

Chapter 8 of *Data Analysis for Experimental Design*

Rich Gonzalez

September 12, 2009

There are a few ways one can perform contrasts in R. A straightforward one is to define contrasts with the factor definition. This general way will allow many R commands to recognize that you want specific contrasts for a given factor. The contrast attribute is appended to the factor definition, and other R functions can make use of the contrast definition. For the following I'm using the data with 5 groups from Chapter 6 (which is different than what appears in Chapter 7).

The first 4 commands are identical to those in Chapter 6. The fifth command adds the contrasts attribute to the factor. Let's assume I want the four contrasts (1,-1,0,0,0), (1,1,-2,0,0), (1,1,1,3,0), (1,1,1,1,4). The command contrasts allows me to define an entire matrix of contrasts.

The summary command needs to be instructed to split the output by contrasts, otherwise summary will return the omnibus tests. The split argument does this by defining how to split the between-subjects factor (here called "group") into the four contrasts that I defined in the contrast definition.

```
group <- c(rep(1,8), rep(2,8), rep(3,8), rep(4,8), rep(5,8))
data <- c(16, 18, 5, 12, 11, 12, 23, 19, 16, 7, 10, 4, 7, 23, 12, 13, 2,
        10, 9, 13, 11, 9, 13, 9, 5, 8, 8, 11, 1, 9, 5, 9, 7, 11, 12, 9, 14, 19, 16, 24)
group <- factor(group)
contrasts(group) <- cbind(cont1=c(1,-1,0,0,0),cont2=c(1,-2,1,0,0),
                          cont3=c(1,1,1,-3,0),cont4=c(1,1,1,1,-4))
dimnames(contrasts(group)) <- c("cont1", "cont2", "cont3", "cont4")
output <- aov(data~group)
summary(output, split=list(group = list(cont1=1, cont2=2, cont3=3, cont4=4)))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
group	4	314.40	78.60	3.2518	0.02277 *
group: cont1	1	36.00	36.00	1.4894	0.23047
group: cont2	1	65.33	65.33	2.7029	0.10912
group: cont3	1	140.17	140.17	5.7989	0.02144 *
group: cont4	1	72.90	72.90	3.0160	0.09124 .
Residuals	35	846.00	24.17		

The output shows separate tests for each contrast with the labels being of the form “factor name: contrast name”.

Other R commands also take the contrasts defined with the factor. For example the same p-values arise from running the `lm()` command, but the regression command reports in terms of estimated regression coefficients and t-tests, whereas the `aov` reports sum of squares and F tests.

```
output <- lm(data~group)
summary(output)
```

```
Call:
lm(formula = data ~ group)
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-9.500 -2.625  0.250  2.375 11.500
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  11.3000     0.7774  14.536 <2e-16 ***
groupcont1    5.0000     2.4582   2.034  0.0496 *
groupcont2   -2.3333     1.4193  -1.644  0.1091
groupcont3    1.2083     0.5018   2.408  0.0214 *
groupcont4   -0.6750     0.3887  -1.737  0.0912 .
```

```
Residual standard error: 4.916 on 35 degrees of freedom
Multiple R-squared:  0.2709,    Adjusted R-squared:  0.1876
F-statistic: 3.252 on 4 and 35 DF,  p-value: 0.02277
```

Some interesting bells and whistles of contrasts in R. If you only care about running two out of a larger set of contrasts, you just need to specify those two and R will compute an example set for the remaining contrasts.

```
contrasts(group) <- cbind(cont1=c(1,-1,0,0,0),cont2=c(1,-2,1,0,0))
```

```
      cont1 cont2
1      1      1 -0.2581989 -0.2581989
2     -1     -2 -0.2581989 -0.2581989
3      0      1 -0.2581989 -0.2581989
4      0      0  0.8872983 -0.1127017
5      0      0 -0.1127017  0.8872983
```

It isn't always the case that those contrasts computed by R are necessarily meaningful, they just provide one way to round out an entire set of contrasts for a given factor.

Those of you familiar with programming in R can write functions to compute their own contrast tests. A general version would require some careful program to be able to extract the necessary information such as the mean square error term from different objects, such as an aov object and an lm object.