12

Contrasts

Contrasts are the essence of hypothesis testing and model simplification in Anova. They are used to compare means or groups of means with other means or groups of means, in what are 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), or
- 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, on the grounds that they were **unplanned**. You are not supposed to decide what comparisons to make **after** you have seen the analysis, but scientists do this all the time – you cannot change human nature. The key point is that you should only 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).

There are two important points to understand about contrasts:

- there is a huge number of **possible** contrasts,
- there are only $k - 1$ **orthogonal** contrasts.

where k is the number of factor levels. Two contrasts are said to be orthogonal to one another if the comparisons are statistically independent. Technically, two contrasts are orthogonal if **the products of their contrast coefficients sum to zero** (we shall see what this means in a moment).

Let's take a simple example. Suppose we have one factor with five 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:

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

Statistics: An Introduction using R M. J. Crawley
 ISBNs: 0-470-02298-1 (PBK); 0-470-02297-3 (PPC)

but we could also compare pairs of means:

$$\{a,b\}\ vs.\ \{c,d\},\ \{a,b\}\ vs.\ \{c,e\},\ \{a,b\}\ vs.\ \{d,\ e\},\ \{a,c\}\ vs.\ \{b,\ d\},\ \{a,c\}\ vs.\ \{b,e\},\ \text{etc.}$$

or triplets of means:

$$\{a,b,c\}\ vs.\ d,\ \{a,b,c\}\ vs.\ e,\ \{a,b,d\}\ vs.\ c,\ \{a,b,d\}\ vs.\ e,\ \{a,c,d\}\ vs.\ b,\ \text{etc.}$$

or groups of four means:

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

You are probably getting the idea. There are absolutely masses of possible contrasts. In practice, however, we should only compare things once, either directly or implicitly. So the two contrasts:

$$a\ vs.\ b \text{ and } a\ vs.\ c$$

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 because you have already carried it out, implicitly. 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 four might be different kinds of drug treatment, so we make this our first contrast. Because $k - 1 = 4$ we only have three 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 two 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

Contrast coefficients are a numerical way of embodying the hypothesis we want to test. The rules for constructing contrast coefficients are straightforward:

- treatments to be lumped together get the same sign (plus or minus),
- groups of means contrasted get the 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 five-level factor {*a,b,c,d,e*} we want to begin by comparing the four 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 no difference which sign is chosen).

They are to be compared with 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 add up to 0. The positive and negative coefficients have to add up to the same value. In our example, comparing four means with one mean, a natural choice of coefficients would be -1 for each of $\{a,b,c,e\}$ and $+4$ for d. Alternatively we could have selected $+0.25$ for each of $\{a,b,c,e\}$ and -1 for d.

factor level:	a	b	c	d	e
contrast one 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 (two 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 two coefficients, c:	1	1	-1	0	-1

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

factor level:	a	b	c	d	e
contrast three coefficients, c:	1	-1	0	0	0
contrast four coefficients, c:	0	0	1	0	-1

An Example of Contrasts in R

The example comes from the competition experiment we analysed in Chapter 9 in which the biomass of control plants is compared with the biomass of plants grown in conditions where competition was reduced in one of four different ways. There are two treatments in which the roots of neighbouring plants were cut (to 5 cm depth or 10 cm) and two treatments in which the shoots of neighbouring plants were clipped (25% or 50% of the neighbours cut back to ground level; see p. 167).

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

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

We start with the one-way analysis of variance:

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

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

Clipping treatment has a highly significant effect on biomass – but have we fully understood the result of this experiment? Probably not. For example, which factor levels had the biggest effect on biomass, and were all of the competition treatments significantly different from the controls? To answer these questions, we need to use summary.lm:

```
summary.lm(model1)
```

```
Coefficients:
             Estimate  Std. Error  t value  Pr(>|t|)
(Intercept)    465.17       28.75  16.177  9.33e-15  ***
clippingn25     88.17       40.66   2.168   0.03987   *
clippingn50    104.17       40.66   2.562   0.01683   *
clippingr10    145.50       40.66   3.578   0.00145   **
clippingr5     145.33       40.66   3.574   0.00147   **

Residual standard error: 70.43 on 25 degrees of freedom
Multiple R-Squared: 0.4077,    Adjusted R-squared: 0.3129
F-statistic: 4.302 on 4 and 25 DF,    p-value:  0.008752
```

This looks as if we need to keep all five parameters, because all five rows of the summary table have one or more significance stars. If fact, this is not the case. This example highlights the major shortcoming of **treatment contrasts**: they do not show how many significant factor levels we need to retain in the minimal adequate model.

A Priori Contrasts

In this experiment, there are several planned comparisons we should like to make. The obvious place to start is by comparing the control plants that were exposed to the full rigours of competition, with all of the other treatments.

```
levels(clipping)
```

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

That is to say, we want to contrast the first level of clipping with the other four levels. The contrast coefficients, therefore, would be 4, −1, −1, −1, −1. The next planned comparison might contrast the shoot-pruned treatments (n25 and n50) with the root-pruned treatments (r10 and r5). Suitable contrast coefficients for this would be 0, 1, 1, −1, −1 (because we are ignoring the control in this contrast). A third contrast might compare the two depths of root-pruning; 0, 0, 0, 1, −1. The last orthogonal contrast would therefore have to compare the two intensities of shoot-pruning: 0, 1, −1, 0, 0. Because the factor called 'clipping' has five levels there are only $5 - 1 = 4$ orthogonal contrasts.

R is outstandingly good at dealing with contrasts, and we can associate these five user-specified *a priori* contrasts with the categorical variable 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))
```

We can check that this has done what we wanted by typing

```
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
```

which produces the matrix of contrast coefficients that we specified. Note that all the columns add to zero (i.e. each set of contrast coefficients is correctly specified). Note also that the products of any two of the columns sum to zero (this shows that all the contrasts are orthogonal, as intended), e.g. comparing contrasts 1 and 2 gives products $0+(-1)+(-1)+1+1=0$.

Now we can re-fit the model and inspect the results of our specified contrasts, rather than the default treatment contrasts:

```
model2 <-aov(biomass ~clipping)
summary.lm(model2)
```

```
Coefficients:
             Estimate  Std. Error  t value   Pr(>|t|)
(Intercept)  561.80000   12.85926   43.688     <2e-16 ***
clipping1    -24.15833    6.42963   -3.757   0.000921 ***
clipping2    -24.62500   14.37708   -1.713   0.099128 .
clipping3      0.08333   20.33227    0.004   0.996762
clipping4     -8.00000   20.33227   -0.393   0.697313

Residual standard error: 70.43 on 25 degrees of freedom
Multiple R-Squared: 0.4077,     Adjusted R-squared: 0.3129
F-statistic: 4.302 on 4 and 25 DF,  p-value: 0.008752
```

Instead of requiring five parameters (as suggested by out initial treatment contrasts), this analysis shows that we need only two parameters: the overall mean (561.8) and the contrast between the controls and the four competition treatments ($p = 0.000921$). All the other contrasts are non-significant.

Model Simplification by Step-wise Deletion

An alternative to specifying the contrasts ourselves (as above) is to aggregate non-significant factor levels in a step-wise *a posteriori* procedure. To demonstrate this, we revert to treatment contrasts:

```
contrasts(clipping) <- -NULL
options(contrasts = c("contr.treatment","contr.poly"))
```

Now we fit the model with all five factor levels as a starting point:

```
model3 <- -aov(biomass ~ clipping)
summary.lm(model3)
```

```
Coefficients:
              Estimate  Std. Error  t value   Pr(>|t|)
(Intercept)     465.17       28.75   16.177   9.33e-15 ***
clippingn25      88.17       40.66    2.168    0.03987   *
clippingn50     104.17       40.66    2.562    0.01683   *
clippingr10     145.50       40.66    3.578    0.00145  **
clippingr5      145.33       40.66    3.574    0.00147  **
```

Looking down the list of parameter estimates, we see that the most similar are the effects of root pruning to 10 and 5 cm (145.5 vs. 145.33). We shall begin by simplifying these to a single root-pruning treatment called root. The trick is to use 'levels gets' to change the names of the appropriate factor levels. Start by copying the original factor name:

```
clip2 <- -clipping
```

Now inspect the level numbers of the various factor level names:

```
levels(clip2)
```

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

The plan is to lump together r10 and r5 under the same name, 'root'. These are the fourth and fifth levels of clip2, so we write:

```
levels(clip2)[4:5] <- -"root"
```

and to see what has happened type

```
levels(clip2)
```

```
[1] "control"  "n25"  "n50"  "root"
```

and we see that 'r10' and 'r5' have indeed been replaced by 'root'. The next step is to fit a new model with clip2 in place of clipping, and to test whether the new simpler model is significantly worse as a description of the data using anova:

```
model4 <- aov(biomass ~ clip2)
anova(model3,model4)
```

```
Analysis of Variance Table
Model 1: biomass~clipping
Model 2: biomass~clip2
Res.Df     RSS  Df   Sum of Sq          F     Pr(>F)
1    25  124020
2    26  124020  -1 -0.0833333  0.0000168     0.9968
```

As we expected, this model simplification was completely justified. The next step is to investigate the effects using summary.lm:

```
summary.lm(model4)
```

```
Coefficients:
             Estimate  Std. Error  t value   Pr(>|t|)
(Intercept)    465.17       28.20   16.498   2.66e-15 ***
clip2n25        88.17       39.87    2.211   0.036029   *
clip2n50       104.17       39.87    2.612   0.014744   *
clip2root      145.42       34.53    4.211   0.000269 ***
```

It looks as if the two shoot clipping treatments (n25 and n50) are not significantly different from one another (they differ by just 16.0 with a standard error of 39.87). We can lump these together into a single shoot-pruning treatment as follows:

```
clip3 <- clip2
levels(clip3)[2:3] <- "shoot"
levels(clip3)
```

```
[1] "control"  "shoot"  "root"
```

Then fit a new model with clip3 in place of clip2:

```
model5 <- aov(biomass ~ clip3)
anova(model4,model5)
```

```
Analysis of Variance Table

Model 1: biomass~clip2
Model 2: biomass~clip3
     Res. Df      RSS   Df  Sum of Sq        F  Pr(>F)
1         26   124020
2         27   124788   -1       -768    0.161  0.6915
```

Again, this simplification was fully justified. Do the root and shoot competition treatments differ?

```
clip4 <- clip3
levels(clip4)[2:3] <- "pruned"
levels(clip4)
```

```
[1] "control"  "pruned"
```

Now fit a new model with clip4 in place of clip3:

```
model6 <- aov(biomass ~ clip4)
anova(model5,model6)
```

```
Analysis of Variance Table

Model 1: biomass~clip3
Model 2: biomass~clip4
    Res.Df      RSS   Df  Sum of Sq        F    Pr(>F)
1       27   124788
2       28   139342   -1    -14553   3.1489  0.08726.
```

This simplification was close to significant, but we are ruthless ($p > 0.05$, so we accept the simplification). Now we have the minimal adequate model:

```
summary.lm(model6)
```

```
Coefficients:
             Estimate  Std. Error  t value   Pr(>|t|)
(Intercept)     465.2        28.8   16.152   1.11e-15 ***
clip4pruned     120.8        32.2    3.751   0.000815 ***
```

it has just two parameters: the mean for the controls (465.2) and the difference between the control mean and the four treatment means ($465.2 + 120.8 = 586.0$):

```
tapply(biomass,clip4,mean)
```

```
 control    pruned
465.1667  585.9583
```

We know that these two means are significantly different from the p value $= 0.000815$, but just to show how it is done, we can make a final model 7 that has no explanatory variable at all (it fits only the overall mean). This is achieved by writing `y ~ 1` in the model formula:

```
model7 <- aov(biomass ~ 1)
anova(model6,model7)
```

```
Analysis of Variance Table

Model 1: biomass~clip4
Model 2: biomass~1
```

```
    Res.Df      RSS    Df  Sum of Sq       F     Pr(>F)
1       28   139342
2       29   209377    -1     -70035  14.073   0.000815  ***
```

Note that the p value is exactly the same as in model 6. The p values in R are calculated such that they avoid the need for this final step in model simplification: they are 'p on deletion' values.

Contrast Sums of Squares by Hand

The key point to understand is that **the treatment sum of squares *SSA* is the sum of all $(k-1)$ 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

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

where the error variance, s^2, comes from the error mean square column of the Anova table. The contrast is significant (i.e. the two contrasted groups have significantly different means) if the calculated value is larger than the critical value of F with one and $k(n-1)$ degrees of freedom. We demonstrate these ideas by continuing our example.

The five mean biomass values were:

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

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

We have already established that the contrast between the controls and the other four treatments was highly significant (above). Here we develop the theme by assessing the significance of the type of competition treatment. The root pruned plants (r10 and r5) were larger than the shoot pruned plants (n25 and n50), suggesting that below ground competition might be more influential than above ground. It remains to be seen whether these differences are significant by using contrasts. To compare defoliation and root pruning (i.e. a comparison of competition for light with below-ground competition), the contrast coefficients are

```
      control     n25    n50     r10      r5
c_i         0      -1     -1       1       1
```

To calculate a new contrast sum of squares, we need the treatment totals, T,

```
tapply(biomass,clipping,sum)
```

```
control    n25    n50    r10     r5
   2791   3320   3416   3664   3663
```

to which we apply the formula. The controls have zero weight so we ignore them.

$$SSC = \frac{\left[\frac{1}{6}(-1 \times 3320) + (-1 \times 3416) + (1 \times 3664) + (1 \times 3663)\right]^2}{\frac{1}{6}((-1)^2 + (-1)^2 + 1^2 + 1^2)} = \frac{\left(\frac{591}{6}\right)^2}{\frac{4}{6}} = 14553.38.$$

The error variance is 4960.81 (from the Anova table, above), so the F test for this contrast is

$$F = \frac{14553.38}{4960.81} = 2.93367.$$

Notice that this F value is the square of the t value obtained by contrast number 2, above ($1.7128^2 = 2.933684$). We need to test the significance of this by comparing our calculated F value with the critical value with 1 and 25 d.f.. We use qf for this

```
qf(0.95,1,25)
```

```
[1] 4.241699
```

Our calculated value is less than the value in tables, so this contrast was not significant.

Comparison of the Three Kinds of Contrasts

In order to show the differences between treatment, Helmert and sum contrasts, we shall reanalyse this competition experiment.

1. Treatment contrasts

This is the default in R. These are the contrasts you get, unless you explicitly choose otherwise.

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

Here are the contrast coefficients as set under treatment contrasts

```
contrasts(clipping)
```

```
        n25  n50  r10  r5
control   0    0    0    0
n25       1    0    0    0
```

```
n50         0        1         0        0
r10         0        0         1        0
r5          0        0         0        1
```

Notice that the contrasts are **not** orthogonal (the products of the coefficients do not sum to zero).

```
output.treatment < -lm(biomass ~ clipping)
summary(output.treatment)
```

```
Coefficients:
               Estimate   Std. Error   t value   Pr(>|t|)
(Intercept)      465.17        28.75    16.177   9.33e-15    ***
clippingn25       88.17        40.66     2.168    0.03987     *
clippingn50      104.17        40.66     2.562    0.01683     *
clippingr10      145.50        40.66     3.578    0.00145     **
clippingr5       145.33        40.66     3.574    0.00147     **
```

With treatment contrasts, the factor levels are arranged in alphabetical sequence, and the level that comes first in the alphabet is made into the intercept. In our example this is 'control', so we can read off the control mean as 465.17, and the standard error of a mean as 28.75. The remaining four rows are differences between means, and the standard errors are standard errors of differences. Thus, clipping neighbours back to 25 cm increases biomass by 88.17 over the controls and this difference is significant at $p = 0.03987$. And so on. The downside of treatment contrasts is that all the rows appear to be significant despite the fact that rows 2–5 are actually not significantly different from one another, as we saw earlier.

2. Helmert contrasts

This is the default in S-Plus, so beware if you are switching back and forth between the two languages.

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

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

Notice that the contrasts are orthogonal (the products sum to zero) and their coefficients sum to zero, unlike treatment contrasts, above.

```
output.helmert<-lm(biomass~clipping)
summary(output.helmert)
```

```
Coefficients:
            Estimate   Std. Error   t value   Pr(>|t|)
(Intercept) 561.800      12.859    43.688     <2e-16    ***
clipping1    44.083      20.332     2.168     0.0399     *
clipping2    20.028      11.739     1.706     0.1004
clipping3    20.347       8.301     2.451     0.0216     *
clipping4    12.175       6.430     1.894     0.0699     .
```

With Helmert contrasts, the intercept is the overall mean (561.8). The first contrast (on row 2, labelled contrast '1') compares the first mean with the average of the first and second factor levels in alphabetical sequence (control plus n25, see above); its parameter value is the mean of the first two factor levels, minus the mean of the first factor level:

```
(465.16667+553.33333)/2-465.166667
```

```
[1] 44.08332
```

The third row contains the contrast between the third factor level (n50) and the two levels already compared (control and n25); its value is the difference between the average of the first three factor levels and the average of the first two factor levels

```
(465.16667+553.33333+569.333333)/3-(465.166667+553.3333)/2
```

```
[1] 20.02779
```

The fourth row contains the contrast between the fourth factor level (r10) and the three levels already compared (control, n25 and n50); its value is the difference between the average of the first four factor levels and the average of the first three factor levels

```
(465.16667+553.33333+569.333333+610.66667)/4-
(553.3333+ 465.166667+ 569.3333)/3
```

```
[1] 20.34725
```

The fifth and final row contains the contrast between the fifth factor level (r5) and the four levels already compared (control, n25, n50 and r10); its value is the difference between the average of the first five factor levels, and the average of the first four factor levels

```
mean(biomass)-(465.16667+553.33333+569.333333+610.66667)/4
```

```
[1] 12.175
```

So much for the parameter estimates. Now look at the standard errors. We have seen none of these values in any of the analyses we have done to date. The standard error in

row 1 is the standard error of the overall mean, with s^2 taken from the overall Anova table: $\sqrt{\frac{s^2}{k.n}}$

```
sqrt(4961/30)
```

```
[1] 12.85950
```

The standard error in row 2 is a comparison of **a group of two means with a single mean** $(2 \times 1 = 2)$. This is multiplied by the sample size n in the denominator: $\sqrt{\frac{s^2}{2 \times n}}$

```
sqrt(4961/(2*6))
```

```
[1] 20.33265
```

The standard error in row 3 is a comparison of **a group of three means with a group of two means** $(3 \times 2 = 6)$: $\sqrt{\frac{s^2}{6 \times n}}$

```
sqrt(4961/(3*2*6))
```

```
[1] 11.73906
```

The standard error in row 4 is a comparison of **a group of four means with a group of three means** $(4 \times 3 = 12)$: $\sqrt{\frac{s^2}{12 \times n}}$

```
sqrt(4961/(4*3*6))
```

```
[1] 8.30077
```

The standard error in row 5 is a comparison of **a group of five means with a group of four means** $(5 \times 4 = 20)$: $\sqrt{\frac{s^2}{20 \times n}}$

```
sqrt(4961/(5*4*6))
```

```
[1] 6.429749
```

It is true that the parameter estimates and their standard errors are much more difficult to understand in Helmert than in treatment contrasts. However, the advantage of Helmert contrasts is that they give you proper orthogonal contrasts, and hence give a much clearer picture of which factor levels need to be retained in the minimal adequate model. They do not eliminate the need for careful model simplification, however. As we saw earlier, this example requires only two parameters in the minimal adequate model, but Helmert contrasts (above) suggest the need for three (albeit only marginally significant) parameters.

3. Sum contrasts

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

I do not know anyone who uses sum contrasts, so I won't use up space explaining them. If you are interested, see Statistical Computing, Crawley 2002, p. 341.

Aliasing

Aliasing occurs when there is no information available on which to base an estimate of a parameter value. Parameters can be aliased for one of two reasons:

- there are no data in the dataframe from which to estimate the parameter (e.g. missing values, partial designs or correlation amongst the explanatory variables), or
- the model is structured in such a way that the parameter value cannot be estimated (e.g. over specified models with more parameters than necessary).

Intrinsic aliasing occurs when it is due to **the structure of the model**. **Extrinsic aliasing** occurs when it is due to **the nature of the data**.

If we had a factor with four levels (say none, light, medium and heavy use) then we could estimate four means from the data, one for each factor level. But the model looks like this:

$$y = \mu + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4$$

where the x's are dummy variables having the value 0 or 1 (see p. 164). Clearly there is no point in having five parameters in the model if we can estimate only four independent terms. One of the parameters must be intrinsically aliased.

There are innumerable ways of dealing with this, but three equally logical options are:

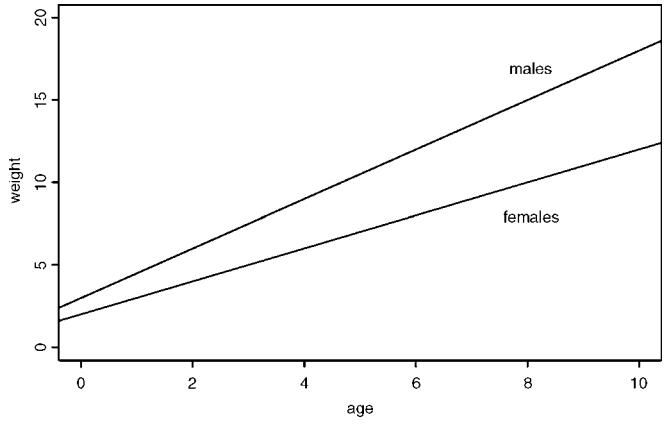
- set the grand mean μ to 0, so that the four β's are the four individual treatment means,
- set the first term β_1 to 0 so that μ is the mean of the first group and the β's are the differences between the first group mean and the other group means,
- set the sum of the β's to 0 so that μ is the grand mean and each β is a departure from the grand mean.

Suppose that in a factorial experiment, all of the animals receiving level 2 of diet (factor A) and level 3 of temperature (factor B) have died accidentally as a result of attack by a fungal pathogen. This particular combination of diet and temperature contributes no data to the response variable, so the interaction term A(2):B(3) cannot be estimated. It is **extrinsically aliased**, and its parameter estimate is set to zero. If one continuous variable is perfectly correlated with another variable that has already been fitted to the data (perhaps because it is a constant multiple of the first variable), then the second term is aliased and adds nothing to the model. Suppose that $x_2 = 0.5x_1$ then fitting a model with $x_1 + x_2$ will lead to x_2 being **intrinsically aliased** and given a zero parameter estimate (see the example of galls on leaves, above). If all the values of a particular explanatory variable are set to zero for a given level of a particular factor, then that level is **intentionally aliased**. This sort of aliasing is a useful programming trick in Ancova when we wish a covariate to be fitted to some levels of a factor but not to others.

Contrasts and the Parameters of Ancova Models

In analysis of covariance, we estimate a slope and an intercept for each level of one or more factors. Suppose we are modelling growth (the response variable) as a function of gender and age. Gender is a factor with two levels (male and female) and age is a continuous measure. The maximal model therefore has four parameters: two slopes (a slope for males and a slope for females) and two intercepts (one for males and one for females) like this:

$$weight_{\text{male}} = a_{\text{male}} + b_{\text{male}} \times age$$
$$weight_{\text{female}} = a_{\text{female}} + b_{\text{female}} \times age$$



The difficulty arises because there are several different ways of expressing the values of the four parameters in the summary.lm table:

- two slopes, and two intercepts (as in the equations, above),
- one slope and one difference between slopes, and one intercept and one difference between intercepts, or
- the overall mean slope and the overall mean intercept, and one difference between slopes and one difference between intercepts.

In the second case (two estimates and two differences) a decision needs to be made about which factor level to associate with the estimate, and which level with the difference (e.g. should males be expressed as the intercept and females as the difference between intercepts, or vice versa)? When the factor levels are unordered (the typical case), then R takes the factor level that comes first in the alphabet as the estimate and the others are expressed as differences. In our example, the parameter estimates would be female, and male parameters would be expressed as differences from the female values,

because 'f' comes before 'm' in the alphabet. This should become clear from an example:

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

```
[1] "weight"  "gender"  "age"
```

First we work out the two regressions separately so that we know the values of the two slopes and the two intercepts:

```
lm(weight[gender=="male"]~age[gender=="male"])
```

```
Coefficients:
  Intercept)  age[ gender == "male"]
    3.115178              1.560808
```

```
lm(weight[gender=="female"]~age[gender==="female"])
```

```
Coefficients:
 (Intercept)    age[ gender == "female"]
   1.966277                  0.9962039
```

So the intercept for males is 3.115 and the intercept for females is 1.966. The difference between the first (female) and second intercepts (male) is therefore

$$3.115 - 1.9266 = +1.1884.$$

Now we can do an overall regression, ignoring gender:

```
lm(weight~age)
```

```
Coefficients:
 (Intercept)         age
   2.540728    1.278506
```

This tells us that the average intercept is 2.541 and the average slope is 1.279.

Next we can carry out an analysis of covariance and compare the output produced by each of the three different contrast options allowed by S-Plus: **Helmert** (the default), **treatment** (the default in R and in Glim) and **sum**.

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

The Ancova estimates separate slopes and intercepts for each gender because we use the asterisk operator:

```
lm(weight~age*gender)
```

```
Coefficients:
 (Intercept)         age         gender   age:gender
   2.540728    1.278506     0.5744508    0.2823018
```

Let's see if we can work out what the four parameter values represent. The first parameter 2.5407 (labelled 'Intercept') is the intercept of the overall regression, ignoring gender (see above). The parameter labelled age (1.2785) is a **slope** because age is our continuous explanatory variable. Again, you will see that it is the slope for the regression of weight against age, ignoring gender. The third parameter labelled **gender** (0.5744) must have something to do with intercepts because **gender** is our categorical variable. If we want to reconstruct the second intercept (for males) we need to add 0.5744 to the overall intercept: $2.5407 + 0.5744 = 3.1151$. To get the intercept for females we need to subtract it $2.5407 - 0.5744 = 1.9663$. The fourth parameter (0.2823) labelled **age:gender** is the difference between the overall mean slope (1.279) and the male slope: $1.2785 + 0.2823 = 1.5608$. To get the slope of weight against age for females we need to subtract the interaction term from the age term: $1.2785 - 0.2823 = 0.9962$.

The advantage of Helmert contrasts is in hypothesis testing, because it is easy to see which terms we need to retain in a simplified model by inspecting their significance levels in the summary.lm table. The disadvantage is that it is hard to reconstruct the slopes and the intercepts from the estimated parameters values (see also p. 220). Let's repeat the analysis using **treatment contrasts** as used by R and by Glim:

```
options(contrasts = c("contr.treatment", "contr.poly"))
lm(weight ~ age*gender)
```

```
Coefficients:
  (Intercept)         age      gender   age:gender
     1.966277   0.9962039    1.148902    0.5646037
```

The Intercept (1.9662) is now the intercept for females (because 'f' comes before 'm' in the alphabet). The **age** parameter (0.9962) is the slope of the graph of weight against age for females. The **gender** parameter (1.1489) is the difference between the (female) intercept and the male intercept ($1.966277 + 1.148902 = 3.1151$). The **age:gender** interaction term is the difference between slopes of the female and male graphs ($0.9962 + 0.5646 = 1.5608$). So with treatment contrasts, the parameters (in order 1 to 4) are an intercept, a slope, a difference between two intercepts, and a difference between two slopes. Many people are more comfortable with this method of presentation than they are with Helmert contrasts.

Finally, we look at the third option which is **sum contrasts**:

```
options(contrasts = c("contr.sum", "contr.poly"))
lm(weight ~ age*gender)
```

```
Coefficients:
  (Intercept)         age       gender   age:gender
     2.540728    1.278506   -0.5744508   -0.2823018
```

The first two terms are the same as those produced by Helmert contrasts: the overall intercept and slope of the graph relating weight to age ignoring gender. The gender parameter (-0.5744) is **sign reversed** compared with the Helmert option: it shows how

to calculate the female (the **first**) intercept from the overall intercept 2.5407 – 0.5746 = 1.9661. The interaction term also has reversed sign – to get the slope for females, add the interaction term to the slope for age: $1.2785 - 0.2823 = 0.9962$.

Multiple Comparisons

The thorny issue of multiple comparisons arises because when we do more than one test we are likely to find 'false positives' at an inflated rate (i.e. by rejecting a true null hypothesis more often than α). The old-fashioned approach was to use Bonferroni's correction; in looking up a value for Student's *t*, you divide your α value by twice the number of comparisons you have done. If the result is still significant then all is well, but it often will not be. Bonferroni's correction is very harsh and will often throw out the baby with the bathwater. An old-fashioned alternative was to use Duncan's Multiple Range Tests (you may have seen these in old statistics books, where lower case letters were written at the head of each bar in a barplot: bars with different letters were significantly different, while bars with the same letter were not significantly different). The modern approach is to use contrasts wherever possible, and where it is essential to do multiple comparisons, then to use the wonderfully named Tukey's Honest Significant Differences (see Statistical Computing, Crawley 2002), and see

```
?TukeyHSD
```