

16.7: Different Ways to Specify Contrasts

In the previous section, I showed you a method for converting a factor into a collection of contrasts. In the method I showed you, we specify a set of binary variables, in which define a table like this one:

```
knitr::kable(tibble::tribble(
  ~V1,          ~V2,          ~V3,
  "\"placebo\"",  "0",          "0",
  "\"anxifree\"", "1",          "0",
  "\"joyzepam\"", "0",          "1"
), col.names = c("drug", "druganxifree", "drugjoyzepam"))
```

| drug | druganxifree | drugjoyzepam |
|------------|--------------|--------------|
| "placebo" | 0 | 0 |
| "anxifree" | 1 | 0 |
| "joyzepam" | 0 | 1 |

Each row in the table corresponds to one of the factor levels, and each column corresponds to one of the contrasts. This table, which always has one more row than columns, has a special name: it is called a **contrast matrix**. However, there are lots of different ways to specify a contrast matrix. In this section I discuss a few of the standard contrast matrices that statisticians use, and how you can use them in R. If you're planning to read the section on unbalanced ANOVA later on (Section 16.10) it's worth reading this section carefully. If not, you can get away with skimming it, because the choice of contrasts doesn't matter much for balanced designs.

16.7.1 Treatment contrasts

In the particular kind of contrasts that I've described above, one level of the factor is special, and acts as a kind of "baseline" category (i.e., `placebo` in our example), against which the other two are defined. The name for these kinds of contrast is **treatment contrasts**. The name reflects the fact that these contrasts are quite natural and sensible when one of the categories in your factor really is special because it actually does represent a baseline. That makes sense in our clinical trial example: the `placebo` condition corresponds to the situation where you don't give people any real drugs, and so it's special. The other two conditions are defined in relation to the placebo: in one case you replace the placebo with Anxifree, and in the other case you replace it with Joyzepam.

R comes with a variety of functions that can generate different kinds of contrast matrices. For example, the table shown above is a matrix of treatment contrasts for a factor that has 3 levels. But suppose I want a matrix of treatment contrasts for a factor with 5 levels? The `contr.treatment()` function will do this:

```
contr.treatment( n=5 )
```

```
##    2 3 4 5
## 1 0 0 0 0
## 2 1 0 0 0
## 3 0 1 0 0
## 4 0 0 1 0
## 5 0 0 0 1
```

Notice that, by default, the *first* level of the factor is always treated as the baseline category (i.e., it's the one that has all zeros, and doesn't have an explicit contrast associated with it). In Section 16.6.3 I mentioned that you can use the `relevel()` function to

change which category is the first level of the factor.²⁴³ There's also a special function in R called `contr.SAS()` that generates a treatment contrast matrix in which the *last* category is treated as the baseline:

```
contr.SAS( n=5 )
```

```
##    1 2 3 4
## 1 1 0 0 0
## 2 0 1 0 0
## 3 0 0 1 0
## 4 0 0 0 1
## 5 0 0 0 0
```

However, you can actually select any category you like as the baseline within the `contr.treatment()` function, by specifying the `base` argument in that function. See the help documentation for more details.

16.7.2 Helmert contrasts

Treatment contrasts are useful for a lot of situations, and they're the default in R. However, they make most sense in the situation when there really is a baseline category, and you want to assess all the other groups in relation to that one. In other situations, however, no such baseline category exists, and it may make more sense to compare each group to the mean of the other groups. This is where **Helmert contrasts**, generated by the `contr.helmert()` function, can be useful. The idea behind Helmert contrasts is to compare each group to the mean of the “previous” ones. That is, the first contrast represents the difference between group 2 and group 1, the second contrast represents the difference between group 3 and the mean of groups 1 and 2, and so on. This translates to a contrast matrix that looks like this:

```
contr.helmert( n=5 )
```

```
##    [,1] [,2] [,3] [,4]
## 1   -1   -1   -1   -1
## 2    1   -1   -1   -1
## 3    0    2   -1   -1
## 4    0    0    3   -1
## 5    0    0    0    4
```

One useful thing about Helmert contrasts is that every contrast sums to zero (i.e., all the columns sum to zero). This has the consequence that, when we interpret the ANOVA as a regression, the intercept term corresponds to the grand mean $\mu..$ if we are using Helmert contrasts. Compare this to treatment contrasts, in which the intercept term corresponds to the group mean for the baseline category. This property can be very useful in some situations. It doesn't matter very much if you have a balanced design, which we've been assuming so far, but it will turn out to be important later when we consider unbalanced designs in Section 16.10. In fact, the main reason why I've even bothered to include this section on specifying is that contrasts become important if you want to understand unbalanced ANOVA.

16.7.3 zero contrasts

The third option that I should briefly mention are “sum to zero” contrasts, which are used to construct pairwise comparisons between groups. Specifically, each contrast encodes the difference between one of the groups and a baseline category, which in this case corresponds to the last group:

```
contr.sum( n=5 )
```

```
##      [,1] [,2] [,3] [,4]
## 1      1      0      0      0
## 2      0      1      0      0
## 3      0      0      1      0
## 4      0      0      0      1
## 5     -1     -1     -1     -1
```

Much like Helmert contrasts, we see that each column sums to zero, which means that the intercept term corresponds to the grand mean when ANOVA is treated as a regression model. When interpreting these contrasts, the thing to recognise is that each of these contrasts is a pairwise comparison between group 5 and one of the other four groups. Specifically, contrast 1 corresponds to a “group 1 minus group 5” comparison, contrast 2 corresponds to a “group 2 minus group 5” comparison, and so on.

16.7.4 Viewing and setting the default contrasts in R

Every factor variable in R is associated with a contrast matrix. It has to be, otherwise R wouldn’t be able to run ANOVAs properly! If you don’t specify one explicitly, or R will implicitly specify one for you. Here’s what I mean. When I created the `clin.trial` data, I didn’t specify any contrast matrix for either of the factors. You can see this by using the `attr()` function to print out the “contrasts” attribute of the factors. For example:

```
attr( clin.trial$drug, "contrasts" )
```

```
## NULL
```

The `NULL` output here means that R is telling you that the `drug` factor doesn’t have any attribute called “contrasts” for which it has any data. There is no contrast matrix stored anywhere explicitly for this factor. However, if we now ask R to tell us what contrasts are set up for this factor, it give us this:

```
contrasts( clin.trial$drug )
```

```
##           anxifree joyzepam
## placebo           0           0
## anxifree           1           0
## joyzepam           0           1
```

These are the same treatment contrast that we set up manually in Section 16.6. How did R know to set up treatment contrasts, even though I never actually told it anything about what contrasts I wanted? The answer is that R has a hidden list of default “options” that it looks up to resolve situations like this. You can print out all of the options by typing `options()` at the command prompt, but it’s not a very enlightening read. There are a lot of options, and we’re only interested in contrasts right now. Instead of printing out all of the options, we can ask for just one, like this:

```
options( "contrasts" )
```

```
## $contrasts
##           unordered           ordered
## "contr.treatment" "contr.poly"
```

What this is telling us is that the default contrasts for unordered factors (i.e., nominal scale variables) are treatment contrasts, and the default for ordered factors (i.e., interval scale variables) are “polynomial” contrasts. I don’t discuss ordered factors much in this book, and so I won’t go into what polynomial contrasts are all about. The key thing is that the `options()` function also allows you to reset these defaults (though only for the current session: they’ll revert to the original settings once you close R). Here’s the command:

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

Once we've done this, we can inspect the contrast settings again:

```
options("contrasts")
```

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

Now we see that the default contrasts for unordered factors have changed. So if I now ask R to tell me what contrasts are associated with the `drug` factor, it gives a different answer because I changed the default:

```
contrasts( clin.trial$drug )
```

```
##           [,1] [,2]
## placebo    -1  -1
## anxifree     1  -1
## joyzepam     0   2
```

Those are Helmert contrasts. In general, if you're changing the default settings for something in R, it's a good idea to reset them to their original values once you're done. So let's do that:

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

16.7.5 Setting the contrasts for a single factor

In the previous section, I showed you how to alter the default contrasts. However, suppose that all you really want to do is change the contrasts associated with a single factor, and leave the defaults as they are. To do this, what you need to do is specifically assign the contrast matrix as an "attribute" of the factor. This is easy to do via the `contrasts()` function. For instance, suppose I wanted to use sum to zero contrasts for the `drug` factor, but keep the default treatment contrasts for everything else. I could do that like so:

```
contrasts( clin.trial$drug ) <- contr.sum(3)
```

And if I now inspect the contrasts, I get the following

```
contrasts( clin.trial$drug)
```

```
##           [,1] [,2]
## placebo      1   0
## anxifree      0   1
## joyzepam     -1  -1
```

However, the contrasts for everything else will still be the defaults. You can check that we have actually made a specific change to the factor itself by checking to see if it now has an attribute, using the command `attr(clin.trial$drug, "contrasts")`. This will print out the same output shown above, because the contrast has in fact been attached to the `drug` factor, and does not rely on the defaults. If you want to wipe the attribute and revert the defaults, use a command like this:

```
contrasts( clin.trial$drug ) <- NULL
```

16.7.6 Setting the contrasts for a single analysis

One last way of changing contrasts. You might find yourself wanting to change the contrasts only for one specific analysis. That's allowed too, because the `aov()` and `lm()` functions have a `contrasts` argument that you can use. To change contrasts for one specific analysis, we first set up a list variable that names²⁴⁴ the contrast types that you want to use for each of the factors:

```
my.contrasts <- list( drug = contr.helmert, therapy = contr.helmert )
```

Next, fit the ANOVA model in the usual way, but this time we'll specify the `contrasts` argument:

```
mod <- aov( mood.gain ~ drug*therapy, clin.trial, contrasts = my.contrasts )
```

If you try a command like `summary(aov)` you won't see any difference in the output because the choice of contrasts does not affect the outcome when you have a balanced design (this won't always be true later on). However, if you want to check that it has actually worked, you can inspect the value of `mod$contrasts` :

```
mod$contrasts
```

```
## $drug
##           [,1] [,2]
## placebo    -1   -1
## anxifree     1   -1
## joyzepam     0    2
##
## $therapy
##           [,1]
## no.therapy  -1
## CBT          1
```

As you can see, for the purposes of this one particular ANOVA, R has used Helmert contrasts for both variables. If I had omitted the part of the command that specified the `contrasts` argument, you'd be looking at treatment contrasts here because it would have reverted to whatever values the `contrasts()` function prints out for each of the factors.

This page titled 16.7: Different Ways to Specify Contrasts is shared under a [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/) license and was authored, remixed, and/or curated by Danielle Navarro via [source content](#) that was edited to the style and standards of the LibreTexts platform.