

## An Example of Applying Contrasts in R: the Chickens Example

This analysis uses the “Hand\_010 Weight of Chickens” data set. Recall that this is an RCBD with eight blocks and one treatment factor at three levels.

```
> Chickens <- read.csv(file.choose())
> head(Chickens)
  Weight Block Treatment Treatment.Code
1   3.93     1   Control              1
2   3.78     2   Control              1
3   3.88     3   Control              1
4   3.93     4   Control              1
5   3.84     5   Control              1
6   3.75     6   Control              1
> attach(Chickens)
```

**\*\* The next few lines demonstrate the use of the class() function. Note that**  
**\*\* Weight is classified as a “numeric” variable, Block and Treatment.Code are**  
**\*\* classified as “integer” variables, and Treatment is classified as a**  
**\*\* character variable.**

```
> class(Weight)
[1] "numeric"
> class(Block)
[1] "integer"
> class(Treatment)
[1] "character"
> class(Treatment.Code)
[1] "integer"
```

**\*\* We know that if we wish to use Block and/or Treatment Code in an aov()**  
**\*\* model statement, we need to reclassify them as factors. Note the use of**  
**\*\* the levels() function to identify the levels of Block once it is**  
**\*\* reclassified as a factor.**

```
> Block <- as.factor(Block)
> class(Block)
[1] "factor"
> levels(Block)
[1] "1" "2" "3" "4" "5" "6" "7" "8"
```

**\*\* If we try the levels() function with Treatment, we don't get what we might**  
**\*\* be expecting. Even though the aov() function will consider Treatment as a**  
**\*\* factor, there are advantages to explicitly declaring Treatment as a factor.**  
**\*\* One advantage is that we can use the levels() function to identify the**  
**\*\* levels of the factor. Another advantage is that we can reorder the levels**  
**\*\* of the factor.**

```
> levels(Treatment)
NULL
> Treatment <- as.factor(Treatment)
> levels(Treatment)
[1] "Control" "HighDose" "LowDose"
> Treatment <- factor(Treatment, levels = c("Control", "LowDose", "HighDose"))
> levels(Treatment)
[1] "Control" "LowDose" "HighDose"
```

**\*\* Why reorder the levels of Treatment above? Note that R lists the levels in**  
**\*\* alphabetical order, and this can be confusing when we are trying to set up**  
**\*\* contrasts. Reordering the levels puts them in the “natural” order (in this**  
**\*\* case, of increasing dose) that we likely listed them in the original data**  
**\*\* set and model.**

**\*\* Next we run the basic aov() analysis of an RCBD with one treatment factor.**

```
> Mod1 <- aov(Weight ~ Block + Treatment)
```

```
> anova(Mod1)
```

Analysis of Variance Table

Response: Weight

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Block	7	0.054229	0.007747	1.6259	0.2077402
Treatment	2	0.132358	0.066179	13.8889	0.0004745 ***
Residuals	14	0.066708	0.004765		

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

**\*\* We will now partition the Treatment SS with two orthogonal contrasts. We  
\*\* need to set up a contrast matrix, where the coefficients for each contrast  
\*\* are stored column-wise rather than row-wise. How you make the matrix is up  
\*\* to you, there are many ways to do this. Here's a simple method that helps  
\*\* keep your contrasts straight.**

```
> C1 <- c(1, -0.5, -0.5)
```

```
> C2 <- c(0, 1, -1)
```

```
> Matrix <- cbind(C1, C2)
```

```
> Matrix
```

	C1	C2
[1,]	1.0	0
[2,]	-0.5	1
[3,]	-0.5	-1

**\*\* In the above contrasts, Contrast 1 compares the control (no dose) to the  
\*\* average of the treatments that received a dose. Contrast 2 compares Low  
\*\* Dose to High Dose. You should be able to verify that this is an orthogonal  
\*\* pair of contrasts.**

**\*\* Now we apply the matrix containing the set of contrast coefficients to the  
\*\* Treatment source with the contrasts() function.**

```
> contrasts(Treatment) <- Matrix
```

```
> Mod1 <- aov(Weight ~ Block + Treatment)
```

```
> summary(Mod1)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Block	7	0.05423	0.00775	1.626	0.207740
Treatment	2	0.13236	0.06618	13.889	0.000475 ***
Residuals	14	0.06671	0.00476		

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

**\*\* Nothing looks different so far. We now apply the summary.aov() function to  
\*\* Mod1. Note the rather fussy way that "split" and "list" get used.**

```
> summary.aov(Mod1, split = list(Treatment = list("Control vs Dose" = 1,  
+ "Low Dose vs High Dose" = 2)))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Block	7	0.05423	0.00775	1.626	0.207740
Treatment	2	0.13236	0.06618	13.889	0.000475 ***
Treatment: Control vs Dose	1	0.12813	0.12813	26.891	0.000138 ***
Treatment: Low Dose vs High Dose	1	0.00422	0.00422	0.887	0.362340
Residuals	14	0.06671	0.00476		

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

```
>
```

**\*\* Voilà! Our Treatment source has been partitioned into the two orthogonal  
\*\* contrasts. Each contrast gets tested by taking the appropriate F ratio, in  
\*\* this case by dividing the MS of the contrast by the MSE. The df associated  
\*\* with each contrast F-test is 1, 14. Note that the F-test for first contrast**

**\*\* is significant at the 5% level but the second test is not. How do you  
\*\* interpret these results?**

**\*\* It is important to verify that R is doing what you are expecting it to do  
\*\* when it calculates the SS for each contrast, it is easy to get burned on  
\*\* this. Let's calculate the SS for each contrast directly. First get the  
\*\* Treatment means. We could use model.tables to do this.**

```
> model.tables(Mod1, "means")
```

Tables of means

Grand mean

3.969583

Block

Block

	1	2	3	4	5	6	7	8
	3.960	3.893	3.953	4.007	3.960	3.953	4.070	3.960

Treatment

Treatment

	Control	LowDose	HighDose
	3.866	4.005	4.038

**\*\* the tapply() function will give us the Treatment means with a couple more  
\*\* decimal places; using these will result in less rounding error in the  
\*\* calculation of the SS.**

```
> tapply(Weight, Treatment, mean)
```

	Control	LowDose	HighDose
	3.86625	4.00500	4.03750

**\*\* For Contrast 1 - the first line is the value of the contrast, the next line  
\*\* calculates the SS for Contrast 1:**

```
> 3.86625 - 0.5*(4.005 + 4.0375)
[1] -0.155
> (3.86625 - 0.5*(4.005 + 4.0375))^2/(1.5/8)
[1] 0.1281333
```

**\*\* For Contrast 1 - the first line is the value of the contrast, the next line  
\*\* calculates the SS for Contrast 1:**

```
> (4.005 - 4.0375)
[1] -0.0325
> (4.005 - 4.0375)^2/(2/8)
[1] 0.004225
>
```

**\*\* The contrast SS's are verified!**

**\*\* Here's one more useful command, it is a strange one! It is due to Rose  
Maier and is contained in the document at:**

**[https://rstudio-pubs-static.s3.amazonaws.com/65059\\_586f394d8eb84f84b1baaf56ffb6b47f.html](https://rstudio-pubs-static.s3.amazonaws.com/65059_586f394d8eb84f84b1baaf56ffb6b47f.html)**

**\*\* The beauty of this command is that you can see how R actually applied the  
contrast coefficients to the factor level means.**

```
> attributes(Mod1$qr$qr)$contrasts
```

\$Block

[1] "contr.treatment"

\$Treatment

	C1	C2
Control	1.0	0
LowDose	-0.5	1
HighDose	-0.5	-1