An introductory explanation of contrast coding in R linear models

v.1.2 : December 2014 (minor correction June 2015) : Athanassios Protopapas

Linear regression in R offers an amazing flexibility to targeted analyses estimating the quantities that are relevant to the theoretical questions and their significance. But learning to use this powerful tool is not trivial. It has taken me a long time to reach the very basic and incomplete understanding presented below. I hope others will save some time by going through these steps more quickly, guided to some extent by the following examples.

Specifically, in this document I try to explain the concept of coding factor levels into regression coefficients, to form contrasts that test differences between conditions. In general, linear regression expresses an association between a dependent variable y and one or more independent variables x. This association is continuous in that x and y are both numeric, and is expressed in the well-known kind of formula $y = \beta x + \epsilon$, where ϵ is the residual error, that is, whatever part of y cannot be attributed to x. Because y and x are not necessary scaled around a zero mean, the formula is usually $y = \beta_0 + \beta_1 x + \epsilon$, where β_0 is the "Intercept", that is, the predicted value of y when $x = 0$. You can think of β_0 as the coefficient of the number 1, which is a constant (not a variable), so the complete implied formula is $y = \beta_0 1 + \beta_1 x + \epsilon$.

That's all nice and useful, but it won't work when the independent variables are nominal, that is, "Factor" variables in R. You can't do math with factor levels. So, we need a way to "code" the levels of the factor into numbers, so that they are distinctly represented and can be used for the regression calculations. For example, let's say we have the simple case of a single two-level factor, such as treatment vs. control. We can set $x = 0$ to represent control and $x = 1$ to represent treatment. In other words, a unit of x corresponds to the difference between the two conditions (levels). The Intercept is what happens when $x = 0$, so in this case it is equivalent to the control condition. Using this coding, we now have numbers in the new variable x, instead of factor levels, so we can run the usual regression $y = \beta_0 1 + \beta_1 x + \epsilon$ and estimate the β coefficients and their significance, which is mathematically equivalent to a oneway ANOVA with the corresponding factor.

Alternatively, we could set $x = +1$ to represent treatment and $x = -1$ to represent control. The intercept would correspond to the mean of the two conditions, because zero is in the middle between $+1$ and -1. A unit difference of x would correspond to the difference between one condition and the mean; the difference between the two conditions would be equal to a difference of 2 in x units.

Whatever numbers we choose, the linear regression coefficients and their significance (i.e., whether they differ from zero) will be calculated. But they won't always mean the same thing. So, in this document I go through some examples to demonstrate what the β coefficients mean under a couple of different kinds of coding, so that you can choose what you really need for your analyses and then interpret the results properly.

Preliminaries

Let's create some random data for dependent variable v.
There are two factors: w, with two levels; and h, with three levels.
Include a dummy grouping ("subject") id in variable s.

```
f <- expand.grid(w=c("left","right"),h=c("low","mid","high")) # factor level combinations
d <- data.frame(w=rep(f$w,100),h=rep(f$h,100)) # data frame with factor levels only
d$v <- rnorm(nrow(d),100,15)+ifelse(d$w=="left",-3,3)+ifelse(d$h=="low",-5,ifelse(d$h=="high",5,
0))
# The dv includes random numbers (M=100, SD=15) with moderate effects for w and h built in (6 for
w and 10 for h)
d$s <- as.factor(paste("s",rep(1:25,each=nrow(d)/25),sep="")) # a dummy "subject" variable
str(d)
```

```
## 'data.frame':    600 obs. of  4 variables:
##  $ w: Factor w/ 2 levels "left","right": 1 2 1 2 1 2 1 2 1 2 ...
##  $ h: Factor w/ 3 levels "low","mid","high": 1 1 2 2 3 3 1 1 2 2 ...
##  $ v: num  104.2 104.4 91.4 100.5 89 ...
##  $ s: Factor w/ 25 levels "s1","s10","s11",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
summary(d)
```

```
##      w          h             v                s
##  left :300   low :200   Min.   : 55.93   s1     : 24
##  right:300   mid :200   1st Qu.: 89.87   s10    : 24
##              high:200   Median : 99.73   s11    : 24
##                         Mean   : 99.44   s12    : 24
##                         3rd Qu.:109.06   s13    : 24
##                         Max.   :146.67   s14    : 24
##                                          (Other):456
```

Calculate means; we will need these to understand the values of the beta coefficients.

```
M <- mean(d$v) # grand mean
mw <- with(d,tapply(v,w,mean))          # means of v at the two w levels
mh <- with(d,tapply(v,h,mean))          # means of v at the three h levels
m2 <- with(d,tapply(v,list(w,h),mean)) # means of v at the six w:h combinations
dw <- diff(mw) # difference between levels of w
dh <- diff(mh) # differences between successive levels of h
d2 <- diff(m2) # differences between levels of w at each level of h
```

Analysis with a single factor, treatment-coded

Let's look at a oneway ANOVA first, with two levels (factor w only).

```
d$e2 <- (d$v-M)^2      # SS total
d$mw <- mw[d$w]
d$w2 <- (d$v-d$mw)^2   # SS within

d$b2 <- (d$mw-M)^2     # SS between
ss <- apply(d[,c("e2","w2","b2")],2,sum)
dft <- nrow(d)-1       # df total
dfb <- nlevels(d$w)-1   # df between
dfw <- dft-dfb         # df within
Fv <- (ss["b2"]/dfb) / (ss["w2"]/dfw) #  F = MSB/MSW
p <- 1.0-pf(Fv,dfb,dfw)               #  p value
unname(Fv)
```

```
## [1] 29.46825
```

```
unname(p)
```

```
## [1] 8.272332e-08
```

or, directly:

```
a1 <- aov(v~w,d)
summary(a1)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## w             1   6770    6770   29.47 8.27e-08 ***
## Residuals   598 137390     230
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Through linear regression (same formula):

```
l1 <- lm(v~w,d)
summary(l1) # F statistic on the last line
```

```
##
## Call:
## lm(formula = v ~ w, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.157  -9.751  -0.367   9.951  48.645
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  96.0837     0.8751 109.795  < 2e-16 ***
## wright        6.7183     1.2376   5.428 8.27e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.16 on 598 degrees of freedom
## Multiple R-squared:  0.04696,    Adjusted R-squared:  0.04537
## F-statistic: 29.47 on 1 and 598 DF,  p-value: 8.272e-08
```

```
anova(l1)   # same F statistic with SS/MS details
```

```
## Analysis of Variance Table
##
## Response: v
##            Df Sum Sq Mean Sq F value    Pr(>F)
## w           1   6770  6770.3  29.468 8.272e-08 ***
## Residuals 598 137390   229.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In this case, the effect of w, i.e. the beta for w, is just the difference between left and right

```
coef(l1)["wright"]
```

```
##   wright
## 6.718281
```

```
dw # display for comparison
```

```
##    right
## 6.718281
```

…and the “Intercept” is the mean for left

```
coef(l1)["(Intercept)"]
```

```
## (Intercept)
##    96.08369
```

```
mw["left"]
```

```
##     left
## 96.08369
```

…so, Intercept+wright is the value for right

```
coef(l1)["(Intercept)"]+coef(l1)["wright"]
```

```
## (Intercept)
##     102.802
```

```
mw["right"]
```

```
##   right
## 102.802
```

… and the significance of this effect is the same as a t test between the two levels

```
sqrt(summary(a1)[[1]]["w","F value"]) # F is t squared
```

```
## [1] 5.428466
```

```
summary(a1)[[1]]["w","Pr(>F)"]
```

```
## [1] 8.272332e-08
```

```
#
coef(summary(l1))["wright","t value"]
```

```
## [1] 5.428466
```

```
coef(summary(l1))["wright","Pr(>|t|)"]
```

```
## [1] 8.272332e-08
```

```
#
t.test(d$v[d$w=="right"],d$v[d$w=="left"])
```

```
##
##  Welch Two Sample t-test
##
## data:  d$v[d$w == "right"] and d$v[d$w == "left"]
## t = 5.4285, df = 597.997, p-value = 8.272e-08
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  4.287706 9.148856
## sample estimates:
## mean of x mean of y
## 102.80197  96.08369
```

This is called **treatment coding**

```
contr.treatment(2) # 2 is for the two levels of the factor
```

```
##   2
## 1 0
## 2 1
```

…meaning, that there is one variable in the linear model for the effect of w, and this variable has the value of 0 for one level (“left”, the reference level) and 1 for the other level (“right”, the comparison level). . The effect, beta, is the difference between the two values $(1 - 0)$, i.e., the difference in the dependent variable between the two levels of the factor, so, beta equals the difference in v between w=right and w=left.
The statistical test evaluates the significance of this difference (is $\beta > 0$?), which is equivalent to the reduction in variance by the inclusion of the w factor in the linear model (=the ANOVA calculation).

The model matrix corresponding to this coding is:

```
model.matrix(~w,expand.grid(w=c("left","right")))
```

```
##   (Intercept) wright
## 1           1      0
## 2           1      1
## attr(,"assign")
## [1] 0 1
## attr(,"contrasts")
## attr(,"contrasts")$w
## [1] "contr.treatment"
```

There are two rows (corresponding to the two factor levels) and two columns (two coefficients in the model):

- On the first row, we have Intercept=1 and the model w variable=0 (i.e., “left”), so this corresponds to the value of v for w=“left”

- On the second row, we have Intercept=1 and the model w variable=1 (i.e., “right”), so this coresponds to the sum of *[value for “left”]* plus *[difference between values for “left” and “right”]*

The two terms of the model that are evaluated for significance correspond to the columns:

- One term for the Intercept (i.e., value for “left”)
- One term for the beta coefficient (i.e., difference between “left” and “right”)

Analysis with a two-level factor, deviation-coded

Let’s try the linear regression again, with different coding:

```
l2 <- lm(v~w,d,contrasts=list(w=contr.sum))
summary(l2) # F statistic on the last line, identical to that for l1; check summary(l1)!
```

```
##
## Call:
## lm(formula = v ~ w, data = d, contrasts = list(w = contr.sum))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.157  -9.751  -0.367   9.951  48.645
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  99.4428     0.6188 160.702  < 2e-16 ***
## w1           -3.3591     0.6188  -5.428 8.27e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.16 on 598 degrees of freedom
## Multiple R-squared:  0.04696,    Adjusted R-squared:  0.04537
## F-statistic: 29.47 on 1 and 598 DF,  p-value: 8.272e-08
```

```
anova(l2)    # same as above and as anova(l1)
```

```
## Analysis of Variance Table
##
## Response: v
##            Df Sum Sq Mean Sq F value    Pr(>F)
## w           1   6770  6770.3  29.468 8.272e-08 ***
## Residuals 598 137390   229.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The F and t statistics are identical to those in a1/l1, of course, because we are testing the effect (on residual variance) of including the same distinction i.e., the w factor, distinguishing “left” from “right”.

What about the coefficients?

The “Intercept” is the grand mean

```
coef(l2)["(Intercept)"]
```

```
## (Intercept)
##    99.44283
```

```
M
```

```
## [1] 99.44283
```

The effect of w (called “w1”, i.e., the 1st contrast on w), is half of the difference between left and right, which is equal to the difference of either condition from the grand mean:

```
coef(l2)["w1"]
```

```
##        w1
## -3.359141
```

```
-dw/2 # display for comparison
```

```
##     right
## -3.359141
```

```
mw["left"]-M # "left" corresponds to the positive effect (+1)
```

```
##      left
## -3.359141
```

```
M-mw["right"] # "right" corresponds to the negative effect (-1)
```

```
##     right
## -3.359141
```

…so, left is `Intercept+w1` and right is `Intercept-w1`

```
coef(l2)["(Intercept)"]+coef(l2)["w1"]
```

```
## (Intercept)
##    96.08369
```

```
mw["left"]
```

```
##     left
## 96.08369
```

```
coef(l2)["(Intercept)"]-coef(l2)["w1"]
```

```
## (Intercept)
##     102.802
```

```
mw["right"]
```

```
##   right
## 102.802
```

Why is the effect *half* of the difference? Let's look at the contrast

```
contr.sum(2) # for two levels of the factor
```

```
##   [,1]
## 1    1
## 2   -1
```

There is again one and only one possible contrast between the two levels, but now the variable coding the contrast is -1 for w="right" and +1 for w="left". so the difference between right and left is $1-(-1)=2$. What is this number 2? It is 2 times the effect size, $2\times\beta$, hence beta is half of the difference.

And why is the intercept equal to the grand mean? Well, the intercept is what you get when all regressors (independent variables) are equal to zero. Here our coding variable is -1 for w="right" and +1 for w="left", so it is 0 for their average, which is equally distant from both. So, the intercept is not somehow defined separately; it is implied by the contrast. As soon as you define the variable(s) coding the factor levels, the intercept is simply whatever you happen to get when all variables are equal to zero.

Here is the corresponding model matrix:

```
model.matrix(~w,expand.grid(w=c("left","right")),contrasts=list(w=contr.sum))
```

```
##   (Intercept) w1
## 1           1  1
## 2           1 -1
## attr(,"assign")
## [1] 0 1
## attr(,"contrasts")
## attr(,"contrasts")$w
##       [,1]
## left     1
## right   -1
```

Two columns correspond to the two coefficients and two lines correspond to the two cases examined.

- The first column is the Intercept, now equal to the mean between the two levels.
- The second column is the model effect variable, taking values 1/-1 for the two levels.

So, for "left" we get the first line: *Grand mean+one effect beta*.
For "right" we get the second line: *Grand mean−one effect beta*.

The two terms of the model that are evaluated for significance correspond to the columns:

- One term for the grand mean (testing if the mean of v differs from zero).
- One term for the beta coefficient (testing if the half of the difference between "left" and "right" is different from zero).

Note, in statistical terms, if the difference between left and right differs from zero, then half of the difference also differs from zero, because the standard error is also halved, so the t value remains the same.

If we wanted “left” to be -1 and right to be +1 we could just ask for our own “custom” contrast:

```
l2a <- lm(v~w,d,contrasts=list(w=c(-1,1)))
summary(l2a) # same as l2 except for the sign of the w1 effect
```

```
##
## Call:
## lm(formula = v ~ w, data = d, contrasts = list(w = c(-1, 1)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.157  -9.751  -0.367   9.951  48.645
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  99.4428     0.6188 160.702  < 2e-16 ***
## w1            3.3591     0.6188   5.428 8.27e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.16 on 598 degrees of freedom
## Multiple R-squared:  0.04696,    Adjusted R-squared:  0.04537
## F-statistic: 29.47 on 1 and 598 DF,  p-value: 8.272e-08
```

compare:

```
l2$contrasts
```

```
## $w
##       [,1]
## left     1
## right   -1
```

```
l2a$contrasts
```

```
## $w
##       [,1]
## left    -1
## right    1
```

Analysis with a three-level factor, treatment-coded

```
l3 <- lm(v~h,d)
anova(l3)
```

```
## Analysis of Variance Table
##
## Response: v
##            Df Sum Sq Mean Sq F value    Pr(>F)
## h           2   9853  4926.5  21.898 6.641e-10 ***
## Residuals 597 134307   225.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Of course, the F-statistic of the regression model is identical to the corresponding ANOVA:

```
summary(aov(v~h,d))
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## h             2   9853    4927    21.9 6.64e-10 ***
## Residuals   597 134307     225
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

What about the coefficients?

```
summary(l3)
```

```
##
## Call:
## lm(formula = v ~ h, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.960 -10.301  -0.342   9.788  41.872
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   94.998      1.061  89.571  < 2e-16 ***
## hmid           3.534      1.500   2.356   0.0188 *
## hhigh          9.800      1.500   6.534 1.37e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15 on 597 degrees of freedom
## Multiple R-squared:  0.06835,    Adjusted R-squared:  0.06523
## F-statistic:  21.9 on 2 and 597 DF,  p-value: 6.641e-10
```

The Intercept is the value of v at the “reference level” (which is “low” by default, as given above)

```
coef(l3)["(Intercept)"]
```

```
## (Intercept)
##    94.99818
```

```
mh["low"]
```

```
##      low
## 94.99818
```

The second line is “hmid” and refers to the difference between mid and low (because low is the reference)

```
coef(l3)["hmid"]
```

```
##     hmid
## 3.533851
```

```
dh["mid"]
```

```
##      mid
## 3.533851
```

The third line is “hhigh” refers to the difference between high and low (because low is the reference)

```
coef(l3)["hhigh"]
```

```
##    hhigh
## 9.800084
```

```
mh["high"]-mh["low"]
```

```
##     high
## 9.800084
```

So this model tests for significance (i.e., if they differ from zero) (a) the value at “low”, (b) the difference between low and mid, and (c) the difference between high and low.
The difference between mid and high is not tested, so we don’t know if it differs from zero.

What happened here? First of all, when there are 3 levels, we can have two independent comparisons.
The default coding for two comparisons is:

```
contr.treatment(3)
```

```
##   2 3
## 1 0 0
## 2 1 0
## 3 0 1
```

So there are two dummy variables in the linear model to code the three-level factor.

- One variable codes the level being “mid” (so it is 1 when level is “mid” and 0 otherwise)
- The other variable codes the level being “high” (so it is 1 when level is “high” and 0 otherwise).

When both variables are 0 the level is whatever is left, that is, “low”. This, then, is the intercept.

The corresponding model matrix is:

```
model.matrix(~h,expand.grid(h=c("low","mid","high")))
```

```
##   (Intercept) hmid hhigh
## 1           1    0     0
## 2           1    1     0
## 3           1    0     1
## attr(,"assign")
## [1] 0 1 1
## attr(,"contrasts")
## attr(,"contrasts")$h
## [1] "contr.treatment"
```

The first level (first line) is just the Intercept, corresponding to “low”.
The second level (second line) is the Intercept plus the first effect variable, summing up to “mid” (i.e., “low” plus the difference between “low” and “mid” equals “mid”).
The third level (third line) is the Intercept plus the second effect variable, summing up to “high”.

Three-level factor, deviation-coded

```
l4 <- lm(v~h,d,contrasts=list(h=contr.sum))
anova(l4) # same as l3
```

```
## Analysis of Variance Table
##
## Response: v
##            Df Sum Sq Mean Sq F value    Pr(>F)
## h           2   9853  4926.5  21.898 6.641e-10 ***
## Residuals 597 134307   225.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Of course, the F-statistic of the regression model is identical to the corresponding ANOVA:

```
summary(l4)
```

```
##
## Call:
## lm(formula = v ~ h, data = d, contrasts = list(h = contr.sum))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.960 -10.301  -0.342   9.788  41.872
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  99.4428     0.6123 162.400  < 2e-16 ***
## h1           -4.4446     0.8660  -5.133 3.87e-07 ***
## h2           -0.9108     0.8660  -1.052    0.293
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15 on 597 degrees of freedom
## Multiple R-squared:  0.06835,    Adjusted R-squared:  0.06523
## F-statistic:  21.9 on 2 and 597 DF,  p-value: 6.641e-10
```

Again, there are three effects, three coefficients tested against zero:

The first line is the Intercept, which is the grand mean

```
coef(l4)["(Intercept)"]
```

```
## (Intercept)
##    99.44283
```

```
M
```

```
## [1] 99.44283
```

The second and third line contain contrasts h1 and h2; what are they?

```
coef(l4)["h1"]
```

```
##        h1
## -4.444645
```

```
coef(l4)["h2"]
```

```
##         h2
## -0.9107939
```

Let's examine the contrast vectors for a three-level factor

```
contr.sum(3)
```

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

We see that the first column, corresponding to variable h1, is 1 for “low”, 0 for “mid” and -1 for “high”.
The second column, corresponding to variable h2, is 0 for “low”, 1 for “mid” and -1 for “high”.
So, “low” is coded as: h1=1 & h2=0, “mid” is coded as h1=0 & h2=1, “high” is coded as h1=-1 & h2=-1. Two variables distinguish three levels of the factor. This particular combination of values results in h1 coding the difference between “low” and the Intercept (the grand mean) and h2 coding the difference between “mid” and the Intercept (the grand mean) so that h1+h2 codes for the difference between the Intercept and “high”

Let’s try it out:

```
mh["low"]-M  # equal to coef(l4)["h1"]
```

```
##       low
## -4.444645
```

```
mh["mid"]-M  # equal to coef(l4)["h2"]
```

```
##        mid
## -0.9107939
```

```
M-mh["high"] # equal to coef(l4)["h1"]+coef(l4)["h2"]
```

```
##      high
## -5.355439
```

If that was unclear, look at the contrast vector again:

- The first line is 1 0, so h1 alone must code for the difference between the grand mean and the first level
- The second line is 0 1, so h2 alone must code for the difference between the grand mean and the second level
- The third line is -1 -1, so h1 and h2 together account for the difference between the grand mean and the third level

Let’s examine the model matrix to complete the picture

```
model.matrix(~h,expand.grid(h=c("low","mid","high")),contrasts=list(h=contr.sum))
```

```
##   (Intercept) h1 h2
## 1           1  1  0
## 2           1  0  1
## 3           1 -1 -1
## attr(,"assign")
## [1] 0 1 1
## attr(,"contrasts")
## attr(,"contrasts")$h
##      [,1] [,2]
## low     1    0
## mid     0    1
## high   -1   -1
```

Three effects to consider (to test if they differ from zero): Intercept, h1, h2
Line 1, `1  1  0` : “low” equals intercept (mean) + h1 (difference of low from mean)
Line 2, `1  0  1` : “mid” equals intercept (mean) + h2 (difference of mid from mean)
Line 3, `1 -1 -1` : “high” equals intercept (mean) -(h1+h2) (difference of high from mean)

Two things to remember:

1. Intercept is the grand mean (in deviation coding; contr.sum) or the reference level (in treatment coding; `contr.treatment`).
2. Columns are model variables, lines are factor levels.

Analysis with two factors

OK, now moving on to two-way analyses. If you’re not yet clear about the previous stuff, stop here and go back.

```
l5 <- lm(v~w*h,d)
anova(l5)
```

```
## Analysis of Variance Table
##
## Response: v
##            Df Sum Sq Mean Sq F value    Pr(>F)
## w           1   6770  6770.3 31.5338 3.012e-08 ***
## h           2   9853  4926.5 22.9461 2.517e-10 ***
## w:h         2      5     2.7  0.0124    0.9877
## Residuals 594 127531   214.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Identical to:

```
summary(aov(v~w*h,d))
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## w             1   6770    6770  31.534 3.01e-08 ***
## h             2   9853    4927  22.946 2.52e-10 ***
## w:h           2      5       3   0.012    0.988
## Residuals   594 127531     215
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Regression coefficients:

```
summary(l5)
```

```
##
## Call:
## lm(formula = v ~ w * h, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.622  -9.076  -0.347   9.256  43.269
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   91.5148     1.4653  62.456  < 2e-16 ***
## wright         6.9668     2.0722   3.362 0.000823 ***
## hmid           3.7613     2.0722   1.815 0.070009 .
## hhigh          9.9454     2.0722   4.799 2.02e-06 ***
## wright:hmid   -0.4549     2.9305  -0.155 0.876699
## wright:hhigh  -0.2907     2.9305  -0.099 0.921016
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.65 on 594 degrees of freedom
## Multiple R-squared:  0.1153, Adjusted R-squared:  0.1079
## F-statistic: 15.49 on 5 and 594 DF,  p-value: 2.451e-14
```

The F statistic on the last line here refers to the variance reduction by all model terms together, that is, all sums of squares from the ANOVA table. Add all the SS and df to verify:

```
SSB <- sum(summary(aov(v~w*h,d))[[1]][1:3,"Sum Sq"]) / sum(summary(aov(v~w*h,d))[[1]][1:3,"Df"])
SSW <- summary(aov(v~w*h,d))[[1]]["Residuals","Sum Sq"] / summary(aov(v~w*h,d))[[1]]["Residual
s","Df"]
SSB/SSW # this should equal the reported F statistic
```

```
## [1] 15.49015
```

```
summary(l5)$fstatistic["value"]
```

```
##    value
## 15.49015
```

Back to the `summary(l5)` table, making sense of the coeffients:

Intercept is the value at the reference levels (w=left, h=low)

```
coef(l5)["(Intercept)"]
```

```
## (Intercept)
##    91.51478
```

```
m2["left","low"] # recall, m2 is the array of means for combinations of h and w levels
```

```
## [1] 91.51478
```

The first beta (“wright”) is the difference between right and left (a w contrast) for h=low (reference)

```
coef(l5)["wright"]
```

```
##   wright
## 6.966807
```

```
m2["right","low"]-m2["left","low"]
```

```
## [1] 6.966807
```

The second beta (“hmid”) is the difference between mid and low (an h contrast) for w=left (reference)

```
coef(l5)["hmid"]
```

```
##     hmid
## 3.761292
```

```
m2["left","mid"]-m2["left","low"]
```

```
## [1] 3.761292
```

The third beta (“hhigh”) is the difference between high and low (an h contrast) for w=left (reference)

```
coef(l5)["hhigh"]
```

```
##    hhigh
## 9.945432
```

```
m2["left","high"]-m2["left","low"]
```

```
## [1] 9.945432
```

The fourth beta (“wright:hmid”) is the difference between mid and low (an h contrast) for w=right, over and above the difference between mid and low for w=left

```
coef(l5)["wright:hmid"]
```

```
## wright:hmid
##  -0.4548823
```

```
(m2["right","mid"]-m2["right","low"])-(m2["left","mid"]-m2["left","low"])
```

```
## [1] -0.4548823
```

The fifth beta ("wright:hhigh") is the difference between high and low (an h contrast) for w=right, over and above the difference between high and low for w=left

```
coef(l5)["wright:hhigh"]
```

```
## wright:hhigh
##   -0.2906956
```

```
(m2["right","high"]-m2["right","low"])-(m2["left","high"]-m2["left","low"])
```

```
## [1] -0.2906956
```

In the coding, the effects are cumulative:

- The w effect is whatever difference caused by right/left is not accounted for by the grand mean.
- The h effect is whatever difference caused by high/mid/low is not accounted for by the grand mean.
- The w:h effects is whatever difference caused by combinations of w and h is not accounted for by the indvidual effects of w and of h. So these are *interaction* effects.

If the two effects are completely independent, as in this artificial data set, then there is no systematic variability beyond what is already accounted for by the individual factors, so the interaction terms are not significant.

The ANOVA tests for these effects simultaneously, by examining the reduction in error variance caused by the inclusion of a factor (more accurately, a term in the model).
So, the interaction can be tested by directly comparing a model with to a model without it:

```
l5i <- lm(v~w+h,d) # model without the interaction term, only additive effects of the factors
anova(l5,l5i) # it is no coincidence that this produces the same test as the w:h line in anova(l
5)
```

```
## Analysis of Variance Table
##
## Model 1: v ~ w * h
## Model 2: v ~ w + h
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    594 127531
## 2    596 127537 -2   -5.3063 0.0124 0.9877
```

The model matrix for w*h lists the variables (columns) needed to account for all factor level combinations

```
model.matrix(~w*h,f)
```

```
##   (Intercept) wright hmid hhigh wright:hmid wright:hhigh
## 1           1      0    0     0           0            0
## 2           1      1    0     0           0            0
## 3           1      0    1     0           0            0
## 4           1      1    1     0           1            0
## 5           1      0    0     1           0            0
## 6           1      1    0     1           0            1
## attr(,"assign")
## [1] 0 1 2 2 3 3
## attr(,"contrasts")
## attr(,"contrasts")$w
## [1] "contr.treatment"
##
## attr(,"contrasts")$h
## [1] "contr.treatment"
```

Row 1 is Intercept,
Row 2 is Intercept+*[difference between right and Intercept]* = right (at h = low),
Row 3 is Intercept+*[difference between mid and Intercept]* = mid (at w = left),
…and so on.

In a model like this, we find out if mid is different from low when w=left, and if the difference between mid and low is different when w=left than when w=right. But we **don't** know if there is a significant difference between mid and low for w=right, and we don't know if there is a significant difference between mid and low on average.
For that, we need deviation coding.

Two factors: one treatment-coded, one deviation-coded

Step 1: One variable deviation-coded, one variable treatment-coded

```
l6 <- lm(v~w*h,d,contrasts=list(w=contr.sum))
anova(l6) # identical to anova(l5)
```

```
## Analysis of Variance Table
##
## Response: v
##            Df Sum Sq Mean Sq F value    Pr(>F)
## w           1   6770  6770.3 31.5338 3.012e-08 ***
## h           2   9853  4926.5 22.9461 2.517e-10 ***
## w:h         2      5     2.7  0.0124    0.9877
## Residuals 594 127531   214.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

What individual effects (coefficients) are we evaluating for significance here?

```
summary(l6)
```

```
##
## Call:
## lm(formula = v ~ w * h, data = d, contrasts = list(w = contr.sum))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.622  -9.076  -0.347   9.256  43.269
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  94.9982     1.0361  91.688  < 2e-16 ***
## w1           -3.4834     1.0361  -3.362 0.000823 ***
## hmid          3.5339     1.4653   2.412 0.016178 *
## hhigh         9.8001     1.4653   6.688 5.22e-11 ***
## w1:hmid       0.2274     1.4653   0.155 0.876699
## w1:hhigh      0.1453     1.4653   0.099 0.921016
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.65 on 594 degrees of freedom
## Multiple R-squared:  0.1153, Adjusted R-squared:  0.1079
## F-statistic: 15.49 on 5 and 594 DF,  p-value: 2.451e-14
```

Because we have `contr.sum` for w, this factor contributes to the Intercept at the mean.
Because `contr.treatment` is implied for h, this factor contributes to the Intercept at its reference level (low).

The first coefficient is the value for low on average (i.e., at the mean of left and right)

```
coef(l6)["(Intercept)"]
```

```
## (Intercept)
##    94.99818
```

```
mean(m2[,"low"])
```

```
## [1] 94.99818
```

The second coefficient is half the difference between left and right at h = low

```
coef(l6)["w1"]
```

```
##        w1
## -3.483404
```

```
(m2["left","low"]-m2["right","low"])/2
```

```
## [1] -3.483404
```

Why half? Because the distance in beta units between left and right is the distance between -1 and +1, i.e., 2.

Actually, the second coefficient is the difference between left and the mean of right and left, all at h=low, because

our reference level for this contrast (the intercept) is the mean of the levels and not any individual levels.

```
m2["left","low"]-mean(m2[,"low"])
```

```
## [1] -3.483404
```

The third coefficient is the difference between mid and low on average (i.e., at the mean of left and right)

```
coef(l6)["hmid"]
```

```
##     hmid
## 3.533851
```

```
mean(m2[,"mid"])-mean(m2[,"low"])
```

```
## [1] 3.533851
```

The fourth coefficient is the difference between high and low on average (i.e., at the mean of left and right)

```
coef(l6)["hhigh"]
```

```
##    hhigh
## 9.800084
```

```
mean(m2[,"high"])-mean(m2[,"low"])
```

```
## [1] 9.800084
```

The fifth coefficient is the difference between *[half the difference between left and right at h = mid]* and *[half the difference between left and right at h = low]*

```
coef(l6)["w1:hmid"]
```

```
##   w1:hmid
## 0.2274411
```

```
(m2["left","mid"]-m2["right","mid"])/2-(m2["left","low"]-m2["right","low"])/2
```

```
## [1] 0.2274411
```

The sixth coefficient is the difference between *[half the difference between left and right at h = high]* and *[half the difference between left and right at h = low]*

```
coef(l6)["w1:hhigh"]
```

```
##  w1:hhigh
## 0.1453478
```

```
(m2["left","high"]-m2["right","high"])/2-(m2["left","low"]-m2["right","low"])/2
```

```
## [1] 0.1453478
```

What use are these coefficients? Depends on the theoretical question.

The model.matrix presents the combinations in somewhat more digestible form:

```
cbind(f,model.matrix(~w*h,f,contrasts=list(w=contr.sum)))
```

```
##       w    h (Intercept) w1 hmid hhigh w1:hmid w1:hhigh
## 1  left  low           1  1    0     0       0        0
## 2 right  low           1 -1    0     0       0        0
## 3  left  mid           1  1    1     0       1        0
## 4 right  mid           1 -1    1     0      -1        0
## 5  left high           1  1    0     1       0        1
## 6 right high           1 -1    0     1       0       -1
```

As always, rows are factor level combinations, columns are model terms (evaluated for significance). For example, v at left/low equals `Intercept+w1` ; v at right/mid equals `Intercept-w1+hmid-w1:hmid` ; etc.

Two factors, both deviation-coded

```
l7 <- lm(v~w*h,d,contrasts=list(w=contr.sum,h=contr.sum))
anova(l7) # same as anova(l6)
```

```
## Analysis of Variance Table
##
## Response: v
##            Df Sum Sq Mean Sq F value    Pr(>F)
## w           1   6770  6770.3 31.5338 3.012e-08 ***
## h           2   9853  4926.5 22.9461 2.517e-10 ***
## w:h         2      5     2.7  0.0124    0.9877
## Residuals 594 127531   214.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(l7)
```

```
##
## Call:
## lm(formula = v ~ w * h, data = d, contrasts = list(w = contr.sum,
##     h = contr.sum))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.622  -9.076  -0.347   9.256  43.269
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  99.4428     0.5982 166.239  < 2e-16 ***
## w1           -3.3591     0.5982  -5.615 3.01e-08 ***
## h1           -4.4446     0.8460  -5.254 2.08e-07 ***
## h2           -0.9108     0.8460  -1.077    0.282
## w1:h1        -0.1243     0.8460  -0.147    0.883
## w1:h2         0.1032     0.8460   0.122    0.903
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.65 on 594 degrees of freedom
## Multiple R-squared:  0.1153, Adjusted R-squared:  0.1079
## F-statistic: 15.49 on 5 and 594 DF,  p-value: 2.451e-14
```

Now, what do these estimates refer to?

The first coefficient (Intercept) is the grand mean

```
coef(l7)["(Intercept)"]
```

```
## (Intercept)
##    99.44283
```

```
M
```

```
## [1] 99.44283
```

The second coefficient (w1) is half the difference between left and right, on average (i.e, including low, mid, and high)

```
coef(l7)["w1"]
```

```
##        w1
## -3.359141
```

```
-diff(apply(m2,1,mean))/2 # average over low/mid/high, then compute the difference between left a
nd right
```

```
##     right
## -3.359141
```

The third coefficient (h1) is the difference between low and the grand mean

```
coef(l7)["h1"]
```

```
##        h1
## -4.444645
```

```
mean(m2[,"low"])-M
```

```
## [1] -4.444645
```

This is averaging over left and right or, in other words, “evaluated at the mean” of w.

The fourth coefficient (h2) is the difference between mid and the grand mean (averaging over left and right)

```
coef(l7)["h2"]
```

```
##         h2
## -0.9107939
```

```
mean(m2[,"mid"])-M
```

```
## [1] -0.9107939
```

The fifth coefficient (w1:h1) is the difference between *[the difference left and right at h=low]* and *[the mean difference left and right]* (over all levels of h)

```
coef(l7)["w1:h1"]
```

```
##     w1:h1
## -0.124263
```

```
-(diff(m2[,"low"])/2 -diff(apply(m2,1,mean))/2)
```

```
##     right
## -0.124263
```

The sixth coefficient (w1:h1) is the difference between *[the difference left and right at h=mid]* and *[the mean difference left and right]* (over all levels of h)

```
coef(l7)["w1:h2"]
```

```
##     w1:h2
## 0.1031782
```

```
-(diff(m2[,"mid"])/2 -diff(apply(m2,1,mean))/2)
```

```
##     right
## 0.1031782
```

The model matrix may help understand these:

```
cbind(f,model.matrix(~w*h,f,contrasts=list(w=contr.sum,h=contr.sum)))
```

```
##       w    h (Intercept) w1 h1 h2 w1:h1 w1:h2
## 1  left  low           1  1  1  0     1     0
## 2 right  low           1 -1  1  0    -1     0
## 3  left  mid           1  1  0  1     0     1
## 4 right  mid           1 -1  0  1     0    -1
## 5  left high           1  1 -1 -1    -1    -1
## 6 right high           1 -1 -1 -1     1     1
```

We could also use custom contrast coefficients. For example, we may want the beta for the w contrast to equal the distance between left and right (rather than between each and the grand mean)

```
l8 <- lm(v~w*h,d,contrasts=list(w=c(-0.5,0.5),h=contr.sum))
anova(l8) # same as anova(l6), anova(l7)
```

```
## Analysis of Variance Table
##
## Response: v
##            Df Sum Sq Mean Sq F value    Pr(>F)
## w           1   6770  6770.3 31.5338 3.012e-08 ***
## h           2   9853  4926.5 22.9461 2.517e-10 ***
## w:h         2      5     2.7  0.0124    0.9877
## Residuals 594 127531   214.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(l8)
```

```
##
## Call:
## lm(formula = v ~ w * h, data = d, contrasts = list(w = c(-0.5,
##     0.5), h = contr.sum))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.622  -9.076  -0.347   9.256  43.269
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  99.4428     0.5982 166.239  < 2e-16 ***
## w1            6.7183     1.1964   5.615 3.01e-08 ***
## h1           -4.4446     0.8460  -5.254 2.08e-07 ***
## h2           -0.9108     0.8460  -1.077    0.282
## w1:h1         0.2485     1.6919   0.147    0.883
## w1:h2        -0.2064     1.6919  -0.122    0.903
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.65 on 594 degrees of freedom
## Multiple R-squared:  0.1153, Adjusted R-squared:  0.1079
## F-statistic: 15.49 on 5 and 594 DF,  p-value: 2.451e-14
```

Here, the beta for the w effect is equal to the difference between left and right:

```
coef(l8)["w1"]
```

```
##       w1
## 6.718281
```

```
diff(apply(m2,1,mean))
```

```
##    right
## 6.718281
```

This is the w “main effect”, that is, the effect of w (left vs. right) averaging over levels of h. So, the test for the w1 coefficient is identical to the w effect in the ANOVA:

```
coef(summary(l8))["w1","t value"]^2
```

```
## [1] 31.53383
```

```
summary(aov(v~w*h,d))[[1]]["w ","F value"]
```

```
## [1] 31.53383
```

```
coef(summary(l8))["w1","Pr(>|t|)"]
```

```
## [1] 3.012338e-08
```

```
summary(aov(v~w*h,d))[[1]]["w ","Pr(>F)"]
```

```
## [1] 3.012338e-08
```

This is only true if evaluated at the mean over the levels of the other factor(s) (i.e., h=contr.sum)

There is no “main effect” for h in these lm models, because the three levels of the factor require two variables to be encoded, and these are evaluated separately. To obtain main effects, you will need to apply an ANOVA.

Or, only use experimental designs with two-level factors:

```
l9 <- lm(v~w*h,d,subset=(h!="mid"),contrasts=list(w=c(-0.5,0.5),h=c(-0.5,0.5)))
anova(l9)
```

```
## Analysis of Variance Table
##
## Response: v
##            Df Sum Sq Mean Sq F value    Pr(>F)
## w           1   4653  4653.2 21.0784 5.930e-06 ***
## h           1   9604  9604.2 43.5053 1.356e-10 ***
## w:h         1      2     2.1  0.0096    0.9221
## Residuals 396  87420   220.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(l9)
```

```
##
## Call:
## lm(formula = v ~ w * h, data = d, subset = (h != "mid"), contrasts = list(w = c(-0.5,
##     0.5), h = c(-0.5, 0.5)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.622  -9.392  -0.347   9.306  43.269
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  99.8982     0.7429 134.471  < 2e-16 ***
## w1            6.8215     1.4858   4.591 5.93e-06 ***
## h1            9.8001     1.4858   6.596 1.36e-10 ***
## w1:h1        -0.2907     2.9716  -0.098    0.922
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.86 on 396 degrees of freedom
## Multiple R-squared:  0.1402, Adjusted R-squared:  0.1337
## F-statistic: 21.53 on 3 and 396 DF,  p-value: 6.143e-13
```

Here, we excluded level “mid” from factor h. (That is, we ignored the “mid” data, as if they did not exist, as if factor h

only had two levels. If “mid” data actually exist, then omitting them does not give you a main effect for h, because some of the data relevant for evaluating h are missing.)

Testing the significance of the w1 coefficient is equivalent to testing the significance of factor w:

```
s9 <- summary(l9)
a9 <- summary(aov(v~w*h,d,subset=(h!="mid")))
coef(s9)["w1","t value"]^2
```

```
## [1] 21.07839
```

```
a9[[1]]["w ","F value"]
```

```
## [1] 21.07839
```

```
coef(s9)["w1","Pr(>|t|)"]
```

```
## [1] 5.929727e-06
```

```
a9[[1]]["w ","Pr(>F)"]
```

```
## [1] 5.929727e-06
```

… and testing the significance of the h1 coefficient is equivalent to testing the significance of factor h:

```
coef(s9)["h1","t value"]^2
```

```
## [1] 43.50532
```

```
a9[[1]]["h ","F value"]
```

```
## [1] 43.50532
```

```
coef(s9)["h1","Pr(>|t|)"]
```

```
## [1] 1.356293e-10
```

```
a9[[1]]["h ","Pr(>F)"]
```

```
## [1] 1.356293e-10
```

The Intercept is equal to the grand mean:

```
coef(s9)["(Intercept)","Estimate"]
```

```
## [1] 99.89822
```

```
mean(m2[,-2]) # excluding "mid"
```

```
## [1] 99.89822
```

The regression coefficient for the effect of w is equal to the mean difference between left and right (i.e., averaged over the two levels of h, since level “mid” was excluded from the analysis):

```
coef(s9)["w1","Estimate"]
```

```
## [1] 6.82146
```

```
diff(apply(m2[,-2],1,mean))
```

```
##   right
## 6.82146
```

Similarly, the regression coefficient for the effect of h is equal to the mean difference between high and low (averaged over the two levels of w):

```
coef(s9)["h1","Estimate"]
```

```
## [1] 9.800084
```

```
diff(apply(m2[,-2],2,mean))
```

```
##     high
## 9.800084
```

If we had used `contr.sum` instead of `c(-0.5,0.5)` for the contrasts of the two variables then the coefficient would be equal to *half* the difference between the two levels, as explained above.

What about the interaction effect? The coefficient is equal to the difference between the differences:

```
coef(s9)["w1:h1","Estimate"]
```

```
## [1] -0.2906956
```

```
unname(diff(m2[,3]-m2[,1]))
```

```
## [1] -0.2906956
```

This is *[the difference between left and right at h=high]* minus *[the difference between left and right at h=low]*. If the two differences are the same, it means that difference between left and right does not depend on whether we are low or high; in other words, the effect of w does not depend on h; in other words, there is no interaction. If the effect of one variable depends on the level of the other, this is the definition of an interaction.

This last analysis is, most likely, what you are after. But, in order to undestand how to get it and why, you had to go through all the others above it. Hopefully it was illuminating.

Now you are ready to go on and understand Helmert coding, forward and backward differences, and so on; and to

create your own contrast matrices. See here (http://statsmodels.sourceforge.net/devel/contrasts.html) and here (http://www.ats.ucla.edu/stat/sas/webbooks/reg/chapter5/sasreg5.htm) for ideas and help.

A custom contrast example

For theoretical reasons, you might be interested in the difference between successive levels of h, that is, between low and mid, and between mid and high, and in whether these differences are the same across levels of w (left and right). You might think that the way to code this is by difference contrasts in both w and h, expressing the desired differences. For example:

```
l10 <- lm(v~w*h,d,contrasts=list(w=c(-0.5,0.5),h=matrix(c(-0.5,0.5,0,0,-0.5,0.5),nrow=3)))
anova(l10)
```

```
## Analysis of Variance Table
##
## Response: v
##            Df Sum Sq Mean Sq F value    Pr(>F)
## w           1   6770  6770.3 31.5338 3.012e-08 ***
## h           2   9853  4926.5 22.9461 2.517e-10 ***
## w:h         2      5     2.7  0.0124    0.9877
## Residuals 594 127531   214.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(l10)
```

```
##
## Call:
## lm(formula = v ~ w * h, data = d, contrasts = list(w = c(-0.5,
##     0.5), h = matrix(c(-0.5, 0.5, 0, 0, -0.5, 0.5), nrow = 3)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.622  -9.076  -0.347   9.256  43.269
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 99.44283    0.59819 166.239  < 2e-16 ***
## w1           6.71828    1.19638   5.615 3.01e-08 ***
## h1           8.88929    1.69194   5.254 2.08e-07 ***
## h2          10.71088    1.69194   6.331 4.82e-10 ***
## w1:h1       -0.49705    3.38388  -0.147    0.883
## w1:h2       -0.08434    3.38388  -0.025    0.980
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.65 on 594 degrees of freedom
## Multiple R-squared:  0.1153, Adjusted R-squared:  0.1079
## F-statistic: 15.49 on 5 and 594 DF,  p-value: 2.451e-14
```

```
s10 <- summary(l10)
```

Let us examine the model matrix to understand this coding and the effects expressed in it.

```
cbind(f,model.matrix(~w*h,f,contrasts=list(w=c(-0.5,0.5),h=matrix(c(-0.5,0.5,0,0,-0.5,0.5),nrow=3))))
```

```
##       w    h (Intercept)   w1   h1   h2 w1:h1 w1:h2
## 1  left  low           1 -0.5 -0.5  0.0  0.25  0.00
## 2 right  low           1  0.5 -0.5  0.0 -0.25  0.00
## 3  left  mid           1 -0.5  0.5 -0.5 -0.25  0.25
## 4 right  mid           1  0.5  0.5 -0.5  0.25 -0.25
## 5  left high           1 -0.5  0.0  0.5  0.00 -0.25
## 6 right high           1  0.5  0.0  0.5  0.00  0.25
```

What is going on here? Among them, h1 and h2 code for the differences between successive levels of h. Overall, low is coded as [h1=-0.5, h2=0.0], mid is coded as [h1=0.5, h2=-0.5], and high is coded as [h1=0.0, h2=0.5].

Of course, the Intercept is the grand mean:

```
s10 <- summary(l10)
coef(s10)["(Intercept)","Estimate"]
```

```
## [1] 99.44283
```

```
M
```

```
## [1] 99.44283
```

Variable w1 codes left as -0.5 and right as +0.5, so a significant w1 effect corresponds to the difference between left and right. This works because w1 is symmetric around the intercept.

```
coef(s10)["w1","Estimate"]
```

```
## [1] 6.718281
```

```
diff(apply(m2,1,mean))
```

```
##    right
## 6.718281
```

Variable h1 codes low as -0.5, mid as +0.5, and high as 0.0, but this is not symmetric around the intercept. Moreover, mid is also coded by a nonzero h2, so you cannot get a the difference between low and mid by examining h2 alone. Looking at the model matrix, we see that h1 is the only nonzero variable coding “low”, with the value of -0.5. So, the effect of h1 in fact corresponds to half the difference between low and the grand mean (intercept):

```
coef(s10)["h1","Estimate"]
```

```
## [1] 8.88929
```

```
2*(M-mh["low"])
```

```
##     low
## 8.88929
```

Variable h2 codes low as 0.0, mid as -0.5, and high as +0.5. As you might expect from the previous effect, this one actually corresponds to half the difference between the grand mean and high:

```
coef(s10)["h2","Estimate"]
```

```
## [1] 10.71088
```

```
2*(mh["high"]-M)
```

```
##     high
## 10.71088
```

So, the interactions examine these differences across left and right. Specifically, `w1:h1` corresponds to the difference between *[half the difference between the mean for left and low at left]* and *[half the difference between the mean for right and low at right]*, and `w1:h2` corresponds to the difference between *[half the difference between the mean for left and high at left]* and *[half the difference between the mean for right and high at right]*:

```
coef(s10)["w1:h1","Estimate"]
```

```
## [1] -0.4970519
```

```
2*(mw["right"]-m2["right","low"])-2*(mw["left"]-m2["left","low"])
```

```
##      right
## -0.4970519
```

```
coef(s10)["w1:h2","Estimate"]
```

```
## [1] -0.08433928
```

```
2*(mw["left"]-m2["left","high"])-2*(mw["right"]-m2["right","high"])
```

```
##        left
## -0.08433928
```

This is probably not very useful. Perhaps we could achieve what we need using treatment coding for h, but setting “mid” to be the reference level?

```
d$hm <- relevel(d$h,ref="mid")
l11 <- lm(v~w*hm,d,contrasts=list(w=c(-0.5,0.5)))
anova(l11)
```

```
## Analysis of Variance Table
##
## Response: v
##            Df Sum Sq Mean Sq F value    Pr(>F)
## w           1   6770  6770.3 31.5338 3.012e-08 ***
## hm          2   9853  4926.5 22.9461 2.517e-10 ***
## w:hm        2      5     2.7  0.0124    0.9877
## Residuals 594 127531   214.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(l11)
```

```
##
## Call:
## lm(formula = v ~ w * hm, data = d, contrasts = list(w = c(-0.5,
##     0.5)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.622  -9.076  -0.347   9.256  43.269
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  98.5320     1.0361  95.099  < 2e-16 ***
## w1            6.5119     2.0722   3.143  0.00176 **
## hmlow        -3.5339     1.4653  -2.412  0.01618 *
## hmhigh        6.2662     1.4653   4.277 2.21e-05 ***
## w1:hmlow      0.4549     2.9305   0.155  0.87670
## w1:hmhigh     0.1642     2.9305   0.056  0.95534
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.65 on 594 degrees of freedom
## Multiple R-squared:  0.1153, Adjusted R-squared:  0.1079
## F-statistic: 15.49 on 5 and 594 DF,  p-value: 2.451e-14
```

```
s11 <- summary(l11)
```

The intercept is now the value at “mid”, averaged over left and right.

```
s11 <- summary(l11)
coef(s11)["(Intercept)","Estimate"]
```

```
## [1] 98.53203
```

```
mh["mid"]
```

```
##      mid
## 98.53203
```

This means that w1 is the difference between left and right at mid (not on average).

```
coef(s11)["w1","Estimate"]
```

```
## [1] 6.511925
```

```
diff(m2[,"mid"])
```

```
##    right
## 6.511925
```

…and hmlow/hmhigh are the differences between low and high from mid (on average)

```
coef(s11)["hmlow","Estimate"]
```

```
## [1] -3.533851
```

```
mh["low"]-mh["mid"]
```

```
##       low
## -3.533851
```

```
coef(s11)["hmhigh","Estimate"]
```

```
## [1] 6.266233
```

```
mh["high"]-mh["mid"]
```

```
##     high
## 6.266233
```

The interactions do correspond to the desired differences, that is, whether the differences between successive levels of h are the same across levels of w (left and right).

```
coef(s11)["w1:hmlow","Estimate"]
```

```
## [1] 0.4548823
```

```
(m2["right","low"]-m2["right","mid"])-(m2["left","low"]-m2["left","mid"])
```

```
## [1] 0.4548823
```

```
coef(s11)["w1:hmhigh","Estimate"]
```

```
## [1] 0.1641867
```

```
(m2["right","high"]-m2["right","mid"])-(m2["left","high"]-m2["left","mid"])
```

```
## [1] 0.1641867
```

In other words, we did get the theoretically desired test, at the expense of losing the main effect of w. Recall that w1 now tests for the difference between left and right at mid, not overall. If we wanted to test for the significance of a main effect of w, without the need to obtain the coefficient, we could always use function `anova` on this model:

```
anova(l11)
```

```
## Analysis of Variance Table
##
## Response: v
##            Df Sum Sq Mean Sq F value    Pr(>F)
## w           1   6770  6770.3 31.5338 3.012e-08 ***
## hm          2   9853  4926.5 22.9461 2.517e-10 ***
## w:hm        2      5     2.7  0.0124    0.9877
## Residuals 594 127531   214.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

But if h had more than three levels then we wouldn't even get the effect of interest. Clearly, the solution is much more complicated. Fortunately, it has been solved for us, and it is provided in the MASS package as `contr.sdif`:

```
suppressWarnings(suppressMessages(library(MASS)))
l12 <- lm(v~w*h,d,contrasts=list(w=contr.sdif,h=contr.sdif))
summary(l12)
```

```
##
## Call:
## lm(formula = v ~ w * h, data = d, contrasts = list(w = contr.sdif,
##     h = contr.sdif))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.622  -9.076  -0.347   9.256  43.269
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  99.4428     0.5982 166.239  < 2e-16 ***
## w2-1          6.7183     1.1964   5.615 3.01e-08 ***
## h2-1          3.5339     1.4653   2.412   0.0162 *
## h3-2          6.2662     1.4653   4.277 2.21e-05 ***
## w2-1:h2-1    -0.4549     2.9305  -0.155   0.8767
## w2-1:h3-2     0.1642     2.9305   0.056   0.9553
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.65 on 594 degrees of freedom
## Multiple R-squared:  0.1153, Adjusted R-squared:  0.1079
## F-statistic: 15.49 on 5 and 594 DF,  p-value: 2.451e-14
```

```
s12 <- summary(l12)
```

You may verify that the results are what we wanted:

```
coef(s12)["(Intercept)","Estimate"]
```

```
## [1] 99.44283
```

```
M
```

```
## [1] 99.44283
```

```
coef(s12)["w2-1","Estimate"]
```

```
## [1] 6.718281
```

```
diff(mw)
```

```
##    right
## 6.718281
```

```
coef(s12)["h2-1","Estimate"]
```

```
## [1] 3.533851
```

```
diff(mh[-3])
```

```
##      mid
## 3.533851
```

```
coef(s12)["h3-2","Estimate"]
```

```
## [1] 6.266233
```

```
diff(mh[-1])
```

```
##     high
## 6.266233
```

```
coef(s12)["w2-1:h2-1","Estimate"]
```

```
## [1] -0.4548823
```

```
(m2["left","low"]-m2["left","mid"])-(m2["right","low"]-m2["right","mid"])
```

```
## [1] -0.4548823
```

```
coef(s12)["w2-1:h3-2","Estimate"]
```

```
## [1] 0.1641867
```

```
(m2["right","high"]-m2["right","mid"])-(m2["left","high"]-m2["left","mid"])
```

```
## [1] 0.1641867
```

What does `contr.sdif` do?

```
contr.sdif(3)
```

```
##          2-1        3-2
## 1 -0.6666667 -0.3333333
## 2  0.3333333 -0.3333333
## 3  0.3333333  0.6666667
```

It is not straightforward to understand this. One hint is that you need to think in terms of the reference baseline, namely the intercept: If you want the intercept to be at the mean, then every variable should add up to zero over the levels of the contrast ($\frac{1}{3}+\frac{1}{3}-\frac{2}{3}=0$). Moreover, you need each variable to code a single difference, so that a difference between two particular factor levels must be equal to a difference of 1.0 for this variable. However, each desired difference must be coded by a single variable only, so that it will be completely taken up by it; other variables must have equal values across the levels expressing this difference.

Check it out for higher values (more levels) than 3 to see how it generalizes.

To help you process this idea and try to think about the interactions, here is the model matrix:

```
cbind(f,model.matrix(~w*h,f,contrasts=list(w=c(-0.5,0.5),h=contr.sdif)))
```

```
##       w    h (Intercept)   w1       h2-1       h3-2    w1:h2-1    w1:h3-2
## 1  left  low           1 -0.5 -0.6666667 -0.3333333  0.3333333  0.1666667
## 2 right  low           1  0.5 -0.6666667 -0.3333333 -0.3333333 -0.1666667
## 3  left  mid           1 -0.5  0.3333333 -0.3333333 -0.1666667  0.1666667
## 4 right  mid           1  0.5  0.3333333 -0.3333333  0.1666667 -0.1666667
## 5  left high           1 -0.5  0.3333333  0.6666667 -0.1666667 -0.3333333
## 6 right high           1  0.5  0.3333333  0.6666667  0.1666667  0.3333333
```

Alternatively, you might have wanted to test the *linear* or *quadratic* effect of h, that is, whether levels of h correspond to linearly increasing values of the dependent variable, or to U-shape (increasing-then-decreasing) values; and whether such *polynomial* effects differ across levels of the other factor. For example, is there a linear effect of h, and if yes, is it different for left and right? To examine this, you'd use the `contr.poly` coding option.

```
l13 <- lm(v~w*h,d,contrasts=list(w=contr.sdif,h=contr.poly))
summary(l13)
```

```
##
## Call:
## lm(formula = v ~ w * h, data = d, contrasts = list(w = contr.sdif,
##     h = contr.poly))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.622  -9.076  -0.347   9.256  43.269
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  99.4428     0.5982 166.239  < 2e-16 ***
## w2-1          6.7183     1.1964   5.615 3.01e-08 ***
## h.L           6.9297     1.0361   6.688 5.22e-11 ***
## h.Q           1.1155     1.0361   1.077    0.282
## w2-1:h.L     -0.2056     2.0722  -0.099    0.921
## w2-1:h.Q      0.2527     2.0722   0.122    0.903
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.65 on 594 degrees of freedom
## Multiple R-squared:  0.1153, Adjusted R-squared:  0.1079
## F-statistic: 15.49 on 5 and 594 DF,  p-value: 2.451e-14
```

```
s13 <- summary(l13)
```

Here is the contrast definition, for a three-level factor, and the model matrix:

```
round(contr.poly(3),4)
```

```
##           .L      .Q
## [1,] -0.7071  0.4082
## [2,]  0.0000 -0.8165
## [3,]  0.7071  0.4082
```

```
cbind(f,model.matrix(~w*h,f,contrasts=list(w=contr.sdif,h=contr.poly)))
```

```
##       w    h (Intercept) w2-1           h.L        h.Q      w2-1:h.L
## 1  left  low           1 -0.5 -7.071068e-01  0.4082483  3.535534e-01
## 2 right  low           1  0.5 -7.071068e-01  0.4082483 -3.535534e-01
## 3  left  mid           1 -0.5 -7.850462e-17 -0.8164966  3.925231e-17
## 4 right  mid           1  0.5 -7.850462e-17 -0.8164966 -3.925231e-17
## 5  left high           1 -0.5  7.071068e-01  0.4082483 -3.535534e-01
## 6 right high           1  0.5  7.071068e-01  0.4082483  3.535534e-01
##     w2-1:h.Q
## 1 -0.2041241
## 2  0.2041241
## 3  0.4082483
## 4 -0.4082483
## 5 -0.2041241
## 6  0.2041241
```

The linear effect of h is proportional to the difference between high and low (by a factor of $\sqrt{2}$, to set the diagonal equal to 1 rather than the sides):

```
coef(s13)["h.L","Estimate"]
```

```
## [1] 6.929706
```

```
diff(mh[-2])/sqrt(2)
```

```
##     high
## 6.929706
```

And this linear effect differs between left and right, as indicated by the interaction:

```
coef(s13)["w2-1:h.L","Estimate"]
```

```
## [1] -0.2055528
```

```
diff(m2["right",-2])/sqrt(2)-diff(m2["left",-2])/sqrt(2)
```

```
##       high
## -0.2055528
```

This makes sense only to the extent there is no significant quadratic effect or interaction, because then there is no “straight line” to speak of its slope.

Other kinds of models

Appication of the above discussion is not limited to lm. The same ideas can be used to construct and interpret mixed-effects and generalized models in exactly the same way.

For example, compare this to `s12`:

```
suppressWarnings(suppressMessages(library(lme4)))
l14 <- lmer(v~w*h+(1|s),d,contrasts=list(w=contr.sdif,h=contr.sdif))
print(summary(l14),corr=F)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: v ~ w * h + (1 | s)
##    Data: d
##
## REML criterion at convergence: 4902.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.70407 -0.61943 -0.02369  0.63170  2.95295
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  s        (Intercept)   0.0     0.00
##  Residual             214.7    14.65
## Number of obs: 600, groups:  s, 25
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  99.4428     0.5982  166.24
## w2-1          6.7183     1.1964    5.62
## h2-1          3.5339     1.4653    2.41
## h3-2          6.2662     1.4653    4.28
## w2-1:h2-1    -0.4549     2.9305   -0.16
## w2-1:h3-2     0.1642     2.9305    0.06
```

Send corrections, comments, and suggestions to protopap[at]gmail[dot]com

Revision of 1 June 2015