is the block:irrigation:density:fertilizer interaction. There are 72 plots at this scale and we have used up 7 + 16 = 23 degrees of freedom in the first two anova tables. In the last anova table fertilizer has 2 d.f., the fertilizer:irrigation interaction has 2 d.f., the fertilizer: density interaction has 4 d.f. and the irrigation:density:fertilizer interaction has a further 4 d.f.. This leaves 71- 23 – 2 – 2 – 4 – 4 = 36 d.f. for error. At this point we can draw up the final anova table.

Source	SS	d.f.	MS	F	p
Fertilizer	1977.444	2	988.722	11.449	0.00014
Irrigation:Fertilizer	953.444	2	476.722	5.52	0.0081
Density:Fertilizer	304.889	4	76.222	0.883	n.s.
Irrigtn:Density:Fertilr	234.722	4	58.681	0.68	n.s.
Error	3108.833	36	86.356		

The 3-way interaction was not significant, nor was the 2-way interaction between density and fertilizer. The 2-way interaction between irrigation and fertilizer, however, was highly significant and, not surprisingly, there was a highly significant main effect of fertilizer.

Obviously, you would not want to have to do calculations like this by hand every day. Fortunately, the computer eats analyses like this for breakfast.

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

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

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

The model is long, but not particularly complicated. Note the two parts: the model formula (the factorial design: irrigation*density*fertilizer), and the Error structure (with plot sizes listed left to right from largest to smallest, separated by slash / operators). The main replicates are blocks, and these provide the estimate of the error variance for the largest treatment plots (irrigation). We use asterisks in the model formula because these are fixed effects (i.e. their factor levels *are* informative).

```
summary(model)
```

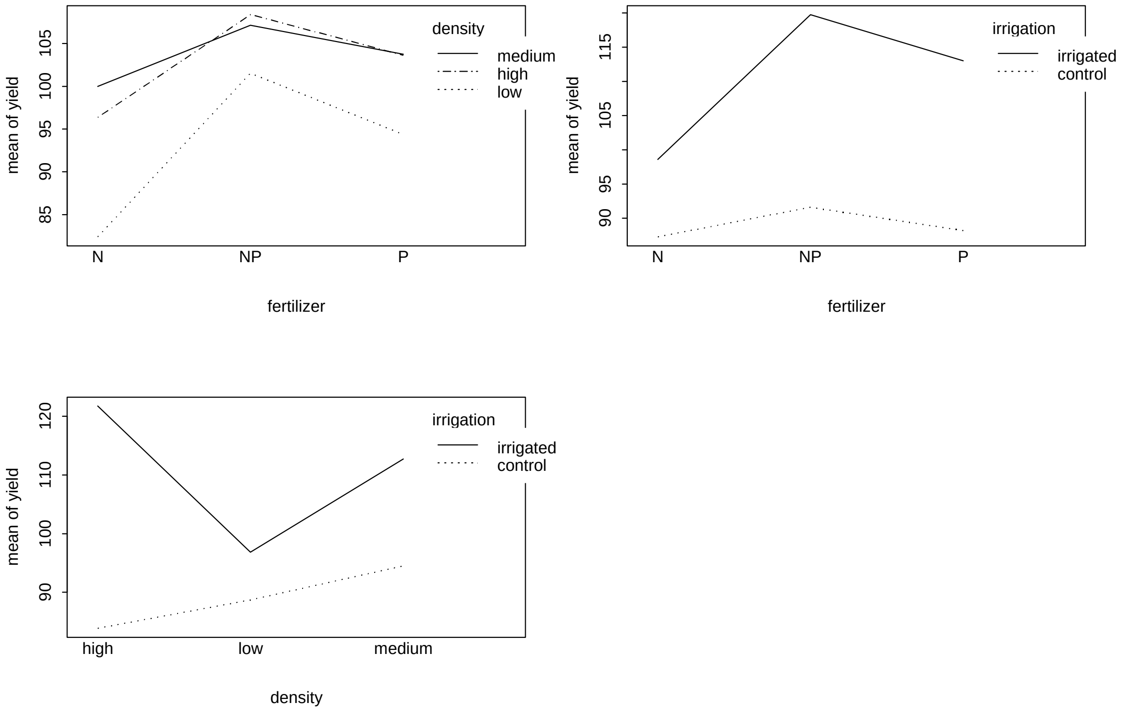
This produces a series of anova tables, one for each plot size, starting with the largest plots (block), then looking at irrigation within blocks, then density within irrigation within block, then finally fertilizer within density within irrigation within block. Notice that the error degrees of freedom are correct in each case (e.g. there are only 3 d.f. for error in assessing the irrigation main effect, but it is nevertheless significant; p = 0.025).

```
Error: block
      Df  Sum Sq Mean Sq
block  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: block:irrigation:density:fertilizer

                              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
```

There are two significant interactions. The best way to understand these is to use the **interaction.plot** directive. The variables are listed in a non-obvious order: first the factor to go on the x axis, then the factor to go as different lines on the plot, then the response variable. There are 3 plots to look at so we make a 2 x 2 plotting area:

```
par(mfrow=c(2,2))
interaction.plot(fertilizer,density,yield)
interaction.plot(fertilizer,irrigation,yield)
interaction.plot(density,irrigation,yield)
```

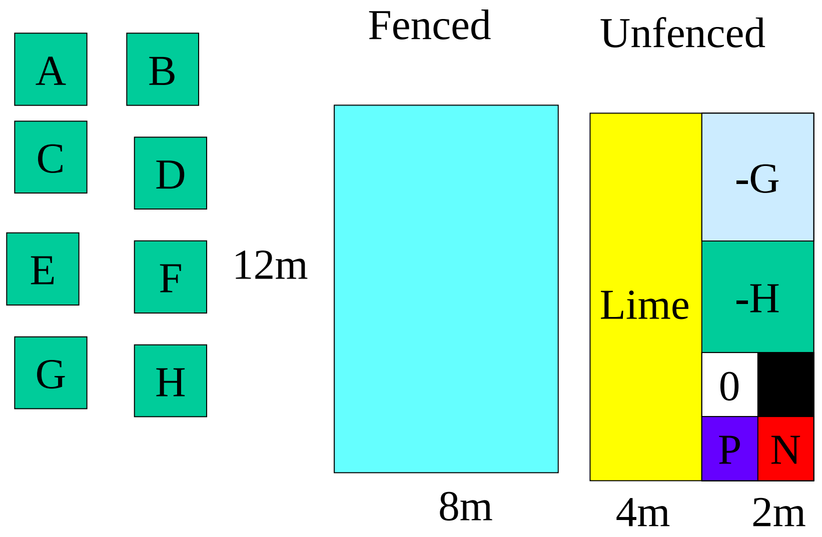
```
par(mfrow=c(1,1))
```



Obviously, the really pronounced interaction is that between irrigation and density, with a reversal of the high to low density difference on the irrigated and control plots. Interestingly, this is not the most significant interaction. That honour goes to the fertilizer:irrigation interaction (top right graph).

A complex split-split-split-split plot field experiment

This example explains the analysis of a field experiment on the impact of grazing and plant competition on the yield of forage biomass (measured as dry matter in tonnes per hectare). To understand what is going on you will need to study the experimental layout quite carefully.



There are 8 large plots (A to H), forming the Blocks of the experiment each measuring 12 x 16 m. Two treatments are applied at this scale: plus and minus Insecticide spray and plus and minus Mollusc pellets. Each of the 4 treatment combinations is replicated twice, and applied at random to one of the 8 Blocks. Each Block is split in half in a randomly selected direction (up and down, or left to right) and Rabbit fencing is allocated at random to half of the Block. Within each split plot the plot is split again, and 1 of 2 liming treatments is allocated at random. Within each split-split plot the area is divided into 3, and 1 of 3 plant competition treatments is allocated at random (control ,minus grass using a grass-specific herbicide, or minus herb using a herb-specific herbicide). Finally, within each split-split-split plot the area is divided into 4 and 1 of 4 nutrient treatments is applied at random (plus and minus nitrogen and plus and minus phosphorus). The whole design, therefore, contains

$$8 \times 2 \times 2 \times 3 \times 4 = 384$$

of the smallest (2m x 2m) plots. The data frame looks like this:

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

```
[1] "Block"    "Insect"    "Mollusc"   "Rabbit"   "Lime"    "Competition"
[7] "Nutrient"    "Biomass"
```

The **response variable** is Biomass, Block is a **random effect** and the other variables are all **fixed effects** applied as treatments (at random of course, which is a bit confusing). As before, analysing the data requires us to specify 2 things: the treatment structure and the error structure. The treatment structure is simply a full factorial

```
Insect*Mollusc*Rabbit*Lime*Competition*Nutrient
```

Specified using the * operator to indicate that all main effects and all interactions are to be fitted. The error term shows how the different plot sizes are related to the explanatory variables. We list, for left to right, the names of the variables relating to progressively smaller plots, with each name separated by the / (slash) operator:

```
Block/Rabbit/Lime/Competition/Nutrient
```

There are 5 variable names in the Error directive because there are 5 different sizes of plots (2 x 2m for nutrients, 4 x 4m for competition, 12 x 4m for lime, 12 x 8m for rabbit grazing and 12 x 16m for insecticide or molluscicide). Where treatments are applied to plots of the same size (e.g. insecticide and molluscicide in this example) we need only specify *one* of the names (it does not matter which: we used Insect but we could equally well have used Mollusc). The analysis is run by combining the treatment and error structure in a single **aov** directive (it may not run on your machine because of memory limitations: but here is the SPlus output anyway)

```
model<-aov(Biomass~Insect*Mollusc*Rabbit*Lime*Competition*Nutrient
                    +Error(Block/Rabbit/Lime/Competition/Nutrient))
```

```
summary(model)
```

```
Error: Block
               Df Sum of Sq  Mean Sq  F Value     Pr(F)
        Insect  1  414.6085 414.6085 34.27117 0.0042482
       Mollusc  1    8.7458   8.7458  0.72292 0.4430877
Insect:Mollusc  1   11.0567  11.0567  0.91394 0.3932091
     Residuals  4   48.3915  12.0979

Error: Rabbit %in% Block
                      Df Sum of Sq  Mean Sq  F Value     Pr(F)
               Rabbit  1  388.7935 388.7935 4563.592 0.0000003
        Insect:Rabbit  1    0.4003   0.4003    4.698 0.0960688
       Mollusc:Rabbit  1    0.0136   0.0136    0.160 0.7096319
Insect:Mollusc:Rabbit  1    0.2477   0.2477    2.908 0.1633515
            Residuals  4    0.3408   0.0852

Error: Lime %in% (Block/Rabbit)
                           Df Sum of Sq  Mean Sq  F Value     Pr(F)
                      Lime  1  86.63703 86.63703 1918.264 0.0000000
               Insect:Lime  1   0.03413  0.03413    0.756 0.4100144
              Mollusc:Lime  1   0.12197  0.12197    2.701 0.1389385
               Rabbit:Lime  1   0.14581  0.14581    3.228 0.1100955
       Insect:Mollusc:Lime  1   0.05160  0.05160    1.143 0.3163116
        Insect:Rabbit:Lime  1   0.00359  0.00359    0.079 0.7852903
       Mollusc:Rabbit:Lime  1   0.09052  0.09052    2.004 0.1945819
Insect:Mollusc:Rabbit:Lime  1   0.46679  0.46679   10.335 0.0123340
                 Residuals  8   0.36131  0.04516
```

```
Error: Competition %in% (Block/Rabbit/Lime)

                                       Df Sum of Sq  Mean Sq  F Value     Pr(F)
                           Competition  2  214.4145 107.2073 1188.317 0.0000000
                    Insect:Competition  2    0.1502   0.0751    0.832 0.4442496
                   Mollusc:Competition  2    0.1563   0.0782    0.866 0.4300752
                    Rabbit:Competition  2    0.1981   0.0991    1.098 0.3457568
                      Lime:Competition  2    0.0226   0.0113    0.125 0.8825194
            Insect:Mollusc:Competition  2    0.4132   0.2066    2.290 0.1176343
             Insect:Rabbit:Competition  2    0.1221   0.0611    0.677 0.5153674
            Mollusc:Rabbit:Competition  2    0.0221   0.0111    0.123 0.8850922
               Insect:Lime:Competition  2    0.0527   0.0263    0.292 0.7487901
              Mollusc:Lime:Competition  2    0.0296   0.0148    0.164 0.8493921
               Rabbit:Lime:Competition  2    0.0134   0.0067    0.074 0.9286778
     Insect:Mollusc:Rabbit:Competition  2    0.0307   0.0154    0.170 0.8442710
       Insect:Mollusc:Lime:Competition  2    0.0621   0.0311    0.344 0.7112350
        Insect:Rabbit:Lime:Competition  2    0.3755   0.1878    2.081 0.1413456
       Mollusc:Rabbit:Lime:Competition  2    0.5007   0.2504    2.775 0.0773730
Insect:Mollusc:Rabbit:Lime:Competition  2    0.0115   0.0057    0.064 0.9385470
                             Residuals 32    2.8870   0.0902

Error: Nutrient %in% (Block/Rabbit/Lime/Competition)
                                                 Df Sum of Sq  Mean Sq  F Value     Pr(F)
                                       Nutrient   3  1426.017 475.3389 6992.589 0.0000000
                                Insect:Nutrient   3     0.213   0.0711    1.046 0.3743566
                               Mollusc:Nutrient   3     0.120   0.0400    0.589 0.6231103
                                Rabbit:Nutrient   3     0.087   0.0291    0.428 0.7332115
                                  Lime:Nutrient   3     0.035   0.0116    0.171 0.9156675
                           Competition:Nutrient   6     0.724   0.1207    1.775 0.1081838
                        Insect:Mollusc:Nutrient   3     0.112   0.0373    0.549 0.6497462
                         Insect:Rabbit:Nutrient   3     0.783   0.2611    3.840 0.0110859
                        Mollusc:Rabbit:Nutrient   3     0.929   0.3096    4.554 0.0044319
                           Insect:Lime:Nutrient   3     0.059   0.0196    0.289 0.8335956
                          Mollusc:Lime:Nutrient   3     0.476   0.1585    2.332 0.0766893
                           Rabbit:Lime:Nutrient   3     0.376   0.1252    1.842 0.1422169
                    Insect:Competition:Nutrient   6     0.556   0.0927    1.363 0.2333048
                   Mollusc:Competition:Nutrient   6     0.347   0.0578    0.851 0.5327535
                    Rabbit:Competition:Nutrient   6     0.431   0.0718    1.057 0.3914693
                      Lime:Competition:Nutrient   6     0.451   0.0752    1.106 0.3616015
                 Insect:Mollusc:Rabbit:Nutrient   3     0.411   0.1370    2.016 0.1143309
                   Insect:Mollusc:Lime:Nutrient   3     0.697   0.2324    3.418 0.0190776
                    Insect:Rabbit:Lime:Nutrient   3     0.435   0.1450    2.132 0.0987222
                   Mollusc:Rabbit:Lime:Nutrient   3     0.055   0.0184    0.270 0.8468838
            Insect:Mollusc:Competition:Nutrient   6     0.561   0.0936    1.376 0.2279809
             Insect:Rabbit:Competition:Nutrient   6     0.728   0.1213    1.784 0.1062479
            Mollusc:Rabbit:Competition:Nutrient   6     0.427   0.0712    1.047 0.3974232
               Insect:Lime:Competition:Nutrient   6     0.258   0.0430    0.632 0.7043702
              Mollusc:Lime:Competition:Nutrient   6     0.417   0.0695    1.023 0.4128486
               Rabbit:Lime:Competition:Nutrient   6     0.406   0.0676    0.994 0.4315034
            Insect:Mollusc:Rabbit:Lime:Nutrient   3     0.350   0.1166    1.715 0.1665522
     Insect:Mollusc:Rabbit:Competition:Nutrient   6     0.259   0.0431    0.635 0.7023702
       Insect:Mollusc:Lime:Competition:Nutrient   6     0.403   0.0672    0.988 0.4355882
        Insect:Rabbit:Lime:Competition:Nutrient   6     0.282   0.0470    0.692 0.6565638
       Mollusc:Rabbit:Lime:Competition:Nutrient   6     0.355   0.0592    0.870 0.5183944
Insect:Mollusc:Rabbit:Lime:Competition:Nutrient   6     0.989   0.1648    2.424 0.0291380
                                      Residuals 144     9.789   0.0680
```

Notice that you get 5 separate ANOVA tables, one for each different plot size. It is the number of plot sizes, not the number of treatments that determines the shape of the split-plot ANOVA table. The number of ANOVA tables would not have changed if we had specified the Nutrient treatment (4 levels) as a 2 by 2 factorial with Nitrogen and Phosphorus each as 2-level treatments. Because Insecticide and Molluscicide were both applied at the same plot size (whole Blocks) they appear in the same ANOVA table.

Interpretation of output tables like this requires a high level of serenity. The first thing to do is to check that the degrees of freedom have been handled properly. There were 8 bocks, with 2 replicates of the large-plot factorial experiment of plus and minus insects and plus and minus molluscs. This means that there are 7 d.f. in total, and so with 1 d.f. for Insect, 1 d.f. for Mollusc and 1 d.f. for the Insect:Mollusc interaction, there should be

7-1-1-1 = 4 d.f. for error. This checks out in the top ANOVA table where Error is labelled as Block. For the largest plots, therefore, the error variance = 12.1. We can now assess the significance of these treatments that were applied to the largest plots. As ever, we begin with the interaction. This is clearly not significant, so we can move on to interpreting the main effects. There is no effect of mollusc exclusion on biomass, but insect exclusion led to a significant increase in mean biomass ($p = 0.0042$).

The 2nd largest plots were those with or without fences to protect them from rabbit grazing. To check the degrees of freedom, we need to work out the total number of rabbit-grazed and fenced plots. There were 8 blocks, each split in half, so there are 16 plots. We have already used 7 d.f. for the insect by mollusc experiment (above) so there are 16 – 7 – 1 = 8 d.f. remaining. Rabbit grazing has 1 d.f. and there are 3 interaction terms, each with 1 d.f. (Rabbit:Insect, Rabbit:Mollusc and Rabbit:Insect:Mollusc). This means that these terms should be assessed by an error variance that has 8 – 1- 3 = 4 d.f. This also checks out. The error variance is 0.085, and shows that there are no significant interactions between rabbit grazing and invertebrate herbivores, but there is a highly significant main effect of rabbit grazing.

The 3rd largest plots were either limed or not limed. In this case there are 8 d.f. for error, and we discover the first significant interaction: Insect:Mollusc:Rabbit:Lime ($p = 0.012$). Like all high-order interactions, this is extremely complicated to interpret. It means that the 3-way interaction between Insect:Mollusc:Rabbit works differently on limed and unlimed plots. It would be unwise to over-interpret this result without further experimentation focussed on the way that this interaction might work. There is a highly significant main effect of lime.

The 4th largest plots received one of 3 plant competition treatments: control, minus grass or minus herb. There are 96 competition plots, and we have used 31 d.f. so far on the larger plots, so there should be 96 – 31 – 1 = 64 d.f. at this level. With 32 d.f. for main effects and interactions, that leaves 32 d.f. for error. This checks out, and the error variance is 0.09. There are no significant interaction terms, but competition had a significant main effect on biomass.

The 5th largest plots (the smallest at 2m x 2m) received one of 4 nutrient treatments: plus or minus nitrogen and plus or minus phosphorus. All the remaining degrees of freedom can be used at this scale, leaving 144 d.f. for error, and an error variance of 0.068. There are several significant interactions: the 6-way Insect:Mollusc:Rabbit:Lime: Competition:Nutrient ($p = 0.029$), a 4-way, Insect:Mollusc:Lime:Nutrient ($p = 0.019$), and two 3-way interactions, Insect:Rabbit:Nutrient ($p = 0.011$) and Mollusc:Rabbit:Nutrient ($p = 0.004$). At this point I can confide in you: I made up these results, so I know that all of these small-plot interactions are due to chance alone.

This raises an important general point. In big, complicated experiments like this it is sensible to use a very high level of alpha in assessing the significance of high order interactions. This compensates for the fact that you are doing a vast number of hypothesis tests, and has the added bonus of making the results much more straightforward to write up. It is a trade-off, of course, because you do not want to be so harsh that you throw out the baby with the bathwater, and miss biologically important and potentially very interesting interactions.

We can inspect the interactions in 2 ways. Tables of interaction means can be produced using **tapply**:

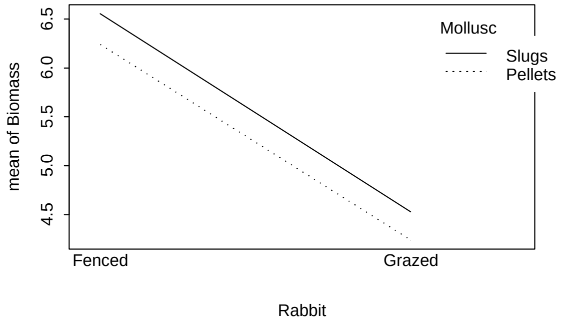
```
tapply(Biomass,list(Mollusc,Rabbit,Nutrient),mean)
```

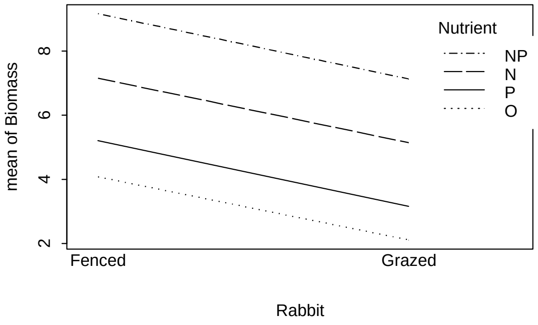
```
, , N
          Fenced   Grazed
Pellets 6.963431 4.984890
  Slugs 7.322273 5.298524

, , NP
          Fenced   Grazed
Pellets 9.056132 6.923177
  Slugs 9.257630 7.324827

, , O
          Fenced   Grazed
Pellets 3.873364 2.069350
  Slugs 4.282690 2.149054

, , P
          Fenced   Grazed
Pellets 5.066604 2.979994
  Slugs 5.351919 3.344672
```

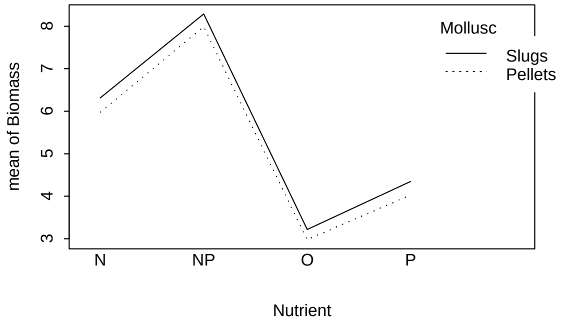
Better still, use **interaction.plot** to inspect the interaction terms, 2 at a time. Take the Mollusc:Rabbit:Nutrient interaction (p = 0.004) whose means we have just calculated. We expect that the interaction plot will show non-parallelness of some form or other. We divide the plotting space into 4, then make 3 separate interaction plots:

```
par(mfrow=c(2,2))
interaction.plot(Rabbit,Mollusc,Biomass)
interaction.plot(Rabbit,Nutrient,Biomass)
interaction.plot(Nutrient,Mollusc,Biomass)
```

It is quite clear from the plots that the interaction, though statistically significant, is not biologically substantial. The plots are virtually parallel in all 3 panels. The graphs demonstrate clearly the interaction between nitrogen and phosphorus (bottom left) but this is not materially altered by either mollusc or rabbit grazing.







Doing the wrong “Factorial” analysis of multi-split-plot data

Here’s how **not** to do it. With data like this there are always lots of different ways of getting the model and/or the error structure wrong. The commonest mistake is to treat the data as a full factorial, like this:

```
model<-aov(Biomass~Insect*Mollusc*Rabbit*Lime*Competition*Nutrient)
```

It looks perfectly reasonable, and indeed, this is the experimental design intended. But it is rife with pseudoreplication. Ask yourself how many independent plots had insecticide applied to them or not. The answer is 8. The analysis carried out here assumes there were 384 ! And what about plant competition treatments ? The answer is 96, but this analysis assumes 384. And so on. Let’s see the consequences of the pseudoreplication: It thinks the error variance is 0.3217 for testing all the interactions and main effects. The only significant interaction is Insecticide by Molluscicide and this appears to have an F ratio of 34.4 on d.f. = 1,192. All 6 main effects appear to be significant at p<0.000001.

summary(model)

```
                                                 Df Sum of Sq  Mean Sq  F Value     Pr(F)
                                         Insect   1   414.609 414.6085 1288.743 0.0000000
                                        Mollusc   1     8.746   8.7458   27.185 0.0000005
                                         Rabbit   1   388.793 388.7935 1208.502 0.0000000
                                           Lime   1    86.637  86.6370  269.297 0.0000000
                                    Competition   2   214.415 107.2073  333.236 0.0000000
                                       Nutrient   3  1426.017 475.3389 1477.514 0.0000000
                                 Insect:Mollusc   1    11.057  11.0567   34.368 0.0000000
                                  Insect:Rabbit   1     0.400   0.4003    1.244 0.2660546
                                 Mollusc:Rabbit   1     0.014   0.0136    0.042 0.8371543
                                    Insect:Lime   1     0.034   0.0341    0.106 0.7450036
                                   Mollusc:Lime   1     0.122   0.1220    0.379 0.5388059
                                    Rabbit:Lime   1     0.146   0.1458    0.453 0.5016203
                             Insect:Competition   2     0.150   0.0751    0.233 0.7920629
                            Mollusc:Competition   2     0.156   0.0782    0.243 0.7845329
                             Rabbit:Competition   2     0.198   0.0991    0.308 0.7353323
                               Lime:Competition   2     0.023   0.0113    0.035 0.9654345
                                Insect:Nutrient   3     0.213   0.0711    0.221 0.8817673
                               Mollusc:Nutrient   3     0.120   0.0400    0.124 0.9455545
                                Rabbit:Nutrient   3     0.087   0.0291    0.090 0.9652343
                                  Lime:Nutrient   3     0.035   0.0116    0.036 0.9907599
                           Competition:Nutrient   6     0.724   0.1207    0.375 0.8942470
                          Insect:Mollusc:Rabbit   1     0.248   0.2477    0.770 0.3813061
                            Insect:Mollusc:Lime   1     0.052   0.0516    0.160 0.6892415
                             Insect:Rabbit:Lime   1     0.004   0.0036    0.011 0.9160383
                            Mollusc:Rabbit:Lime   1     0.091   0.0905    0.281 0.5964109
                     Insect:Mollusc:Competition   2     0.413   0.2066    0.642 0.5272844
                      Insect:Rabbit:Competition   2     0.122   0.0611    0.190 0.8272856
                     Mollusc:Rabbit:Competition   2     0.022   0.0111    0.034 0.9662288
                        Insect:Lime:Competition   2     0.053   0.0263    0.082 0.9214289
                       Mollusc:Lime:Competition   2     0.030   0.0148    0.046 0.9550433
                        Rabbit:Lime:Competition   2     0.013   0.0067    0.021 0.9794192
                        Insect:Mollusc:Nutrient   3     0.112   0.0373    0.116 0.9506702
                         Insect:Rabbit:Nutrient   3     0.783   0.2611    0.811 0.4889277
                        Mollusc:Rabbit:Nutrient   3     0.929   0.3096    0.962 0.4116812
                           Insect:Lime:Nutrient   3     0.059   0.0196    0.061 0.9802371
                          Mollusc:Lime:Nutrient   3     0.476   0.1585    0.493 0.6877598
                           Rabbit:Lime:Nutrient   3     0.376   0.1252    0.389 0.7609242
                    Insect:Competition:Nutrient   6     0.556   0.0927    0.288 0.9421006
                   Mollusc:Competition:Nutrient   6     0.347   0.0578    0.180 0.9820982
                    Rabbit:Competition:Nutrient   6     0.431   0.0718    0.223 0.9688921
                      Lime:Competition:Nutrient   6     0.451   0.0752    0.234 0.9651323
                     Insect:Mollusc:Rabbit:Lime   1     0.467   0.4668    1.451 0.2298613
              Insect:Mollusc:Rabbit:Competition   2     0.031   0.0154    0.048 0.9534089
                Insect:Mollusc:Lime:Competition   2     0.062   0.0311    0.097 0.9079804
                 Insect:Rabbit:Lime:Competition   2     0.376   0.1878    0.584 0.5588583
                Mollusc:Rabbit:Lime:Competition   2     0.501   0.2504    0.778 0.4606637
                 Insect:Mollusc:Rabbit:Nutrient   3     0.411   0.1370    0.426 0.7346104
                   Insect:Mollusc:Lime:Nutrient   3     0.697   0.2324    0.722 0.5397982
                    Insect:Rabbit:Lime:Nutrient   3     0.435   0.1450    0.451 0.7171854
                   Mollusc:Rabbit:Lime:Nutrient   3     0.055   0.0184    0.057 0.9820420
            Insect:Mollusc:Competition:Nutrient   6     0.561   0.0936    0.291 0.9407903
             Insect:Rabbit:Competition:Nutrient   6     0.728   0.1213    0.377 0.8930313
            Mollusc:Rabbit:Competition:Nutrient   6     0.427   0.0712    0.221 0.9695915
               Insect:Lime:Competition:Nutrient   6     0.258   0.0430    0.134 0.9918574
              Mollusc:Lime:Competition:Nutrient   6     0.417   0.0695    0.216 0.9713330
               Rabbit:Lime:Competition:Nutrient   6     0.406   0.0676    0.210 0.9733124
         Insect:Mollusc:Rabbit:Lime:Competition   2     0.011   0.0057    0.018 0.9823388
            Insect:Mollusc:Rabbit:Lime:Nutrient   3     0.350   0.1166    0.362 0.7802436
     Insect:Mollusc:Rabbit:Competition:Nutrient   6     0.259   0.0431    0.134 0.9917702
       Insect:Mollusc:Lime:Competition:Nutrient   6     0.403   0.0672    0.209 0.9737285
        Insect:Rabbit:Lime:Competition:Nutrient   6     0.282   0.0470    0.146 0.9896249
       Mollusc:Rabbit:Lime:Competition:Nutrient   6     0.355   0.0592    0.184 0.9810233
Insect:Mollusc:Rabbit:Lime:Competition:Nutrient   6     0.989   0.1648    0.512 0.7987098
                                      Residuals 192    61.769   0.3217
```

The problems are of 2 kinds. For the largest plots, the error variance is underestimated because of the pseudoreplication, so things appear significant which actually are not. For example, the correct analysis shows that the Insect by Mollusc interaction is not significant (p = 0.39) but the wrong analysis suggests that it is highly significant (p = 0.00000). The other problem is of the opposite kind. Because this analysis does not factor out the large between plot variation at the larger scale, the error variance for testing small plot effects is much too big. In the correct, split plot analysis, the small-plot error variance is very small (0.068) compared with the pseudoreplicated, small-plot error variance of 0.32 (above).

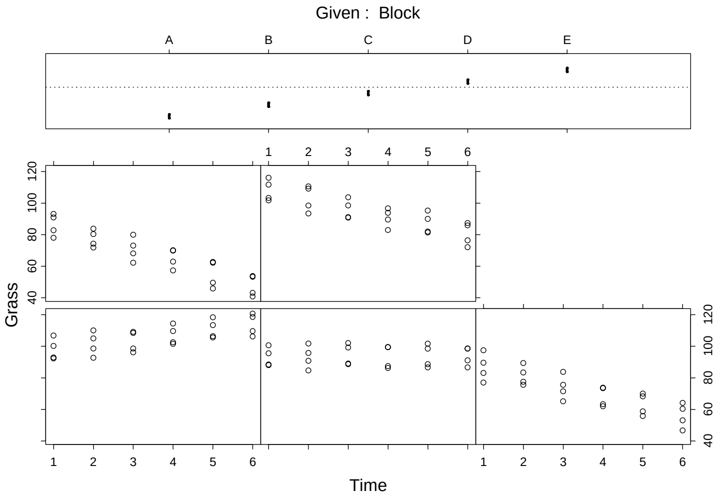
Mixed Effects Models (you can skip this section if you have had enough already)

This example is like the last one except now we have introduced temporal pseudoreplication (repeated measurements) as well as spatial pseudoreplication (different plot sizes) in a split plot experiment. The name “mixed effects” means that we have a mixture of fixed effects (experimental treatments) and random effects (blocks and time in this case). There are 5 replicates and 2 experimental treatments (insect exclusion and mollusc exclusion). Data were gathered from 6 successive harvests on each plot.

```
repeated<-read.table("c:\\temp\\repeated.txt",header=T)
attach(repeated)
names(repeated)
[1] "Block"   "Time"    "Insect"  "Mollusc" "Grass"
```

This is what the data look like (**coplot** draws Grass as a function of Time, *given* Block)

```
coplot(Grass ~ Time | Block)
```



Here is the most obvious way of doing the **wrong** analysis. Just fit Block and the 2 x 2 factorial (Insect*Mollusc) to the whole data set:

```
model<-aov(Grass~Insect*Mollusc+Error(Block))
```

```
summary(model)
```

```
Error: Block

          Df  Sum Sq Mean Sq F value Pr(>F)
Residuals  4 25905.9  6476.5

Error: Within
                Df Sum Sq Mean Sq F value     Pr(>F)
Insect           1 3826.1  3826.1 43.9838 1.218e-009 ***
Mollusc          1  314.6   314.6  3.6169    0.05976 .
Insect:Mollusc   1    0.1     0.1  0.0010    0.97429
Residuals      112 9742.6    87.0
```

The conclusion is clear: there is no interaction, but insect exclusion has a highly significant main effect on grass yield and mollusc exclusion has a close-to-significant effect. However, you can see at once that the analysis is wrong, because there are far too many degrees of freedom for error (d.f. = 112). There are only 5 replicates of the experiment, so this is clearly not right. Both effects are in the expected direction (herbivore exclusion increases grass yield).

```
tapply(Grass,list(Insect,Mollusc), mean)
```

```
               Absent    Present
  Sprayed      93.71961  90.42613
Unsprayed      82.37145  79.18800
```

A sensible way to see what is going on is to do a regression of Grass against Time separately for each treatment in each block, to check whether there are temporal trends, and if so, whether the temporal trends are the same in each block. This is an analysis of covariance (see Practical 6) with different slopes (Time) estimated for every combination of Block:Insect:Mollusc

```
model<-aov(Grass~Block*Insect*Mollusc*Time)
```

```
summary(model)
```

```
                          Df Sum of Sq  Mean Sq  F Value     Pr(F)
                    Block  4  25905.93 6476.483 2288.045 0.0000000
                   Insect  1   3826.05 3826.051 1351.687 0.0000000
                  Mollusc  1    314.63  314.629  111.154 0.0000000
                     Time  1   3555.40 3555.400 1256.070 0.0000000
             Block:Insect  4      1.59    0.397    0.140 0.9667569
            Block:Mollusc  4     10.74    2.685    0.949 0.4404798
           Insect:Mollusc  1      0.09    0.091    0.032 0.8583185
               Block:Time  4   5898.01 1474.502  520.920 0.0000000
              Insect:Time  1      0.39    0.389    0.137 0.7118010
             Mollusc:Time  1     13.90   13.900    4.911 0.0295350
     Block:Insect:Mollusc  4     11.21    2.802    0.990 0.4179119
        Block:Insect:Time  4      3.65    0.914    0.323 0.8619207
       Block:Mollusc:Time  4      2.33    0.583    0.206 0.9343920
      Insect:Mollusc:Time  1     16.38   16.381    5.787 0.0184497
Block:Insect:Mollusc:Time  4      2.58    0.644    0.227 0.9222823
                Residuals 80    226.45    2.831
```

This shows some interesting features of the data. Starting with the highest order interaction and working upwards, we see that there is a significant interaction between Insect, Mollusc and Time. That is to say, the slope of the graph of grass against time is significantly different for different combinations of insecticide and molluscicide. The effect is not massively significant (p = 0.018) but it *is* there (another component of this effect appears as a Mollusc by Time interaction). There is an extremely significant interaction between Block and Time, with different slopes in different Blocks (as we saw in the **coplot**, earlier). At this stage, we need to consider whether the suggestion of an Insect by Mollusc by Time interaction is worth following up. It is plausible that insects had more impact on grass yield at one time of year and molluscs at another. The problem is that this analysis does not take account of the fact that the measurements through time are correlated because they were taken from the same location (a plot within a block).

One simple way around this is to carry out separate analyses of variance for each time period. We use the **subset** directive to restrict the analysis, and put the whole thing in a loop for time i = 1 to 6:

```
for (i in 1:6 ) print(summary(aov(Grass~Insect*Mollusc+Error(Block),subset=(Time==i))))
```

```
Error: Block
          Df  Sum Sq Mean Sq F value Pr(>F)
Residuals  4 1321.30  330.32

Error: Within
               Df Sum Sq Mean Sq  F value     Pr(>F)
Insect          1 652.50  652.50 257.4725 1.794e-009 ***
Mollusc         1  77.76   77.76  30.6823  0.0001280 ***
Insect:Mollusc  1   8.47    8.47   3.3435  0.0924206 .
Residuals      12  30.41    2.53
```

```
Error: Block
          Df  Sum Sq Mean Sq F value Pr(>F)
Residuals  4 2119.29  529.82

Error: Within
               Df Sum Sq Mean Sq  F value     Pr(>F)
Insect          1 620.38  620.38 253.1254 1.978e-009 ***
Mollusc         1  94.88   94.88  38.7122 4.438e-005 ***
Insect:Mollusc  1   0.01    0.01   0.0034     0.9545
Residuals      12  29.41    2.45

Error: Block
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals  4 3308.8   827.2

Error: Within
               Df Sum Sq Mean Sq  F value     Pr(>F)
Insect          1 628.26  628.26 183.8736 1.225e-008 ***
Mollusc         1  71.41   71.41  20.9002  0.0006416 ***
Insect:Mollusc  1   4.88    4.88   1.4273  0.2552897
Residuals      12  41.00    3.42

Error: Block
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals  4 5063.6  1265.9

Error: Within
               Df Sum Sq Mean Sq  F value     Pr(>F)
Insect          1 550.43  550.43 194.9970 8.784e-009 ***
Mollusc         1  26.50   26.50   9.3878   0.009827 **
Insect:Mollusc  1   3.63    3.63   1.2876   0.278649
Residuals      12  33.87    2.82

Error: Block
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals  4 8327.2  2081.8

Error: Within
               Df Sum Sq Mean Sq  F value     Pr(>F)
Insect          1 713.18  713.18 301.8366 7.168e-010 ***
Mollusc         1  30.45   30.45  12.8863   0.003715 **
Insect:Mollusc  1   1.09    1.09   0.4625   0.509353
Residuals      12  28.35    2.36
Error: Block
          Df  Sum Sq Mean Sq F value Pr(>F)
Residuals  4 11710.4  2927.6

Error: Within
               Df Sum Sq Mean Sq  F value     Pr(>F)
Insect          1 667.19  667.19 459.7610 6.178e-011 ***
Mollusc         1  33.34   33.34  22.9714  0.0004388 ***
Insect:Mollusc  1  13.04   13.04   8.9868  0.0111120 *
Residuals      12  17.41    1.45
```

Each one of these tables has the correct degrees of freedom for error (d.f. = 12) and they all tell a reasonably consistent story. All show significant main effects for Insect and Mollusc, Insect is generally more significant than Mollusc, and there was a significant interaction only at Time 6 (this is the Insect:Mollusc:Time interaction we discovered earlier).

The simplest way to get rid of the temporal pseudoreplication is just to average it away. That would produce a single number per treatment per block, or 20 numbers in all. We need to produce new, shorted vectors (length 20 instead of length 120) for each of the variables. The shorter vectors will all need new names: let's define shorter Block as b, Insect as i and Mollusc as m, like this

```
b<-Block[Time==1]
i<-Insect[Time==1]
m<-Mollusc[Time==1]
```

What values should go in the shorter vector g for Grass ? They should be the grass values for block, insect and mollusc, averaged over the 6 values of Time. We need to be very careful here, because the averages need to be in exactly the same order as the subscripts in the new explanatory variables we have created. Let's look at their values:

```
b
```

```
 [1] A A A A B B B B C C C C D D D D E E E E
```

```
i
```

```
[1] Sprayed   Sprayed   Unsprayed Unsprayed Sprayed   Sprayed   Unsprayed Unsprayed
Sprayed   Sprayed
[11] Unsprayed Unsprayed Sprayed   Sprayed   Unsprayed Unsprayed Sprayed   Sprayed
Unsprayed Unsprayed
```

```
m
```

```
[1] Present Absent  Present Absent  Present Absent  Present Absent  Present Absent
Present Absent  Present
[14] Absent  Present Absent  Present Absent  Present Absent
```

The first 4 numbers in our array of means need to come from Block A, the first 2 from insecticide Sprayed and the mollusc numbers to alternate Present, Absent, Present, Absent, etc. This means that the list in the **tapply** directive must be in the order Mollusc, Insect, Block. The most rapidly changing subscripts are first in the list, the most slowly changing last. So the shorted vector of Grass weights, *g*, is computed like this:

```
g<-as.vector(tapply(Grass,list(Mollusc,Insect,Block),mean))
```

The **as.vector** part of the expression is necessary to make *g* into a column vector of length 20 rather than a 3-D matrix 2 x 2 x 5.

Now we can carry out the ANOVA fitting block and the 2 x 2 factorial:

```
model<-aov(g~i*m+Error(b))
summary(model)
```

```
Error: b
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals  4 4317.7  1079.4

Error: Within
          Df Sum Sq Mean Sq   F value     Pr(>F)
i          1 637.68  637.68 1950.6284 1.177e-014 ***
m          1  52.44   52.44  160.4069 2.645e-008 ***
i:m        1   0.02    0.02    0.0463     0.8333
Residuals 12   3.92    0.33
```

Now we have the right number of degrees of freedom for error (d.f. = 12). The analysis shows highly significant main effects for both treatments, but no interaction (recall that it was only significant at Time = 6). Note that the error variance s^2 = 0.327 is larger than s^2 was in 5 of the 6 separate analyses, because of the large Block:Time interaction.

An alternative is to detrend the data by regressing Grass against Time for each treatment combination (Code = 1 to 20), then use the slopes and the intercepts of these regressions in 2 separate ANOVA's in what is called a *derived variables analysis*.

Here is one way to extract the 20 regression intercepts, *a*, which are parameter [1] within the **coef** of the fitted model:

```
a<-1:20
Code<-gl(20,6)
for (k in 1:20) a[k]<-coef(lm(Grass~Time,subset=(Code==k)))[1]

a
```

```
[1]   97.27997 103.22584  87.71079  90.27373  95.41965
[6]  101.91840  87.22158  88.60817  94.41061 103.25222
[11]   85.07103  89.28192  96.10313 100.96869  86.18559
[16]   91.19490 116.83651 121.69893 106.40919 108.55959
```

This shortened vector now becomes the response variable in a simple analysis of variance, using the shortened factors we calculated earlier:

```
model<-aov(a~i*m+Error(b))
summary(model)
```

```
Error: b
          Df  Sum Sq Mean Sq F value Pr(>F)
Residuals  4 1253.30  313.33

Error: Within
          Df Sum Sq Mean Sq F value     Pr(>F)
i          1 611.59  611.59 594.933 1.359e-011 ***
m          1 107.34  107.34 104.420 2.834e-007 ***
i:m        1  12.32   12.32  11.980   0.004707 **
Residuals 12  12.34    1.03
```

Everything in the model has a significant effect on the intercept, including a highly significant interaction between Insects and Molluscs (p = 0.0047). What about the slopes of the Grass against Time graphs; **coef** [2] of the fitted object ?

```
aa<-1:20
for (k in 1:20) aa[k]<-coef(lm(Grass~Time,subset=(Code==k)))[2]

aa
 [1]  3.3885561  2.8508475  3.2423677  3.1764151
 [5]  0.6762409 -0.3742587 -0.1142285  0.1536275
 [9] -5.5199901 -6.7429909 -6.1245833 -6.1412692
[13] -6.9687946 -7.7624054 -7.6693108 -7.9243495
[17] -5.2672184 -5.8186351 -5.5222919 -5.2818614
```

As we saw from the initial graphs, there is great variation in slope from Block to Block, but is there any effect of treatments ?

```
model<-aov(aa~i*m+Error(b))
summary(model)


Error: b
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals  4 337.03   84.26

Error: Within
          Df  Sum Sq Mean Sq F value    Pr(>F)
i          1 0.02223 0.02223  0.5453 0.4744357
m          1 0.79426 0.79426 19.4805 0.0008449 ***
i:m        1 0.93608 0.93608 22.9588 0.0004398 ***
Residuals 12 0.48927 0.04077
```

Yes there is. Perhaps surprisingly, there is a highly significant interaction between Insect and Mollusc in their effects on the slope of the graph of Grass biomass against Time. We need to see the values involved:

```
tapply(aa,list(m,i),mean)
```

```
                  Sprayed  Unsprayed
 Absent        -3.569489  -3.203488
Present        -2.738241  -3.237609
```

All the average slopes are negative, but insect exclusion reduces the slope under one mollusc treatment but increases it under another. The mechanisms underlying this interaction would clearly repay further investigation.

The statistical lesson is that derived variable analysis provided many more insights than either the 6 separate analyses or the analysis of the time-averaged data.

The final technique we shall use is *mixed effects* modelling. This requires a clear understanding of which of our 4 factors are **fixed effects** and which are **random effects**. We applied the insecticide and the molluscicide to plots within Blocks at random, so Mollusc and Insect are *fixed* effects. Blocks are different, but we didn't make them different. They are assumed to come from a population of different locations, and hence are random effects. Things differ through time, but mainly because of the weather, and the passage of the seasons, rather than because of anything we do. So time, as well, is a *random* effect. One of our random effects is **spatial** (Block) and one **temporal** (Time). The factorial experiment (Mollusc*Insect) is nested within each block, and Time codes for repeated measures made within each treatment plot within each Block.

It is worth re-emphasising that if the smallest plot size are pseudoreplicates (rather than small-plot fixed effect treatments), then they should **not** be included as the last term in the Error directive. See the rat's liver example (above), where we did not include Preparations (the measurement error) in the error formula (and it would be a mistake to do so).

Mixed effects models

Linear mixed effects models and the groupedData directive

If you have not read the data file repeated.txt earlier, you should input it now:

```
library(nlme)

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

```
[1] "Block"   "Time"    "Insect"  "Mollusc" "Grass"
```

We need to turn this data frame into grouped data. The key thing to understand is where, exactly, each of the 5 variables (above) is placed in a mixed effects model. We create a formula of the form resp ~ cov | group where resp is the response variable, cov is the primary covariate, and group is the grouping factor.

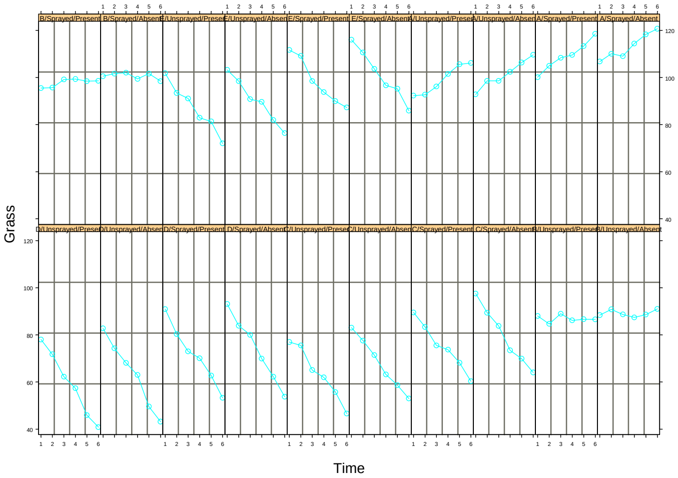
The response variable is easy: it is Grass. This goes on the left of the ~ in the model formula. The manipulated parts of the experiment (the fixed effects) are Insect and Mollusc; 2-level factors representing plus and minus each kind of herbivore. These factors that apply to the whole experiment are called the **outer** factors. They appear in a model formula (linked in this case by the factorial operator *) to the right of the ~ in the outer statement. But what about Block and Time? You can easily imagine a graph of Grass against Time. This is a time series plot, and each treatment in each Block could produce one such time series. Thus, Time goes on the right of the ~ in the model formula. Block is the spatial unit in which the pseudoreplication has occurred, so it is the factor that goes on the right of the conditioning operator | (vertical bar) after the model formula. In general, the conditioning factors can be nested, using the / (slash operator). In medical studies the conditioning factor is often the Subject from which repeated measures were taken.

The new thing here is the **groupedData** directive. It is constructed like this to create a new data frame called nash:

```
nash<-groupedData(Grass~Time|Block/Insect/Mollusc,data=repeated,outer=~Insect*Mollusc)
```

With grouped data, some previously complicated things become very simple. For instance, panel plots are produced automatically.

```
plot(nash)
```



It shows the data for the whole factorial experiment (2 levels of Insecticide and 2 levels of Molluscicide) in a single group of panels (one for each Block). What it shows very clearly, however, are the different time series exhibited in the 5 different Blocks. Note that **groupedData** has ordered the blocks from lowest to highest mean grass biomass (D, C, B, A & E) and that grass biomass increases through time on Block E, is roughly constant on Block B and declines on Blocks D, C and A. To see the time series for the 4 combinations of fixed effects, we use the outer=T option within the plot directive:

```
mixed<-lme(fixed=Grass~Insect*Mollusc,data=nash,random=~Time|Block)

summary(mixed)
```

```
Linear mixed-effects model fit by REML
 Data: nash
       AIC      BIC    logLik
  538.3928 560.4215 -261.1964

Random effects:
 Formula:  ~ Time | Block
 Structure: General positive-definite
              StdDev   Corr
(Intercept) 8.885701 (Inter
       Time 5.193374 -0.261
   Residual 1.644065

Fixed effects: Grass ~ Insect * Mollusc
                   Value Std.Error DF   t-value p-value
   (Intercept)  96.14375  3.848244 112  24.98380  <.0001
        Insect  -5.64657  0.150082 112 -37.62326  <.0001
       Mollusc  -1.61923  0.150082 112 -10.78899  <.0001
Insect:Mollusc   0.02751  0.150082 112   0.18327  0.8549
 Correlation:
               (Intr) Insect Mollsc
        Insect 0
       Mollusc 0      0
Insect:Mollusc 0      0      0

Standardized Within-Group Residuals:
       Min          Q1          Med         Q3       Max
 -1.877155 -0.8299427 -0.01336901 0.7337527 2.253834

Number of Observations: 120
Number of Groups: 5
```

The interpretation is unequivocal: the mixed effects model gives no indication of an interaction between insect and mollusc exclusion (p = 0.8549). The effects we saw in the earlier analyses were confounded with block effects.

Contrasts: Single degree of freedom comparisons

Once the anova table has been completed and the F test carried out to establish that there are indeed significant differences between the means, it is reasonable to ask which factor levels are significantly different from which others. This subject is developed in more detail in the chapter on Multiple Comparisons (see Statistical computing p. 274); here we introduce the technique of contrasts (also known as single degree of freedom comparisons). There are two sorts of contrasts we might want to carry out

- contrasts we had planned to carry out at the experimental design stage (these are referred to as *a priori* contrasts)
- contrasts that look interesting after we have seen the results (these are referred to as *a posteriori* contrasts)

Some people are very snooty about *a posteriori* contrasts, but you can't change human nature. The key point is that you do contrasts *after* the anova has established that there really are significant differences to be investigated. It is not good practice to carry out tests to compare the largest mean with the smallest mean if the anova fails to reject the null hypothesis (tempting though this may be).

Contrasts are used to compare means or groups of means with other means or groups of means. In a drug trial classified by cities, for example, you might want to compare the response to the drugs for all mid west cities with all west coast cities. There are two important points to understand about contrasts:

- there are absolutely loads of possible contrasts
- there are only *k*-1 orthogonal contrasts

Lets take a simple example. Suppose we have one factor with 5 levels and the factor levels are called *a, b, c, d,* and *e*. Let's start writing down the possible contrasts. Obviously we could compare each mean singly with every other:

a vs. b, a vs. c, a vs. d, a vs. e, b vs. c, b vs. d, b vs. e, c vs. d, c vs. e, d vs. e

but we could also compare pairs of means:

{a,b} vs. {c,d}, {a,b} vs. {c,e}, {a,b} vs. {d, e}, {a,c} vs. {b, d}, {a,c} vs. {b,e}, etc.

or triplets of means:

{a,b,c} vs. d, {a,b,c} vs. e, {a,b,d} vs. c, {a,b,d} vs. e, {a,c,d} vs. b, and so on.

or groups of four means

{a,b,c,d} vs. e, {a,b,c,e} vs. d, {b,c,d,e} vs. a, {a,b,d,e} vs c, {a,b,c,e} vs d

I think you get the idea. There are absolutely loads of possible contrasts.

Orthogonal contrasts are different, and it is important that you understand how and why they are different. We refer to the number of factor levels as *k* (this was 5 in the last example). Out of all the many possible contrasts, only *k*-1 of them are orthogonal. In this context, orthogonal means "statistically independent" (it also means "at right angles" which in mathematical terms is another way of saying statistically independent). In practice we should only compare things once, either directly or implicitly. So the two contrasts:

a vs. b and *a vs. c*

implicitly contrasts *b vs. c*. This means that if we have carried out the two contrasts *a vs. b* and *a vs. c* then the third contrast *b vs. c* is **not** an orthogonal contrast. Which particular contrasts are orthogonal depends very much on your choice of the first contrast to make. Suppose there were good reasons for comparing *{a,b,c,e} vs. d*. For example, *d* might be the placebo and the other 4 might be different kinds of drug treatment, so we make this our first contrast. Because *k*-1 = 4 we only have 3 possible contrasts that are orthogonal to this. There may be *a priori* reasons to group *{a,b}* and *{c,e}* so we make this our second orthogonal contrast. This means that we have no degrees of freedom in choosing the last 2 orthogonal contrasts: they have to be *a vs. b* and *c vs. e*.

Just remember that with orthogonal contrasts you only compare things once.

Contrast coefficients

Rules for constructing contrast coefficients are straightforward:

- treatments to be lumped together get like sign (plus or minus)
- groups of means to be to be contrasted get opposite sign
- factor levels to be excluded get a contrast coefficient of 0
- the contrast coefficients, *c*, must add up to 0

Suppose that with our 5-level factor {*a,b,c,d,e*} we want to begin by comparing the 4 levels {*a,b,c,e*} with the single level *d*. All levels enter the contrast, so none of the coefficients is 0. The four terms {*a,b,c,e*} are grouped together so they all get the same sign (minus, for example, although it makes not matter which sign is chosen). They are to be compared to *d*, so it gets the opposite sign (plus, in this case). The choice of what numeric values to give the contrast coefficients is entirely up to you. Most people use whole numbers rather than fractions, but it really doesn't matter. All that matters is that the c's sum to 0. The positive and negative coefficients have to add up to the same value. In our example, comparing 4 means with one mean, a natural choice of coefficients would be -1 for each of {*a,b,c,e*} and +4 for *d*. Alternatively with could have selected +0.25 for each of {*a,b,c,e*} and -1 for *d*. It really doesn't matter.

factor level:	*a*	*b*	*c*	*d*	*e*
contrast 1 coefficients, c:	-1	-1	-1	4	-1

Suppose the second contrast is to compare {*a*,*b*} with {*c*,*e*}. Because this contrast excludes *d*, we set its contrast coefficient to 0. {*a*,*b*} get the same sign (say, plus) and {*c*,*e*} get the opposite sign. Because the number of levels on each side of the contrast is equal (2 in both cases) we can use the name numeric value for all the coefficients. The value 1 is the most obvious choice (but you could use 13.7 if you wanted to be perverse).

factor level:	*a*	*b*	*c*	*d*	*e*
contrast 2 coefficients, c:	1	1	-1	0	-1

There are only 2 possibilities for the remaining orthogonal contrasts: *a vs. b* and *c vs. e*:

factor level:	*a*	*b*	*c*	*d*	*e*
contrast 3 coefficients, c:	1	-1	0	0	0
contrast 4 coefficients, c:	0	0	1	0	-1

The key point to understand is that the treatment sum of squares SSA is the sum of the k-1 orthogonal sums of squares. It is useful to know which of the contrasts contributes most to SSA, and to work this out, we compute the contrast sum of squares SSC as follows:

$$SSC = \frac{\left(\sum \frac{c_i T_i}{n_i}\right)^2}{\sum \frac{c_i^2}{n_i}}$$

The significance of a contrast is judged in the usual way by carrying out an F test to compare the contrast variance with the error variance, s^2. Since all contrasts have a single degree of freedom, the contrast variance is equal to SSC, so the F test is just

$$F = \frac{SSC}{s^2}$$

The contrast is significant (i.e. the two contrasted groups have significantly different means) if the calculated value is larger than the value of F in tables with 1 and *k*(*n*-1) degrees of freedom.

An example should make all this clearer. Suppose we have a plant ecology experiment with 5 treatments: a control, a shoot-clipping treatment where 25% of the neighbouring plants are clipped back to ground level, another where 50% on neighbours are clipped back, a 4th where a circle of roots around the target plant is pruned to a depth of 10cm and a 5th with root pruning to only 5cm depth. The data look like this:

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

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

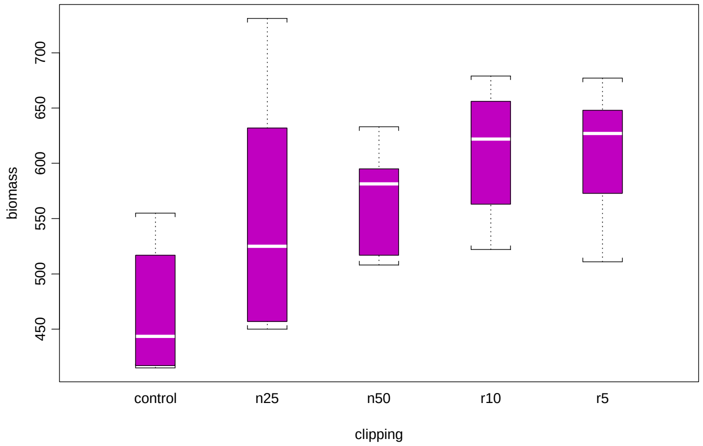
```
levels(clipping)
```

```
[1] "control" "n25"     "n50"     "r10"     "r5"
```

```
tapply(biomass,clipping,mean)
```

```
  control      n25      n50      r10    r5
 465.1667 553.3333 569.3333 610.6667 610.5
```

```
plot(clipping,biomass)
```



The treated plant means are all higher than the controls, suggesting that competition was important in this system. The root pruned plants were larger than the shoot pruned plants, suggesting that below ground competition might be more influential than above ground. The different intensities of shoot pruning differed more than the different intensities of root pruning. It remains to see whether these differences are significant by using contrasts. We can do this long-hand by recoding the factor levels, or make use of SPlus built in facilities for defining contrasts.

We begin by re-coding the factor levels to calculate the contrast sums of squares. First, the overall anova to see whether there are any significant differences at all

```
model1<-aov(biomass~clipping)

summary(model1)
```

```
          Df Sum of Sq  Mean Sq  F Value       Pr(F)
 clipping  4   85356.5 21339.12 4.301536 0.008751641
Residuals 25  124020.3  4960.81
```

Yes. There are highly significant differences ($p < 0.01$). We reject the null hypothesis that all the means are the same, and accept the alternative hypothesis that at least one of the means is significantly different from the others. This is not a particularly informative alternative hypothesis, however. We want to know what, exactly, the experiment has shown.

The treatment sum of squares, SSA = 85356.6 is made up by *k*-1 = 4 orthogonal contrasts. The most obvious contrast to try first is the control versus the rest. We compute a new factor, c1, to reflect this. It has value 1 for all the competition treatments and 2 for the controls:

```
c1<-factor(1+(clipping=="control"))
```

Now we fit this as the single explanatory variable in a one-way anova:

```
model2<-aov(biomass~c1)
summary(model2)
```

```
          Df Sum of Sq  Mean Sq  F Value        Pr(F)
       c1  1   70035.0 70035.01 14.07317 0.0008149398
Residuals 28  139341.8  4976.49
```

This contrast explained 70035.0 out of the total SSA = 85356.5, so it is clear that competition was highly significant. What about the difference between defoliation and root pruning (light versus below-ground competition). The computing is a little more complicated in this case because be need to **weight out** the control individuals from the analysis as well as calculating a new factor to compare “n25” and “n50” with “r10” and “r5”. Note the use of “!=” for “not equal to” in the weight directive:

```
c2<-factor(1+(clipping=="r10")+(clipping=="r5"))
model3<-aov(biomass~c2,weight=(clipping!="control"))
summary(model3)
```

```
          Df Sum of Sq  Mean Sq  F Value      Pr(F)
       c2  1   14553.4 14553.38 2.959217 0.09942696
Residuals 22  108195.6  4917.98
```

This contrast with SSC = 14553.4 is not significant. Between them, the first two contrasts explain 14553.4 + 70035 = 84588.4 out of the total explained variation of SSA = 85356.5. The other two untested (but obviously not significant) orthogonal contrasts are “n25” vs. “n50” and “r10” vs “r5”.

There are sophisticated built-in functions in SPlus for defining and executing contrasts. The contrast coefficients are specified as attributes of the factor called clipping like this

```
contrasts(clipping)<-cbind(c(4,-1,-1,-1,-1),c(0,1,1,-1,-1),c(0,0,0,1,-1),c(0,-1,1,0,0))
```

where we bind the relevant contrast vectors together using **cbind**. To inspect the contrasts association with any factor we just type:

```
contrasts(clipping)
```

```
        [,1] [,2] [,3] [,4]
control    4    0    0    0
    n25   -1    1    0   -1
    n50   -1    1    0    1
    r10   -1   -1    1    0
     r5   -1   -1   -1    0
```

Notice that all of the column totals sum to zero as required ($\sum c_i = 0$, see above). You can see that the different contrasts are all **orthogonal** because *the products of their coefficients all sum to zero*. For example, comparing contrasts 1 and 2 we have

$$(4\times 0)+(-1\times 1)+(-1\times 1)+(-1\times -1)+(-1\times -1) \;\;=\;\; -1+-1+1+1 \;\;=\;\; 0$$

Now when we carry out a one way anova using the factor called clipping, the parameter estimates reflect the differences between the contrasted group means rather than differences between the individual treatment means.

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

We use the function **summary.lm** to obtain a listing of the coefficients for the 4 contrasts we have specified:

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

```
Coefficients:
                 Value Std. Error   t value  Pr(>|t|)
(Intercept)   561.8000    12.8593   43.6884    0.0000
  clipping1   -24.1583     6.4296   -3.7573    0.0009
  clipping2   -24.6250    14.3771   -1.7128    0.0991
  clipping3     0.0833    20.3323    0.0041    0.9968
  clipping4     8.0000    20.3323    0.3935    0.6973
```

This gives all the information we need. The first row shows the overall mean biomass (561.8). The second row shows clipping contrast 1: the overall mean versus the 4 competition levels (585.958). The overall mean was 24.1583 g below the mean of the 4 competition treatments (561.8 – 585.958 = -24.1583), and this difference is highly significant (p = 0.0009). The 3rd row shows the effect of the 2nd contrast between clipped and root-pruned plants. This contrast is not significant (p = 0.0991) despite the fact that the clipped plants produced an average biomass of about 49.25 g less than the root pruned plants (the coefficient (-24.625) is *half* of the difference between the 2 groups of means (561.333 – 610.585) because all the contrasts are calculated relative to overall means (in this case the mean of {a,b,c,e} excluding the controls, 585.96). There is no hint of any difference between the different intensities of root pruning 610.67 – 610.585 = 0.0833 (p = 0.9968) or shoot clipping 569.333 – 561.333 = 8.0 (p = 0.6973), despite the fact that the means differ by 16.0 g (569.333 – 553.333 = 16).

You may have noticed that the standard errors are different in four of the 5 rows of the coefficients table. All the standard errors are based on the same pooled error variance of s^2 = 4960.813. For the grand mean, based on 30 samples, the standard error is $\sqrt{\frac{4960.813}{30}} = 12..859$. The first contrast (row 2) compares a group of 4 means with the overall mean (a comparison based notionally on 30 x 4 = 120 numbers) so the standard error is $\sqrt{\frac{4960.813}{120}} = 6.4296$. The second contrast (row 3) compares the two defoliation means with the two root pruning means (a total of 24 numbers) so the standard error is $\sqrt{\frac{4960.813}{24}} = 14.3771$. The third and fourth contrasts (rows 4 & 5) both involve the comparison of one mean with another (a comparison based on 6 + 6 = 12 numbers) and therefore they have the same standard error $\sqrt{\frac{4960.813}{12}} = 20.3323$.

The contrasts that are built into SPlus are Helmert, Sum and Treatment, and we should compare the output produced by each. First we need to remove our own contrasts from the factor called clipping:

```
contrasts(clipping)<-NULL
```

Helmert contrasts

These are the default option in SPlus (but not in R) and are preferred because they represent orthogonal contrasts (the columns of the contrast matrix sum to 0).

```
options(contrasts=c("contr.helmert","contr.poly"))
model2<-lm(biomass~clipping)
summary(model2)
```

Note that each of the standard errors is different, because one mean is being compared with an increasingly larger group of means (so replication is higher, and the standard error is correspondingly lower, as explained below).

```
Coefficients:
               Value Std. Error  t value Pr(>|t|)
(Intercept) 561.8000  12.8593    43.6884   0.0000
  clipping1  44.0833  20.3323     2.1681   0.0399
  clipping2  20.0278  11.7388     1.7061   0.1004
  clipping3  20.3472   8.3006     2.4513   0.0216
  clipping4  12.1750   6.4296     1.8936   0.0699
```

Now you are not going to like this, but we are going to understand exactly where each of the coefficients comes from. It will be useful to remind ourselves of the overall mean (561.8) and the 5 treatment means to begin with:

```
tapply(biomass,clipping,mean)
```

```
  control      n25      n50      r10    r5
 465.1667 553.3333 569.3333 610.6667 610.5
```

The first parameter (labelled Intercept) is the overall mean. That’s the easy part. The second term (labelled clipping1) is the difference between the mean of clipping treatment 1 (465.1667) and the *average of the means* of treatments 1 and 2: (465.1667+553.333)/2 = 509.25.

```
509.25  -  465.1667
[1] 44.0833
```

The third term is the difference between the *average of the first 3* clipping treatments (465.1667 + 553.333 + 569.333)/3 = 529.2778) and the *average of the first two* means (already calculated as 509.25):

```
529.2778  -  509.25
[1] 20.0278
```

Are you getting this? The 4th term is the *average of the first 4* clipping treatments (465.1667 + 553.333 + 569.333 + 610.667) / 4 = 549.525) minus the *average of the first 3* (529.2778, above)

```
549.625  -  529.2778
[1] 20.3472
```

The last coefficient in table is the difference between the *overall mean* (561.8) and the *mean of the first 4* treatments (just calculated, above):

```
561.8  -  549.625
[1] 12.175
```

I hope you understood all that. Statisticians like Helmert contrasts because they are *proper* contrasts: the columns of the contrast matrix sum to zero. No one else likes them, because they are a pain in the neck.

Sum contrasts

```
options(contrasts=c("contr.sum","contr.poly"))
model3<-lm(biomass~clipping)
summary(model3)
```

```
Coefficients:
                Value Std. Error   t value  Pr(>|t|)
(Intercept)  561.8000   12.8593    43.6884    0.0000
  clipping1  -96.6333   25.7185    -3.7573    0.0009
  clipping2   -8.4667   25.7185    -0.3292    0.7447
  clipping3    7.5333   25.7185     0.2929    0.7720
  clipping4   48.8667   25.7185     1.9001    0.0690
```

As with Helmert contrasts, the intercept is the overall mean. The second row, clipping contrast 1, is the difference between the grand mean and the mean of the controls: 561.8 – 96.6333 = 465.1667. The third row is the difference between the grand mean and the mean of clipping treatment 2: 561.8 – 8.4667 = 553.333. And so on. The standard error of the intercept is the same as with Helmert contrasts: $\sqrt{s^2 / 30}$. All the other standard errors are the same $\sqrt{\frac{4960.813}{12}+\frac{4960.813}{20}}$ = 25.7185

Treatment contrasts

```
options(contrasts=c("contr.treatment","contr.poly"))
model<-lm(biomass~clipping)
summary(model)
```

These are the default contrasts in R and in GLIM. The first coefficient is *the mean of the factor level that comes first in alphabetical order* (the control mean in this example). The *remaining parameters are all differences between means* (the mean in question compared with the control mean). The standard error of the intercept is the standard error of one treatment mean. It is based on 6 replicates, so the standard error is $\sqrt{\frac{s^2}{6}}$ =28.7542 . The remaining parameters are differences between two means, so their standard errors are $\sqrt{2\frac{s^2}{6}}$ = 40.6645.

```
Coefficients:
                Value Std. Error  t value Pr(>|t|)
(Intercept) 465.1667  28.7542    16.1774   0.0000
```

```
clippingn25  88.1667  40.6645      2.1681   0.0399
clippingn50 104.1667  40.6645      2.5616   0.0168
clippingr10 145.5000  40.6645      3.5781   0.0015
 clippingr5 145.3333  40.6645      3.5740   0.0015
```

Obviously, you like what you are used to. But I find these treatment contrasts much more intuitive and much more useful than sum and Helmert contrasts. I can compare any means with any other directly. For example r10 and r5 differ by only 0.1666 (you do the subtraction in your head) and this has a standard error of a difference which is 40.66. Say no more. Neighbour clipping to 25 and 50 cm differ by 16.0 with the same standard error. Not significant. And so on. Easy and intuitive. The downside is that the probabilities in the coefficients table don't refer directly to the need to retain a particular parameter in the model (they refer to the significance of the comparison between that level and level 1, the intercept). To summarise, here is a table comparing the standard error terms used in our own contrast and in the 3 built-in contrasts:

Parameter	Ours	Helm.	Sum	Treat.	Standard error
Grand mean	*	*	*		$\sqrt{\frac{4960.813}{30}} = 12..859$
One treatment mean				*	$\sqrt{\frac{4960.813}{6}} = 28.754$
Difference between 2 means				*	$\sqrt{2 \times \frac{4960.813}{6}} = 40.665$
Averaged over 2 treatments, n = 6	*	*			$\sqrt{\frac{4960.813}{12}} = 20.3323$
Averaged over 4 treatments, n = 6	*				$\sqrt{\frac{4960.813}{24}} = 14.3771$
Averaged over 6 treatments, n = 6		*			$\sqrt{\frac{4960.813}{36}} = 11.7388$
Averaged over 12 treatments, n = 6		*			$\sqrt{\frac{4960.813}{72}} = 8.3006$
Averaged over 20 treatments, n = 6	*	*			$\sqrt{\frac{4960.813}{120}} = 6.4296$
Sums			*		$\sqrt{\frac{4960.813}{12} + \frac{4960.813}{20}}$

Contrasts in analysis of covariance

Remember that the minimal adequate model has *a common slope* and *two different intercepts* for the relationship between fruit production and initial rootstock size.

The important point to understand is that using different contrasts gives you

- different parameter values
- different standard errors

so you need to be extremely careful that you know what is going on. It is safest to use the same form of contrasts in all of your work, then you won’t be misled. The default in S-Plus is Helmert contrasts and the default in R is Treatment contrasts. Old GLIM users will be familiar with Treatment contrasts.

1) Helmert contrasts

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

options(contrasts=c("contr.helmert","contr.poly"))
modelH<-lm(Fruit~Root+Grazing)
summary(modelH)
```

The Intercept is the *average of the two intercepts* (-127. 83 and –91.73). The effect of root is the slope of the graph of Fruit against Root (this is the same with all 3 contrasts). The effect of grazing is the difference (+18.0516) between the Grazed intercept (-127.83) and the average intercept (-109.777); i.e. *half the difference between the two different intercepts*.

```
Coefficients:
                Value Std. Error   t value  Pr(>|t|)
(Intercept) -109.7777    8.3182   -13.1973    0.0000
       Root   23.5600    1.1488    20.5089    0.0000
    Grazing   18.0516    1.6787    10.7533    0.0000
```

Sum contrasts

```
options(contrasts=c("contr.sum","contr.poly"))
modelS<-lm(Fruit~Root+Grazing)
summary(modelS)
```

The coefficients are exactly the same as for Helmert contrasts, except for the fact that the Grazing effect is sign-reversed: it is the difference between the Ungrazed intercept (-91.73) and the average intercept.

```
Coefficients:
                 Value Std. Error   t value  Pr(>|t|)
(Intercept) -109.7777    8.3182   -13.1973    0.0000
       Root   23.5600    1.1488    20.5089    0.0000
    Grazing  -18.0516    1.6787   -10.7533    0.0000
```

Treatment contrasts

```
options(contrasts=c("contr.treatment","contr.poly"))
modelT<-lm(Fruit~Root+Grazing)
summary(modelT)
```

Coefficients:

```
                  Value Std. Error   t value  Pr(>|t|)
(Intercept) -127.8294    9.6641   -13.2272    0.0000
       Root   23.5600    1.1488    20.5089    0.0000
    Grazing   36.1032    3.3574    10.7533    0.0000
```

Here the Intercept is the intercept for the factor level that comes first in the alphabet (Grazed in this case). The second parameter (Root) is the slope of the graph of Fruit against Root. The third parameter (labelled Grazing) is the difference between the 2 intercepts. The effect of grazing is to reduce fruit production, so Ungrazed plants have an intercept 36.1032 higher than Grazed plants. The standard error for Root is the *standard error of a slope* while the standard error for grazing is the *standard error of the difference between two intercepts* (see our hand calculation in Practical 6).