Coding Matrices, Contrast Matrices and Linear Models

Bill Venables

2016-07-09

Contents

1 Introduction

Coding matrices are essentially a convenience for fitting linear models that involve factor predictors. To a large extent, they are arbitrary, and the choice of coding matrix for the factors *should not normally* affect any substantive aspect of the analysis. Nevertheless some users become very confused about this very marginal role of coding (and contrast) matrices.

More pertinently, with Analysis of Variance tables that do not respect the marginality principle, the coding matrices used *do* matter, as they define some of the hypotheses that are implicitly being tested. In this case it is clearly vital to be clear on how the coding matrix works.

My first attempt to explain the working of coding matrices was with the first edition of MASS, (Venables and Ripley, 1992). It was very brief as neither my co-author or I could believe this was a serious issue with most people. It seems we were wrong. With the following three editions of MASS the amount of space given to the issue generally expanded, but space constraints precluded our giving it anything more than a passing discussion.

In this vignette I hope to give a better, more complete and simpler explanation of the concepts. The package it accompanies, `contrastMatrices` offers replacement for the standard coding function in the `stats` package, which I hope will prove a useful, if minor, contribution to the R community.

2 Some theory

Consider a single classification model with p classes. Let the class means be $\mu_1, \mu_2, \ldots, \mu_p$, which under the outer hypothesis are allowed to be all different.

Let $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_p)^\mathsf{T}$ be the vector of class means.

If the observation vector is arranged in class order, the *incidence matrix*, $\mathbf{X}^{n\times p}$ is a binary matrix with the following familiar structure:

$$\mathbf{X}^{n\times p} = \begin{bmatrix} \mathbf{1}_{n_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_{n_2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1}_{n_p} \end{bmatrix} \tag{1}$$

where the class sample sizes are clearly $n_1, n_2, \ldots, n_p$ and $n = n_1 + n_2 + \cdots + n_p$ is the total number of observations.

Under the outer hypothesis[1] the mean vector, $\boldsymbol{\eta}$, for the entire sample can be written, in matrix terms

$$\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\mu} \tag{2}$$

Under the usual null hypothesis the class means are all equal, that is, $\mu_1 = \mu_2 = \cdots = \mu_p = \mu_.$, say. Under this model the mean vector may be written:

$$\boldsymbol{\eta} = \mathbf{1}_n \mu_. \tag{3}$$

Noting that $\mathbf{X}$ is a binary matrix with a singly unity entry in each row, we see that, trivially:

$$\mathbf{1}_n = \mathbf{X}\mathbf{1}_p \tag{4}$$

This will be used shortly.

[1] Sometimes called the *alternative* hypothesis.

2.1 Transformation to new parameters

Let $\mathbf{C}^{p\times p}$ be a non-singular matrix, and rather than using $\boldsymbol{\mu}$ as the parameters for our model, we decide to use an alternative set of parameters defined as:

$$\boldsymbol{\beta} = \mathbf{C}\boldsymbol{\mu} \tag{5}$$

Since $\mathbf{C}$ is non-singular we can write the inverse transformation as

$$\boldsymbol{\mu} = \mathbf{C}^{-1}\boldsymbol{\beta} = \mathbf{B}\boldsymbol{\beta} \tag{6}$$

Where it is convenient to define $\mathbf{B} = \mathbf{C}^{-1}$.

We can write our outer model, (2), in terms of the new parameters as

$$\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\mu} = \mathbf{X}\mathbf{B}\boldsymbol{\beta} \tag{7}$$

So using $\tilde{\mathbf{X}} = \mathbf{X}\mathbf{B}$ as our model matrix (in R terms) the regression coefficients are the new parameters, $\boldsymbol{\beta}$.

Notice that if we choose our transformation $\mathbf{C}$ in such a way to ensure that one of the columns, say the first, of the matrix $\mathbf{B} = \mathbf{C}^{-1}$ is a column of unities, $\mathbf{1}_p$, we can separate out the first column of this new model matrix as an *intercept* column.

Before doing so, it is convenient to label the components of $\boldsymbol{\beta}$ as

$$\boldsymbol{\beta}^\mathsf{T} = (\beta_0, \beta_1, \beta_2, \ldots, \beta_{p-1}) = (\beta_0, \boldsymbol{\beta}_\star^\mathsf{T})$$

that is, the first component is separated out and labelled β_0.

Assuming now that we can arrange for the first column of $\mathbf{B}$ to be a column of unities we can write:

$$\begin{aligned}
\mathbf{X}\mathbf{B}\boldsymbol{\beta} &= \mathbf{X}\begin{bmatrix}\mathbf{1}_p & \mathbf{B}_\star\end{bmatrix}\begin{bmatrix}\beta_0 \\ \boldsymbol{\beta}_\star^{(p-1)\times 1}\end{bmatrix} \\
&= \mathbf{X}\left(\mathbf{1}_p\beta_0 + \mathbf{B}_\star\boldsymbol{\beta}_\star\right) \\
&= \mathbf{X}\mathbf{1}_p\beta_0 + \mathbf{X}\mathbf{B}_\star\boldsymbol{\beta}_\star \\
&= \mathbf{1}_n\beta_0 + \mathbf{X}_\star\boldsymbol{\beta}_\star
\end{aligned} \tag{8}$$

Thus under this assumption we can express the model in terms of a separate *intercept* term, and a set of coefficients $\boldsymbol{\beta}_\star^{(p-1)\times 1}$ with the property that

> The null hypothesis, $\mu_1 = \mu_2 = \cdots = \mu_p$, is true *if and only if* $\boldsymbol{\beta}_\star = \mathbf{0}_{p-1}$, that is, *all the components of* $\boldsymbol{\beta}_\star$ *are zero.*

The $p \times (p-1)$ matrix $\mathbf{B}_\star$ is called a *coding matrix*. The important thing to note about it is that the only restriction we need to place on it is that when a vector of unities is prepended, the resulting matrix $\mathbf{B} = [\mathbf{1}_p\ \mathbf{B}_\star]$ must be non-singular.

In R the familiar `stats` package functions `contr.treatment`, `contr.poly`, `contr.sum` and `contr.helmert` all generate *coding* matrices, and not necessarily contrast matrices as the names might suggest. We look at this in some detail in Section 3 on page 6, *Examples*.

The `contr.*` functions in R are based on the ones of the same name used in S and `S-PLUS`. They were formally described in Chambers and Hastie (1992), although they were in use earlier. (This was the same year as the first edition of MASS appeared as well.)

2.2 Contrast and averaging vectors

We define

An averaging vector as any vector, $\mathbf{c}^{p\times 1}$, whose components add to 1, that is, $\mathbf{c}^\mathsf{T}\mathbf{1}_p = 1$, and

A contrast vector as any *non-zero* vector $\mathbf{c}^{p\times 1}$ whose components add to zero, that is, $\mathbf{c}^\mathsf{T}\mathbf{1}_p = 0$.

Essentially an averaging vector a kind of weighted mean[2] and a contrast vector defines a kind of comparison.

We will call the special case of an averaging vector with equal components:

$$\mathbf{a}^{p\times 1} = \left(\tfrac{1}{p}, \tfrac{1}{p}, \ldots, \tfrac{1}{p}\right)^\mathsf{T} \tag{9}$$

a *simple averaging vector*.

Possibly the simplest contrast has the form:

$$\mathbf{c} = (0, \ldots, 0, 1, 0, \ldots, 0, -1, 0, \ldots, 0)^\mathsf{T}$$

that is, with the only two non-zero components -1 and 1. Such a contrast is sometimes called an *elementary* contrast. If the 1 component is at position i and the -1 at j, then the contrast $\mathbf{c}^\mathsf{T}\boldsymbol{\mu}$ is clearly $\mu_i - \mu_j$, a simple difference.

Two contrasts vectors will be called *equivalent* if one is a scalar multiple of the other, that is, two contrasts $\mathbf{c}$ and $\mathbf{d}$ are equivalent if $\mathbf{c} = \lambda\mathbf{d}$ for some number λ.[3]

2.3 Choosing the transformation

Following on our discussion of transformed parameters, note that the matrix $\mathbf{B}$ has two roles

- It defines the original class means in terms of the new parameters: $\boldsymbol{\mu} = \mathbf{B}\boldsymbol{\beta}$ and
- It modifies the original incidence design matrix, X, into the new model matrix $\mathbf{XB}$.

Recall also that $\boldsymbol{\beta} = \mathbf{C}\boldsymbol{\mu} = \mathbf{B}^{-1}\boldsymbol{\mu}$ so the inverse matrix $\mathbf{B}^{-1}$ determines *how the transformed parameters relate to the original class means*, that is, it determines what the *interpretation* of the new parameters in terms of the primary ones.

Choosing a $\mathbf{B}$ matrix with an initial column of ones is the first desirable feature we want, but we would also like to choose the transformation so that the parameters $\boldsymbol{\beta}$ have a ready interpretation in terms of the class means.

[2] Note, however, that some of the components of an averaging vector may be zero or negative.

[3] If we augment the set of all contrast vectors of p components with the zero vector, $\mathbf{0}_p$, the resulting set is a vector space, $\mathscr{C}$, of dimension $p-1$. The elementary contrasts clearly form a spanning set.

The set of all averaging vectors, $\mathbf{a}$, with p components does not form a vector space, but the difference of any two averaging vectors is either a contrast or zero, that is it is in the contrast space: $\mathbf{c} = \mathbf{a}_1 - \mathbf{a}_2 \in \mathscr{C}$.

Write the rows of $\mathbf{C}$ as $\mathbf{c}_0^\top, \mathbf{c}_1^\top, \ldots, \mathbf{c}_{p-1}^\top$. Then

$$\mathbf{I}_p = \mathbf{CB} = \begin{bmatrix} \mathbf{c}_0^\top \\ \mathbf{c}_1^\top \\ \vdots \\ \mathbf{c}_{p-1}^\top \end{bmatrix} \begin{bmatrix} \mathbf{1}_p \, \mathbf{B}_\star \end{bmatrix} = \begin{bmatrix} \mathbf{c}_0^\top \mathbf{1}_p & \mathbf{c}_0^\top \mathbf{B}_\star \\ \mathbf{c}_1^\top \mathbf{1}_p & \mathbf{c}_1^\top \mathbf{B}_\star \\ \vdots & \vdots \\ \mathbf{c}_0^\top \mathbf{1}_p & \mathbf{c}_{p-1}^\top \mathbf{B}_\star \end{bmatrix} \tag{10}$$

Equating the first columns of both sides gives:

$$\begin{bmatrix} \mathbf{c}_0^\top \mathbf{1}_p \\ \mathbf{c}_1^\top \mathbf{1}_p \\ \vdots \\ \mathbf{c}_0^\top \mathbf{1}_p \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{11}$$

This leads to the important result that if $\mathbf{B} = [\mathbf{1}_p \, \mathbf{B}_\star]$ then

- The first row of $\mathbf{c}_0^\top$ of $\mathbf{C}$ is an *averaging vector* and
- The remaining rows $\mathbf{c}_1^\top, \mathbf{c}_2^\top, \ldots, \mathbf{c}_{p-1}^\top$ will be *contrast vectors*.

It will sometimes be useful to recognize this dichotomy my writing $\mathbf{C}$ in a way that partitions off the first row. We then write:

$$\mathbf{C} = \begin{bmatrix} \mathbf{c}_o^\top \\ \mathbf{C}_\star{}^\top \end{bmatrix} \tag{12}$$

Most importantly, since the argument is reversible, choosing $\mathbf{C}$ as a non-singular matrix in this way, that is as an averaging vector for the first row and a linearly independent set of contrasts vectors as the remaining rows, will ensure that $\mathbf{B}$ has the desired form $\mathbf{B} = [\mathbf{1}_p \, \mathbf{B}_\star]$.

Note that

- How we choose the averaging vector $\mathbf{c}_0$ for the first row will determine the relationship between the intercept coefficient, β_0, and the class means, and
- How we choose the contrasts, $\mathbf{C}_\star$ will determine the interpretation of the regression coefficients $\boldsymbol{\beta}_\star$ in terms of the class means.

Suppose we choose $\mathbf{c}_0$ as a *simple* averaging vector: $\mathbf{c}_0 = \frac{1}{p}\mathbf{1}_p$. The first row of equation (10) now shows that

$$\begin{bmatrix} 1 & \mathbf{0}_{p-1}^\top \end{bmatrix} = \begin{bmatrix} \frac{1}{p}\mathbf{1}_p^\top \mathbf{1}_p & \frac{1}{p}\mathbf{1}_p^\top \mathbf{B}_\star \end{bmatrix} \tag{13}$$

and hence $\mathbf{1}_p^\top \mathbf{B}_\star = \mathbf{0}_{p-1}^\top$, which implies that in this special case the *columns* of the coding matrix, $\mathbf{B}_\star$ are also contrast vectors, (though not necessarily the same contrasts as, or even equivalent to, those in the rows of $\mathbf{C}$).

2.4 Orthogonal contrasts

A set of vectors $\mathbf{c}_1, \ldots, \mathbf{c}_{p-1}$ is called *orthogonal* if $\mathbf{c}_i^\top \mathbf{c}_j = 0$ if $i \neq j$. Suppose now we choose the $\mathbf{C}$ matrix with

- The first row as a simple averaging vector, $\mathbf{c}_0 = \frac{1}{p}\mathbf{1}_p$, and
- The remaining rows, $\mathbf{c}_1, \ldots, \mathbf{c}_{p-1}$ as a set of *orthogonal* contrasts.

A simple averaging vector is orthogonal to every contrast by the definition of contrast, so in this case the rows of the matrix $\mathbf{C}$ are all orthogonal vectors. This implies that $\mathbf{C}^\top\mathbf{C}$ is a diagonal matrix:

$$\mathbf{C}\mathbf{C}^\top = \begin{bmatrix} \frac{1}{p^2}\mathbf{1}_p^\top\mathbf{1}_p & 0 & 0 & \cdots & 0 \\ 0 & \mathbf{c}_1^\top\mathbf{c}_1 & 0 & \cdots & 0 \\ 0 & 0 & \mathbf{c}_2^\top\mathbf{c}_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \mathbf{c}_{p-1}^\top\mathbf{c}_{p-1} \end{bmatrix} = \mathbf{D} \quad \text{say.} \tag{14}$$

Noting that, trivially, $\mathbf{I}_p = \mathbf{C}\mathbf{C}^\top(\mathbf{C}\mathbf{C}^\top)^{-1}$ and since $\mathbf{C}\mathbf{B} = \mathbf{I}_p$ it follows that,

$$\mathbf{B} = \mathbf{C}^\top(\mathbf{C}\mathbf{C}^\top)^{-1} = \mathbf{C}^\top\mathbf{D}^{-1} \tag{15}$$

Furthermore since in this case $\mathbf{D}^{-1}$ is a *diagonal* matrix, it follows that each of the columns of $\mathbf{B}$ is a (strictly positive) scalar multiple of the *corresponding row* of $\mathbf{C}$. So in particular the contrasts in $\mathbf{B}$ and $\mathbf{C}$ are, one for one, equivalent.

We may summarise this by noting that

- The columns of the coding matrix, $\mathbf{B}_\star$, are contrast vectors *if and only if* the averaging vector $\mathbf{c}_0$ is a simple averaging vector, and
- In addition, the corresponding contrast vectors in the columns of $\mathbf{B}$ and in the rows of $\mathbf{C}$ are *equivalent*, if and only if either of them form an orthogonal set.

3 Examples

Note on numerical displays: This section will display a number of patterned matrices and to make the patterns more visible we use the `fractional` package to replace the numbers by vulgar fractions and any zeros by dots.

We also use the package `dplyr` but mostly just for the pipe operator `%>%`, to simplify the appearance of the code.

```
library(dplyr)
library(fractional)
```

To see the effect compare the two displays of the same matrix shown in Figure 1 on the next page.

```
M <- (cbind(diag(4), 0)/7 - cbind(0, diag(4))/3) %>% print
          [,1]       [,2]       [,3]       [,4]       [,5]
[1,] 0.1428571 -0.3333333  0.0000000  0.0000000  0.0000000
[2,] 0.0000000  0.1428571 -0.3333333  0.0000000  0.0000000
[3,] 0.0000000  0.0000000  0.1428571 -0.3333333  0.0000000
[4,] 0.0000000  0.0000000  0.0000000  0.1428571 -0.3333333
M <- (cbind(diag(4), 0)/7 - cbind(0, diag(4))/3) %>% fractional %>% print
     [,1] [,2] [,3] [,4] [,5]
[1,]  1/7 -1/3    .    .    .
[2,]    .  1/7 -1/3    .    .
[3,]    .    .  1/7 -1/3    .
[4,]    .    .    .  1/7 -1/3
```

Figure 1: Effect of using the `fractional` package on displays

3.1 Control versus treatments `contr.treatment` and `code_control`

The default coding matrix for "unordered" factors is given by the `stats` package function `contr.treatment`. We can see what the matrix looks like by an example.

```
levs <- letters[1:5]
Bstar <- contr.treatment(levs) %>% fractional %>% print

  b c d e
a . . . .
b 1 . . .
c . 1 . .
d . . 1 .
e . . . 1
```

To see what the implied regression coefficients mean we look at the $\mathbf{C}$ matrix. The columns of $\mathbf{B}_\star$ are clearly not contrasts, so we expect that the correspond averaging vector will not be a simple averaging vector.

```
B <- cbind(Ave = 1, Bstar) %>% fractional %>% print

  Ave b c d e
a 1   . . . .
b 1   1 . . .
c 1   . 1 . .
d 1   . . 1 .
e 1   . . . 1

C <- solve(B) %>% fractional %>% print

    a  b  c  d  e
Ave  1  .  .  .  .
b   -1  1  .  .  .
c   -1  .  1  .  .
```

```
d   -1   .   .   1   .
e   -1   .   .   .   1
```

Hence the regression coefficients, including the intercept, are

$$\beta_0 = \mu_1, \quad \beta_1 = \mu_2 - \mu_1, \beta_2 = \mu_3 - \mu_1, \ldots, \beta_{p-1} = \mu_p - \mu_1$$

So the contrasts are all *elementary* contrasts of the succeeding class means with the first.

The convenience function, `mean_contrasts`, provides this kind of information more directly:

```
mean_contrasts(contr.treatment(levs))
```

```
    m1 m2 m3 m4 m5
Ave  1  .  .  .  .
b   -1  1  .  .  .
c   -1  .  1  .  .
d   -1  .  .  1  .
e   -1  .  .  .  1
```

The package alternative to `contr.treatment` is `code_control`. It generates a very different coding matrix, but the contrasts are the same. The averaging vector, however is now simple, that is, with equal weights:

```
Bstar <- code_control(levs) %>% fractional %>% print
```

```
  b-a  c-a  d-a  e-a
1 -1/5 -1/5 -1/5 -1/5
2  4/5 -1/5 -1/5 -1/5
3 -1/5  4/5 -1/5 -1/5
4 -1/5 -1/5  4/5 -1/5
5 -1/5 -1/5 -1/5  4/5
```

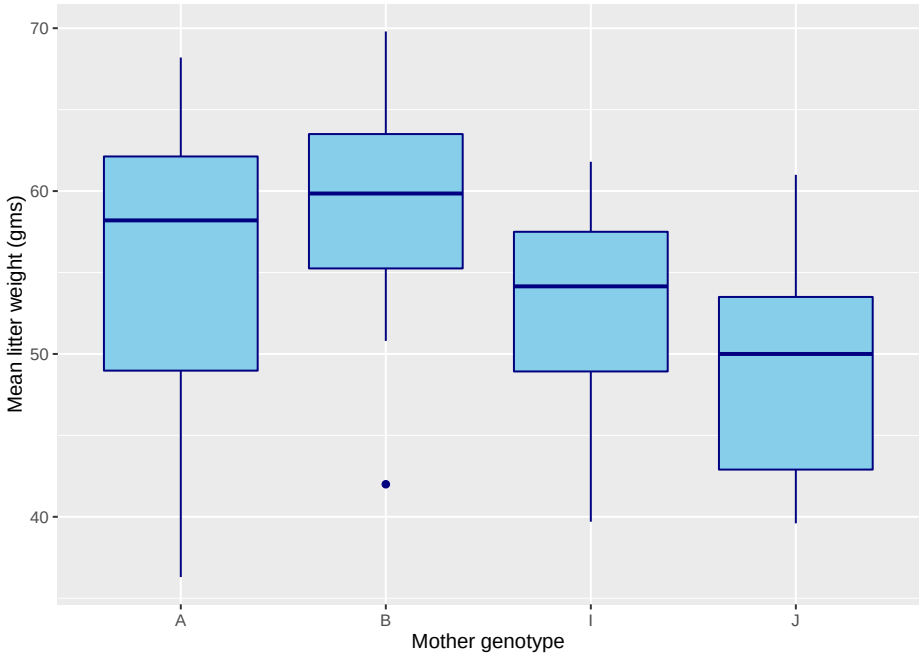
```
mean_contrasts(Bstar)
```

```
    m1  m2  m3  m4  m5
Ave 1/5 1/5 1/5 1/5 1/5
b-a  -1   1   .   .   .
c-a  -1   .   1   .   .
d-a  -1   .   .   1   .
e-a  -1   .   .   .   1
```

Hence the intercept term will now be the average of the class means, but the remaining coefficients will be the same.

We can verify this by a small example.

```
geno <- MASS::genotype
ggplot(geno) + aes(x = Mother, y = Wt) + ylab("Mean litter weight (gms)") +
  geom_boxplot(fill = "sky blue", col = "navy") + xlab("Mother genotype")
```




```
Mmeans <- with(geno, tapply(Wt, Mother, mean))
rbind(Means = Mmeans) %>% booktabs
```

	A	B	I	J
Means	55.40	58.70	53.36	48.68

```
m1 <- aov(Wt ~ Mother, geno)
rbind("From m1:"  = coef(m1),
      "By hand:" = c(Mmeans[1], Mmeans[-1] - Mmeans[1])) %>% booktabs
```

	(Intercept)	MotherB	MotherI	MotherJ
From m1:	55.40	3.30	-2.04	-6.72
By hand:	55.40	3.30	-2.04	-6.72

Changing to the alternative coding function gives a different intercept term, but the same contrasts:

```
m2 <- update(m1, contrasts = list(Mother = "code_control"))
rbind("From m2:" = coef(m2),
      "By hand:" = c(mean(Mmeans), Mmeans[-1] - Mmeans[1])) %>% booktabs
```

	(Intercept)	MotherB-A	MotherI-A	MotherJ-A
From m2:	54.04	3.30	-2.04	-6.72
By hand:	54.04	3.30	-2.04	-6.72

Notice that the intercept term is the average of the class means, which is different from the overall mean in this case, since the class sizes are not equal:

```
rbind("Comparison:" = c(coef(m2)[1], "Grand mean" = mean(geno$Wt))) %>% booktabs
```

	(Intercept)	Grand mean
Comparison:	54.04	53.97

3.1.1 Difference contrasts

A variant of the control versus treatment contrast scheme is what we call "difference" contrasts in Venables and Ripley (2002). Rather than compare each mean with the first, under this scheme each class mean is compared to the one preceding sit, as in a table of differences. The package provides two two versions. The first, `contr.diff` is like `contr.treatment` in that the averaging vector is just the first mean:

```
mean_contrasts(contr.diff(5))
```

```
    m1 m2 m3 m4 m5
Ave  1  .  .  .  .
2-1 -1  1  .  .  .
3-2  . -1  1  .  .
4-3  .  . -1  1  .
5-4  .  .  . -1  1
```

The second, `code_diff` is like `code_control` with a simple average vector:

```
mean_contrasts(code_diff(5))
```

```
    m1  m2  m3  m4  m5
Ave 1/5 1/5 1/5 1/5 1/5
2-1  -1   1   .   .   .
3-2   .  -1   1   .   .
4-3   .   .  -1   1   .
5-4   .   .   .  -1   1
```

We suggest these might be useful for ordered factors as they would show, for example, if the means were monotonic in factor level order.

The default coding function for ordered factors is `contr.poly`, which envisages the levels of the factor as corresponding to equally spaced values of an underlying continuous covariate. This assumption is not always met, however.

3.2 Deviation contrasts, `code_deviation` and `contr.sum`

These two functions essentially do the same job apart from a minor change to labelling of the result.

The pattern in the coding matrix is as follows:

```
Bstar <- code_deviation(levs) %>% fractional %>% print
```

```
  MD1 MD2 MD3 MD4
1  1   .   .   .
```

```
2  .   1   .   .
3  .   .   1   .
4  .   .   .   1
5 -1  -1  -1  -1
```

Note that these columns now are contrast vectors. They are linearly independent but they are not orthogonal. Hence we can conclude that it leads to a simple averaging vector. The contrast pattern is shown as follows:

```
mean_contrasts(Bstar)
```

```
    m1   m2   m3   m4   m5
Ave  1/5  1/5  1/5  1/5  1/5
MD1  4/5 -1/5 -1/5 -1/5 -1/5
MD2 -1/5  4/5 -1/5 -1/5 -1/5
MD3 -1/5 -1/5  4/5 -1/5 -1/5
MD4 -1/5 -1/5 -1/5  4/5 -1/5
```

The general pattern is now clear. We have for the regression coefficients in terms of the class means the following:

$$\beta_0 = \tfrac{1}{p}\sum_{j=1}^{p}\mu_j = \bar{\mu}, \qquad \beta_1 = \mu_1 - \bar{\mu},\ \beta_2 = \mu_2 - \bar{\mu},\ \ldots,\ \beta_{p-1} = \mu_{p-1} - \bar{\mu}$$

So if we add a final coefficient $\beta_p = \mu_p - \bar{\mu}$ to make a symmetric arrangement, the model for the class means can be written as

$$\mu_j = \beta_0 + \beta_j, \quad j = 1,\ldots,p, \quad \text{with} \quad \sum_{j=1}^{p}\beta_j = 0$$

The induced constraint is usually described by saying that the "effects" *sum* to zero, and hence the name `contr.sum`. The alternative description is that the contrasts are the *deviations* of the means from their simple average.

3.3 Helmert contrasts, `contr.helmert` and `code_helmert`

Helmert coding matrices were the original default codings in the early releases of R and indeed in S-PLUS and S beforehand. As hardly anyone understood what they were they were immensely unpopular and were eventually democratically overthrown and replaced by control versus treatment contrasts, which everyone believed that they *did* understand, even if this were not entirely true.

Before outlining the likely original reasoning behind their original adoption, we need to see what they were. The standard function gives a simple pattern of codings:

```
Bstar0 <- contr.helmert(levs) %>% fractional %>% print
```

```
  [,1] [,2] [,3] [,4]
a -1   -1   -1   -1
b  1   -1   -1   -1
c  .    2   -1   -1
```

```
d   .    .    3   -1
e   .    .    .    4
```

Note that the columns of this coding matrix are not just contrasts but *orthogonal* contrasts.

The alternative coding we propose gives a set differently scaled contrast vectors, but *equivalent* to the standard coding set:

```
Bstar1 <- code_helmert(levs) %>% fractional %>% print
```

```
  H2   H3   H4   H5
1 -1/2 -1/3 -1/4 -1/5
2  1/2 -1/3 -1/4 -1/5
3    .  2/3 -1/4 -1/5
4    .    .  3/4 -1/5
5    .    .    .  4/5
```

The standard version leads to the contrast matrix:

```
mean_contrasts(Bstar0)
```

```
    m1    m2    m3    m4    m5
Ave   1/5   1/5   1/5   1/5   1/5
     -1/2   1/2     .     .     .
     -1/6  -1/6   1/3     .     .
    -1/12 -1/12 -1/12   1/4     .
    -1/20 -1/20 -1/20 -1/20   1/5
```

and since the columns of `Bstar0` were orthogonal, the mean contrasts are equivalent to them, that is, essentially the same apart from scaling.

The alternative coding leads to the mean contrast pattern:

```
mean_contrasts(Bstar1)
```

```
    m1   m2   m3   m4   m5
Ave  1/5  1/5  1/5  1/5  1/5
H2    -1    1    .    .    .
H3  -1/2 -1/2    1    .    .
H4  -1/3 -1/3 -1/3    1    .
H5  -1/4 -1/4 -1/4 -1/4    1
```

which is slightly easier to describe. The intercept term is again a simple average of the class means. The regression coefficient $\beta_j, j > 0$ represent a comparison of μ_{j+1} with the average of all the means preceding it in the levels order:

$$\beta_j = \begin{cases} \frac{1}{p}\sum_{j=1}^{p}\mu_j & = \overline{\mu_{1:p}} & \text{for } j = 0 \\ \mu_{j+1} - \frac{1}{j}\sum_{k=1}^{j}\mu_k & = \mu_{j+1} - \overline{\mu_{1:j}} & \text{for } j = 1, 2, \ldots, (p-1) \end{cases} \tag{16}$$

The reason Helmert codings were originally chosen is not clear (to me) although I suspect something like the following reasoning took place.

- Coding matrices with contrast vectors as their columns had the advantage of giving an intercept the simple average of the class means. While this is not important in simple one-factor models, it does become more slightly important with multi-factor models (as we shall see in a later section).
- Matrices with *orthogonal* columns are, in general, numerically very stable even for large cases, and were very quick and easy to invert.
- Coding matrices with *orthogonal contrast* columns provided the user, (should they care to look at them), with a vies of the mean contrasts that result, up to equivalence.
- For standard analysis of variance problems the coding matrix used would not normally influence the inferential outcome, anyway, so interpretability *per se* has low priority.

None of these is particularly convincing, which is probably why they were eventually replaced by treatment versus control contrasts in response to the popular prejudice.

4 The double classification

4.1 Incidence matrices

A double classification is defined by two factors, say `f` and `g`. Suppose they have p and q levels respectively. A model specified in R terms as `~ f*G` is essentially equivalent to a single classification model with pq classes defined by the distinct combinations of `f` and `g`. We can generate the incidence matrix explicitly using a formula such as `~ 0+f:g`. An example is as follows.

```
dat <- data.frame(f = rep(letters[1:3], each = 4),
                  g = rep(LETTERS[1:2], each = 2, length.out = 12))
cbind(model.matrix(~0+f, dat), "----" = 0,
      model.matrix(~0+g, dat), "----" = 0,
      model.matrix(~ 0 + f:g, dat)) %>% fractional
```

```
   fa fb fc ---- gA gB ---- fa:gA fb:gA fc:gA fa:gB fb:gB fc:gB
1   1  .  .    .  1  .    .     1     .     .     .     .     .
2   1  .  .    .  1  .    .     1     .     .     .     .     .
3   1  .  .    .  .  1    .     .     .     .     1     .     .
4   1  .  .    .  .  1    .     .     .     .     1     .     .
5   .  1  .    .  1  .    .     .     1     .     .     .     .
6   .  1  .    .  1  .    .     .     1     .     .     .     .
7   .  1  .    .  .  1    .     .     .     .     .     1     .
8   .  1  .    .  .  1    .     .     .     .     .     1     .
9   .  .  1    .  1  .    .     .     .     1     .     .     .
10  .  .  1    .  1  .    .     .     .     1     .     .     .
11  .  .  1    .  .  1    .     .     .     .     .     .     1
12  .  .  1    .  .  1    .     .     .     .     .     .     1
```

If $\mathbf{X}^f$ and $\mathbf{X}^g$ are the incidence matrices for `f` and `g` respectively, and $\mathbf{X}^{fg}$ the incidence matrix for the subclasses, notice that $\mathbf{X}^{fg}$ can be generated by taking $\mathbf{X}^f$ and multiplying it, componentwise, by each column of $\mathbf{X}^g$ in turn and joining the results as the partitions to form a $n \times pq$ matrix, namely $\mathbf{X}^{fg}$. More formally:

- If $\mathbf{f}^\top$ is the $i-$th row of $\mathbf{X}^f$ and
- If $\mathbf{g}^\top$ is the $i-$th row of $\mathbf{X}^g$,
- The $i-$th row of $\mathbf{X}^{fg}$ is their Kronecker product $\mathbf{g}^\top \otimes \mathbf{f}^\top$. (Note the reversed order.)

It is useful to note that the $\mathbf{X}^f$ and $\mathbf{X}^g$ can be recovered from $\mathbf{X}^{fg}$ by adding selected columns together, as well as the intercept term. The relevant relations are

$$\begin{aligned}\mathbf{X}^f &= \mathbf{X}^{fg}\left(\mathbf{1}_q \otimes \mathbf{I}_p\right)\\ \mathbf{X}^g &= \mathbf{X}^{fg}\left(\mathbf{I}_q \otimes \mathbf{1}_p\right)\\ \mathbf{1}_n &= \mathbf{X}^{fg}\left(\mathbf{1}_q \otimes \mathbf{1}_p\right) = \mathbf{X}^{fg}\mathbf{1}_{pq}\end{aligned}$$

An easily proved mild generalization of these relations, namely:

$$\begin{aligned}\mathbf{X}^f\mathbf{B} &= \mathbf{X}^{fg}\left(\mathbf{1}_q \otimes \mathbf{B}\right)\\ \mathbf{X}^g\mathbf{D} &= \mathbf{X}^{fg}\left(\mathbf{D} \otimes \mathbf{1}_p\right)\end{aligned} \tag{17}$$

(for any multiplicatively coherent matrices $\mathbf{B}$ and $\mathbf{D}$) will be useful later.

4.2 Transformations of the mean

Although the full model for a double classification can be regarded as a single classification, the customary transformations of the mean are in practice restricted to those which respect its two-factor origins.

For simplicity we assume that all cells are filled, so under the full model all cell means are *estimable*. Under this assumption we can write the pq cell means in as a $p \times q$ matrix, $\boldsymbol{\mu}_{\bullet\bullet}$. When we deal with the subclass means as a vector, $\boldsymbol{\mu}$ (got by stacking each column of $\boldsymbol{\mu}_{\bullet\bullet}$ underneath each other—the vec operation), the bullet subscripts will be omitted.

$$\boldsymbol{\mu}_{\bullet\bullet}^{p\times q} = \begin{bmatrix}\mu_{11} & \mu_{12} & \cdots & \mu_{1q}\\ \mu_{21} & \mu_{22} & \cdots & \mu_{2q}\\ \vdots & \vdots & \ddots & \vdots\\ \mu_{p1} & \mu_{p2} & \cdots & \mu_{pq}\end{bmatrix}, \qquad \boldsymbol{\mu}^{pq\times 1} = \operatorname{vec}(\boldsymbol{\mu}_{\bullet\bullet}) \tag{18}$$

Let $\mathbf{C}^f$ and $\mathbf{C}^g$ be the contrast matrices corresponding to the full coding matrices $\mathbf{B}^f$ and $\mathbf{B}^g$ respectively. That is:

$$\begin{aligned}\mathbf{B}^f &= \left[\mathbf{1}_p\, \mathbf{B}^f_\star\right] = \mathbf{C}^{f^{-1}}\\ \mathbf{B}^g &= \left[\mathbf{1}_q\, \mathbf{B}^g_\star\right] = \mathbf{C}^{g^{-1}}\end{aligned} \tag{19}$$

with the inverse relations:

$$\begin{aligned}\mathbf{C}^f &= \begin{bmatrix}\mathbf{c}_0^{f\,\mathsf{T}} \\ \mathbf{C}_\star^{f\,\mathsf{T}}\end{bmatrix} = \mathbf{B}^{f\,-1} \\ \mathbf{C}^g &= \begin{bmatrix}\mathbf{c}_0^{g\,\mathsf{T}} \\ \mathbf{C}_\star^{g\,\mathsf{T}}\end{bmatrix} = \mathbf{B}^{g\,-1}\end{aligned} \tag{20}$$

Linear transformations that respect the two-factor structure are of the form:

$$\boldsymbol{\beta}_{\bullet\bullet} = \mathbf{C}^f \boldsymbol{\mu}_{\bullet\bullet} \mathbf{C}^{g\,\mathsf{T}} \quad \text{or in vector terms} \quad \boldsymbol{\beta} = \left(\mathbf{C}^g \otimes \mathbf{C}^f\right) \boldsymbol{\mu} \tag{21}$$

The inverse relationship is then:

$$\boldsymbol{\mu}_{\bullet\bullet} = \mathbf{B}^f \boldsymbol{\beta}_{\bullet\bullet} \mathbf{B}^{g\,\mathsf{T}} \quad \text{and in vector terms} \quad \boldsymbol{\mu} = \left(\mathbf{B}^g \otimes \mathbf{B}^f\right) \boldsymbol{\beta} \tag{22}$$

First consider the transformations of the mean, Equation 21. We partition the matrix as:

$$\boldsymbol{\beta}_{\bullet\bullet} = \mathbf{C}^f \boldsymbol{\mu}_{\bullet\bullet} \mathbf{C}^{g\,\mathsf{T}} = \begin{bmatrix}\mathbf{c}_0^{f\,\mathsf{T}} \\ \mathbf{C}_\star^{f\,\mathsf{T}}\end{bmatrix} \boldsymbol{\mu}_{\bullet\bullet} \begin{bmatrix}\mathbf{c}_0^g & \mathbf{C}_\star^g\end{bmatrix} = \begin{bmatrix}\mathbf{c}_0^{f\,\mathsf{T}} \boldsymbol{\mu}_{\bullet\bullet} \mathbf{c}_0^g & \mathbf{c}_0^{f\,\mathsf{T}} \boldsymbol{\mu}_{\bullet\bullet} \mathbf{C}_\star^g \\ \mathbf{C}_\star^{f\,\mathsf{T}} \boldsymbol{\mu}_{\bullet\bullet} \mathbf{c}_0^g & \mathbf{C}_\star^{f\,\mathsf{T}} \boldsymbol{\mu}_{\bullet\bullet} \mathbf{C}_\star^g\end{bmatrix} = \begin{bmatrix}\beta_{00} & \boldsymbol{\beta}_{0\star}^{\mathsf{T}} \\ \boldsymbol{\beta}_{\star 0} & \boldsymbol{\beta}_{\star\star}\end{bmatrix} \tag{23}$$

With this notation:

- β_{00} will be the *intercept* coefficient
- $\boldsymbol{\beta}_{\star 0}$ and $\boldsymbol{\beta}_{0\star}$ are called the `f` and `g` *main effects*, respectively, and
- $\boldsymbol{\beta}_{\star\star}$ is the `f`×`g` *interaction*

  It is important to note that the the main effects, $\boldsymbol{\beta}_{0\star}$ and $\boldsymbol{\beta}_{\star 0}$, are defined from the subclass means, $\boldsymbol{\mu}_{\bullet\bullet}$ by
  - *Averaging over the levels of the other factor using the averaging vector for its contrast matrix and*
  - *Taking contrasts of the averages using the contrast vectors for the given factor.*

  The operations commute, so they can be done in either order.

The consequences of this simple result are often overlooked. So, for example,

- if `f` has a coding matrix giving a simple averaging vector, (e.g. using `contr.helmert` or `contr.sum`), the main effect for `g` represents contrasts between the levels of `g` for *a simple average of the means* over all the levels of `f`, but
- if `f` has a coding matrix giving an averaging vector that selects the first mean, (e.g. using `contr.treatment` or our `contr.diff`), the main effect for `g` represents contrasts between the levels of `g` for *the means in just the first* level of `f`.

If the model does not allow for interactions, such as an additive model, `~ f + g`, then the two are identical. However if the model does allow for interactions, such as `~ f*g`, then the two are likely to be different. So testing a main effect *in the presence of interactions involving it* does require some careful consideration of what you are exactly testing, and that in turn depends on the averaging vectors implied by the coding vectors you are using.

4.3 Model matrices

The full double classification mean vector can, by definition, be written as

$$\boldsymbol{\eta} = \mathbf{X}^{fg}\boldsymbol{\mu} = \mathbf{X}^{fg}\left(\mathbf{B}^{g} \otimes \mathbf{B}^{f}\right)\boldsymbol{\beta} = \mathbf{X}^{fg}\left([\mathbf{1}_q\ \mathbf{B}^{g}_{\star}] \otimes [\mathbf{1}_p\ \mathbf{B}^{f}_{\star}]\right)\boldsymbol{\beta}$$

If we re-arrange the components of $\boldsymbol{\beta}$ in the order

$$[\beta_{00},\ \boldsymbol{\beta}_{\star 0},\ \boldsymbol{\beta}_{0\star},\ \boldsymbol{\beta}_{\star\star}]^{\mathsf{T}}$$

the transformed model matrix, $\mathbf{X}^{fg}\left([\mathbf{1}_q\ \mathbf{B}^{g}_{\star}] \otimes [\mathbf{1}_p\ \mathbf{B}^{f}_{\star}]\right)$ can be re-arranged into four partitions, namely:

$$\left[\mathbf{1}_n \quad \mathbf{X}^{f}\mathbf{B}^{f}_{\star} \quad \mathbf{X}^{g}\mathbf{B}^{g}_{\star} \quad \mathbf{X}^{fg}\left(\mathbf{B}^{g}_{\star} \otimes \mathbf{B}^{f}_{\star}\right)\right] = \left[\mathbf{1}_n \quad \mathbf{X}^{f}_{\star} \quad \mathbf{X}^{g}_{\star} \quad \mathbf{X}^{fg}_{\star\star}\right], \qquad \text{say.} \tag{24}$$

The sizes of these partitions are, respectively, $n \times 1$, $n \times (p-1)$, $n \times (q-1)$ and $n \times (p-1)(q-1)$.

Notice that $\mathbf{X}^{fg}_{\star\star}$ can also be obtained from $\mathbf{X}^{f}_{\star}$ and $\mathbf{X}^{g}_{\star}$ by taking $\mathbf{X}^{f}_{\star}$ and multiplying it, componentwise, by each of the columns of $\mathbf{X}^{g}_{\star}$ and arranging the results in a partitioned matrix.

Putting this together we see that, if $\mathbf{P}$ is the permutation matrix that affects this re-ordering:

$$\begin{aligned}
\boldsymbol{\eta} &= \mathbf{X}^{fg}\left([\mathbf{1}_q\ \mathbf{B}^{g}_{\star}] \otimes [\mathbf{1}_p\ \mathbf{B}^{f}_{\star}]\right)\boldsymbol{\beta} \\
&= \mathbf{X}^{fg}\left([\mathbf{1}_q\ \mathbf{B}^{g}_{\star}] \otimes [\mathbf{1}_p\ \mathbf{B}^{f}_{\star}]\right)\mathbf{P}^{\mathsf{T}}\mathbf{P}\boldsymbol{\beta} \\
&= \left[\mathbf{1}_n \quad \mathbf{X}^{f}_{\star} \quad \mathbf{X}^{g}_{\star} \quad \mathbf{X}^{fg}_{\star\star}\right]\begin{bmatrix} \beta_{00} \\ \boldsymbol{\beta}_{\star 0} \\ \boldsymbol{\beta}_{0\star} \\ \boldsymbol{\beta}_{\star\star} \end{bmatrix} \\
\implies \quad \boldsymbol{\eta} &= \mathbf{1}_n\beta_{00} + \mathbf{X}^{f}_{\star}\boldsymbol{\beta}_{\star 0} + \mathbf{X}^{g}_{\star}\boldsymbol{\beta}_{0\star} + \mathbf{X}^{fg}_{\star\star}\boldsymbol{\beta}_{\star\star}
\end{aligned} \tag{25}$$

Compare this with the simpler expression for the single classification in Equation 8 on page 3.

5 Synopsis and higher way extensions

It is important to recognise the essential simplicity of what is going on here.

Setting up the model matrices involves *only the coding matrices* for the factors involved.

- For a single classification, `~ f`, the model matrix is obtained by coding the incidence matrix, and joining it to an intercept term, i.e.

$$\mathbf{M} = \left[\mathbf{1}_n \quad \mathbf{X}^{f}_{\star}\right]$$

- To extend this to a double classification, `~ f*g`, take the coded incidence matrix for `g`, $\mathbf{X}^g_\star$, and multiply it, column by column, with each of the partitions already present, and add them to the single classification. I.e. $\mathbf{1}_n \cdot \mathbf{X}^g_\star \to \mathbf{X}^g_\star$ and $\mathbf{X}^f_\star \cdot \mathbf{X}^g_\star \to \mathbf{X}^{ft}_{\star\star}$ so the model matrix becomes:
$$\mathbf{M} = \begin{bmatrix} \mathbf{1}_n & \mathbf{X}^f_\star & \mathbf{X}^g_\star & \mathbf{X}^{fg}_{\star\star} \end{bmatrix}$$
- For higher way models the process of constructing the model matrix, for the complete model, follows in the same way. Each new factor generates a new coded incidence matrix, $\mathbf{X}_\star$. Multiply this, columnwise, with each of the partitions of the model matrix already there and add the extra partitions to the model matrix. Thus each factor doubles the number of partitions, (or terms). So for a 3-factor model, `~ f*g*h`:
$$\mathbf{M} = \begin{bmatrix} \mathbf{1}_n & \mathbf{X}^f_\star & \mathbf{X}^g_\star & \mathbf{X}^{fg}_{\star\star} & \mathbf{X}^h_\star & \mathbf{X}^{fh}_\star & \mathbf{X}^{gh}_{\star\star} & \mathbf{X}^{fgh}_{\star\star\star} \end{bmatrix}$$
  In practice, of course, the terms would be arranged according to the order of the interactions.

To interpret the regression coefficients resulting from a linear model fitted with such a design, however, requires the *contrast matrices* $\mathbf{C} = \mathbf{B}^{-1}$, of which the first row, $\mathbf{c}_0^\mathsf{T}$, is the all-important *averaging vector*.

- By an interpretation of the regression coefficients $\boldsymbol{\beta}$ we mean relating them to the subclass means, $\boldsymbol{\mu}$, which have a natural, unequivocal interpretation.
- For interpretative purposes, it is helpful to think of the subclass means as arranged in an $n-$way array, $\boldsymbol{\mu}_{\bullet\bullet\ldots\bullet}$.
  - The *intercept coefficient*, $\beta_{00\ldots0}$ is got from the means array by averaging over all dimensions using the averaging vectors for the codings of each factor in turn.
  - The *main effects* are got by averaging with respect to all factors not involved, and taking contrasts with respect to the dimension of the factor itself.
  - In general, an interaction of any order is got by averaging over all dimensions in this way for the factors not involved, and taking contrasts for the factors that are.
- It is also important to note that for an interpretation of the sums of squares in an analysis of variance table, an understanding of what coding matrices are used, particularly if the table is non-standard such as in the case of the egregious "Type III" sums of squares, when testing marginal effects, such as main effects when interactions involving them are present in the model. The hypothesis being tested depends on precisely how the main effect is defined, which in turn depends on the averaging vector for the *other* factor. (See Venables (1998) for a polemical discussion, now somewhat dated.)

5.1 The genotype example, continued

The `genotype` data used in the example in Section 3 on page 6 has a single response, `Wt`, and two factor predictors `Litter` and `Mother`, each with four levels labelled by `A, B, I, J`.

Figure 2 shows the mean profiles for the four Litter genotypes across Mother types.

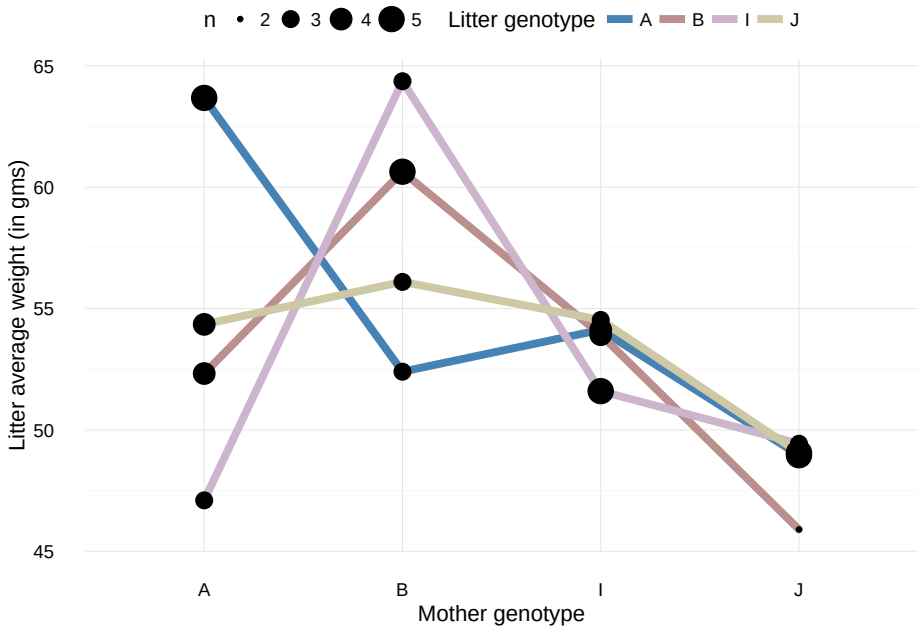


Figure 2: Mean profiles for the `genotype` data

The sequential ANOVA tables show some variation in sums of squares depending on the order in which the terms are introduced, as would have been expected as the subclass numbers are unequal to some degree, but the overall message is clear from either analysis.

```
m2 <- aov(Wt ~ Litter*Mother, geno)
anova(m2) %>% booktabs
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Litter	3	60.16	20.05	0.37	0.7752
Mother	3	775.08	258.36	4.76	0.0057
Litter:Mother	9	824.07	91.56	1.69	0.1201
Residuals	45	2440.82	54.24		

```
anova(update(m2, . ~ Mother*Litter)) %>% booktabs
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Mother	3	771.61	257.20	4.74	0.0059
Litter	3	63.63	21.21	0.39	0.7600
Mother:Litter	9	824.07	91.56	1.69	0.1201
Residuals	45	2440.82	54.24		

To explore the non-standard versions of ANOVA we use John Fox's[4] `car` package which has

[4]John, in his engaging and very pragmatic way provides the technology for generating "Type II" and "Type III" ANOVA tables, but the help information firmly recommends that users *do not* use "Type III" unless they fully understand what they mean—and there is a not very subtle implication that he expects most will not.

an `Anova` function for the purpose.

First consider "Type II", which observes the marginality principle and hence the results do not depend on the choice of coding function:

```
library(car)
Anova(m2, type = "II") %>% booktabs
```

	Sum Sq	Df	F value	Pr(>F)
Litter	63.63	3	0.39	0.7600
Mother	775.08	3	4.76	0.0057
Litter:Mother	824.07	9	1.69	0.1201
Residuals	2440.82	45		

In this case the result is a composite table formed by taking the main effect from the sequential table where it was introduced last. The two-factor interaction is introduced last in both sequential tables and hence is the same for both. It is the only non-marginal term in this case for the full model.

For Type III sums of squares, the protocol is different. The full model is fitted and each term is dropped separately, while leaving the remainder of the model matrix intact. In this case this makes what is actually being tested under the name of a main effect unclear as the definition is critically dependent on the coding matrices being used.

A couple of examples illustrate the point. In the first case the standard `contr.treatment` coding matrices are used, meaning the main effect of one factor is defined by contrasts in the means for the *first level only* of the other factor.

```
Anova(m2, type = "III") %>% booktabs
```

	Sum Sq	Df	F value	Pr(>F)
(Intercept)	20275.71	1	373.81	0.0000
Litter	591.69	3	3.64	0.0197
Mother	582.25	3	3.58	0.0210
Litter:Mother	824.07	9	1.69	0.1201
Residuals	2440.82	45		

Now the main effect of `Litter` seems important, simply because from Figure 2 on the preceding page that looking only at `Mother` level `A`, the `Litter` means do indeed appear to vary quite considerably.

If we change the reference level for `Mother` from the first to the last, however, we would expect from the same diagram the `Litter` would become non-significant. The coding function `contr.SAS`, somewhat ominously, does this switch from first to last levels.

```
Anova(update(m2, contrasts = list(Mother = "contr.SAS")), type = "III") %>%
  booktabs
```

	Sum Sq	Df	F value	Pr(>F)
(Intercept)	11985.41	1	220.97	0.0000
Litter	18.27	3	0.11	0.9525
Mother	582.25	3	3.58	0.0210
Litter:Mother	824.07	9	1.69	0.1201
Residuals	2440.82	45		

And indeed it does so appear. If we vary the `Litter` contrasts also to the same form, the `Mother` term changes as well.

```
Anova(update(m2, contrasts = list(Mother = "contr.SAS",
                                  Litter = "contr.SAS")),
      type = "III") %>% booktabs
```

	Sum Sq	Df	F value	Pr(>F)
(Intercept)	12034.42	1	221.87	0.0000
Litter	18.27	3	0.11	0.9525
Mother	120.78	3	0.74	0.5325
Litter:Mother	824.07	9	1.69	0.1201
Residuals	2440.82	45		

There is really no paradox here. We have chosen coding matrices which produce an averaging vector whose effect is to choose just one level from the set as the 'average'. For `contr.treatment` is is the first level and for `contr.SAS` it is the last. And it so happens that for this data the means for either factor are least variable within the last level of the other factor.

Hence with `contr.treatment` in force both main effects appear to be clearly significant, and with `contr.SAS` both appear to be entirely non-significant. This is simply because they are using different definition of "main effect".

The most commonly advocated resolution of this *essential* arbitrariness is to recommend using coding schemes for which the averaging vector is a simple average, *only*.

Schemes that conform are `contr.helmert`, `contr.sum` and `contr.poly` from the `stats` package and all the `code_*` functions from the present package.

Those that do not conform are `contr.treatment` and `contr.SAS` from the `stats` package and `contr.diff` from the present package.

We finish by checking that different coding schemes, but each having a simple average as the averaging vector, produce the same Type III ANOVA table:

```
Anova(update(m2, contrasts = list(Mother = "contr.sum",
                                  Litter = "contr.poly")),
      type = "III") %>% booktabs
```

	Sum Sq	Df	F value	Pr(>F)
(Intercept)	163782.09	1	3019.56	0.0000
Litter	27.66	3	0.17	0.9161
Mother	671.74	3	4.13	0.0114
Litter:Mother	824.07	9	1.69	0.1201
Residuals	2440.82	45		

```
Anova(update(m2, contrasts = list(Mother = "code_diff",
                                  Litter = "code_helmert")),
      type = "III") %>% booktabs
```

	Sum Sq	Df	F value	Pr(>F)
(Intercept)	163782.09	1	3019.56	0.0000
Litter	27.66	3	0.17	0.9161
Mother	671.74	3	4.13	0.0114
Litter:Mother	824.07	9	1.69	0.1201
Residuals	2440.82	45		

References

Chambers, J. M. and T. J. Hastie (1992). *Statistical Models in* S. London: Chapman & Hall.

Venables, W. N. (1998). Exegeses on linear models. https://www.stats.ox.ac.uk/pub/MASS3/Exegeses.pdf. Paper presented to the **S-PLUS** User's Conference Washington, DC, 8-9th October, 1998.

Venables, W. N. and B. D. Ripley (1992). *Modern Applied Statistics with* S-PLUS. New York: Springer.

Venables, W. N. and B. D. Ripley (2002). *Modern Applied Statistics with* S (Fourth ed.). New York: Springer. ISBN 0-387-95457-0.