# Two way factorials

*Philip Dixon*

*9/25/2020*

### Analysis of data from a two way factorial using contrasts and effects models

The example is the French nature/nurture IQ study. The data are in adoptiveIQ.xlsx. Background on the study is in the background .docx file.

### Reading the data and setting up factors

We will use read_excel() in the readxl library to read the .xlsx file. This is not the only way to read .xlsx files, but this function has the most options (different worksheets, subsets of a worksheet). We will also use emmeans functions for all the "after the ANOVA" analyses.

```
library(readxl)
library(emmeans)
```

As usual, we need to create factor versions of categorical variables.

```
adopt <- read_excel('adoptiveIQ.xlsx')
names(adopt)
```

```
## [1] "IQ"         "Adoptive"   "Biological"
```

```
adopt$adoptive.f <- as.factor(adopt$Adoptive)
adopt$biological.f <- as.factor(adopt$Biological)
```

If we want to use contrasts, the models need a variable that indicates each unique cell (combination of factor levels). This can be created various ways. I concatenate the information in each factor. Later, I show how to get R to do this "on the fly" inside the model.

paste() is the function that concatenates character strings. paste() is pretty smart so you don't need to care about what sort of variable is being used. Numbers are converted to character strings; factors are converted back to their levels. The default separator is a space. You can change that with sep= a character string in quotes.

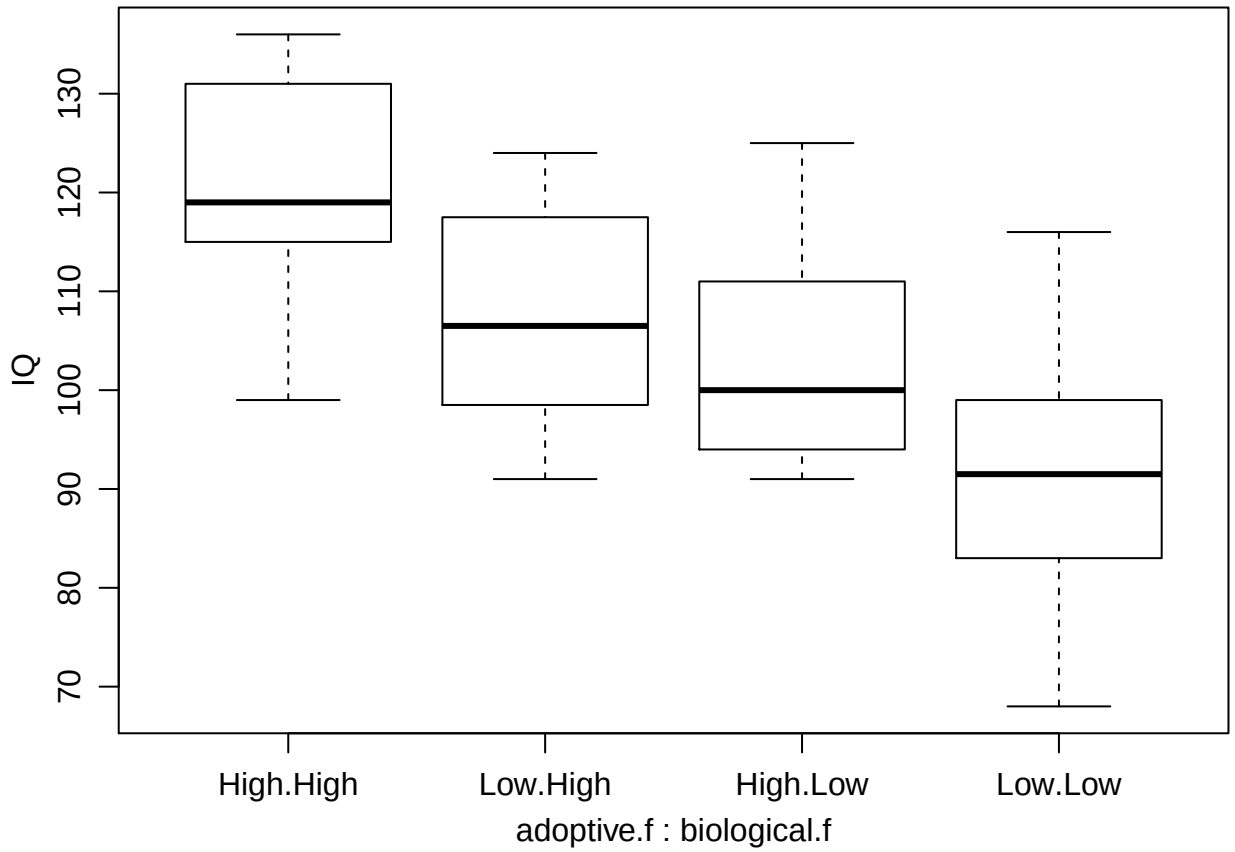
```
adopt$ab <- paste(adopt$adoptive.f, adopt$biological.f, sep='/')
adopt$ab.f <- as.factor(adopt$ab)
table(adopt$ab)
```

```
##
## High/High  High/Low  Low/High   Low/Low
##        10        10         8        10
```

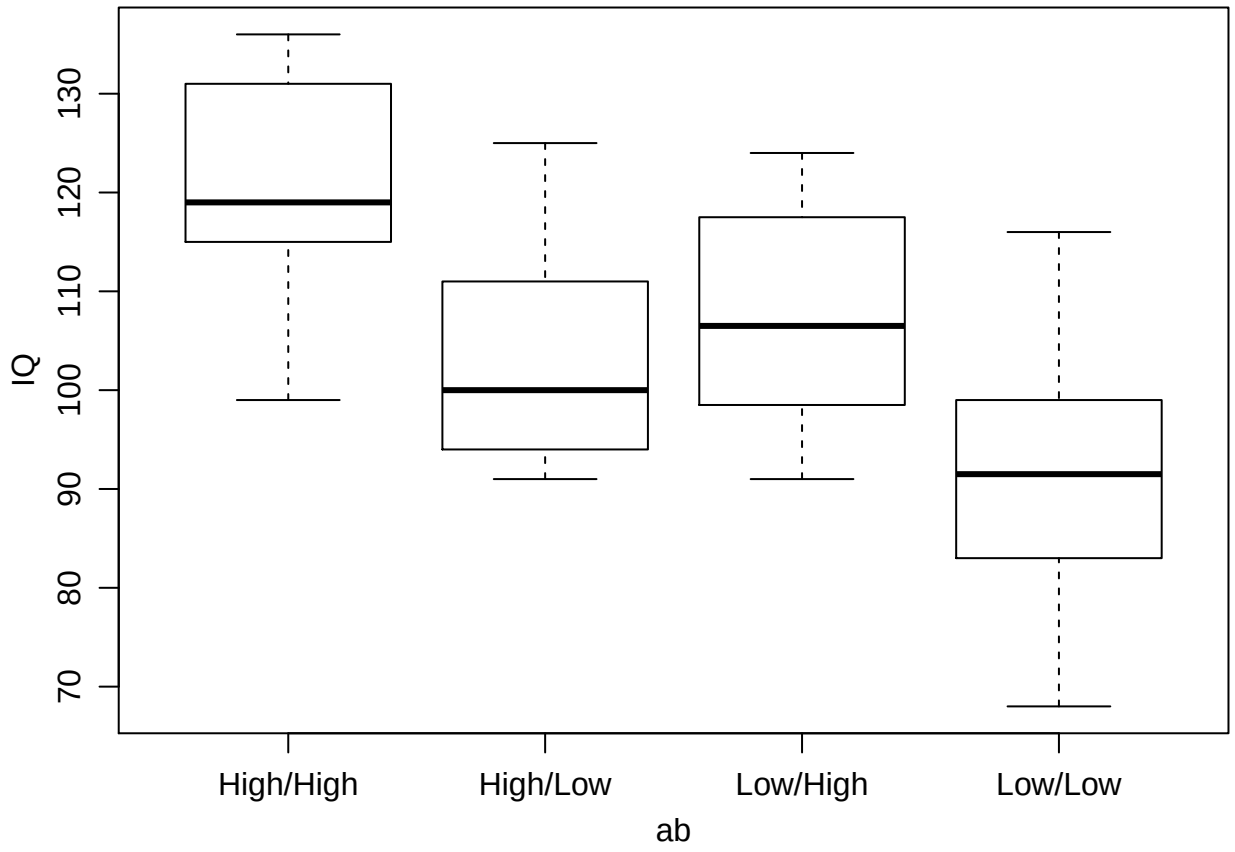
### Looking at the data

It's always a good idea to look at the data. Here are box plots for each cell and an interaction plot of the cell means.

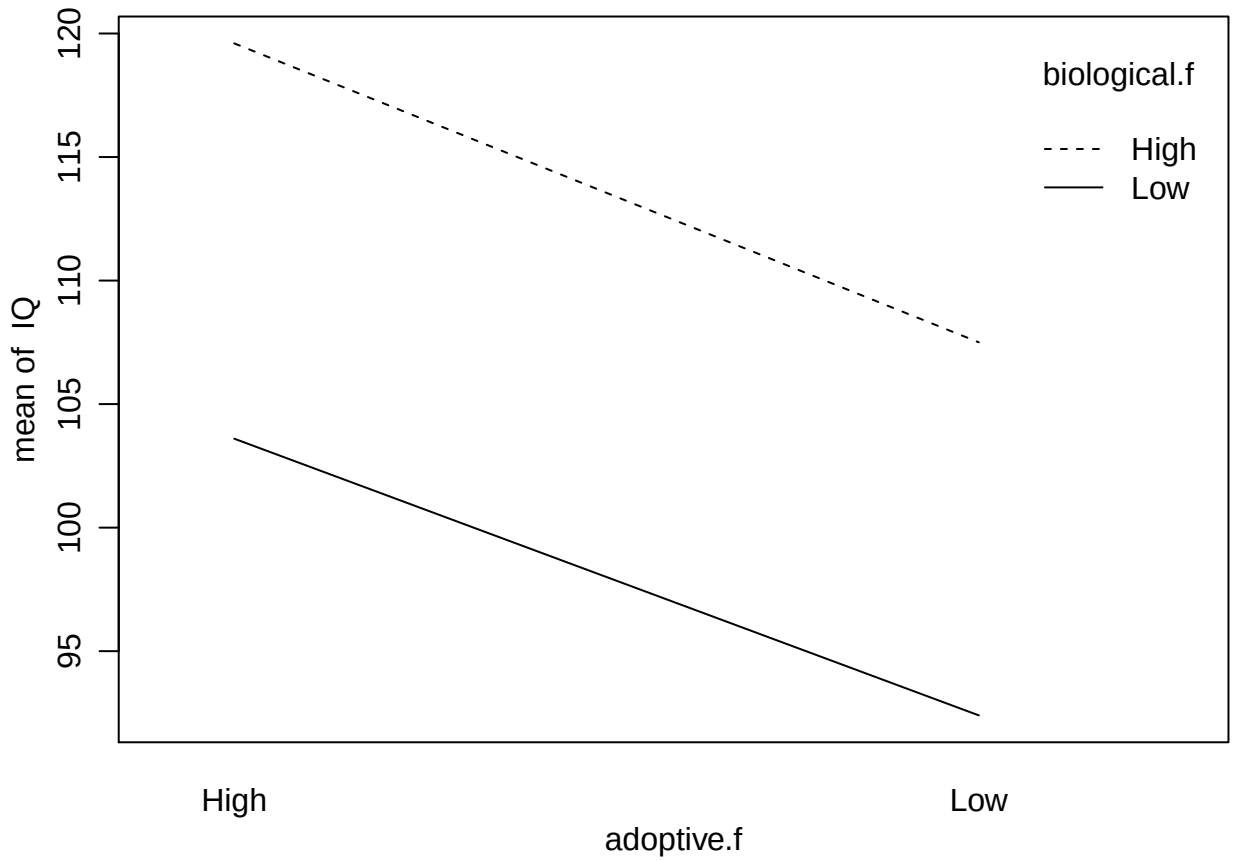
```
par(mar=c(3,3,0,0)+0.3, mgp=c(2,0.8,0))
boxplot(IQ ~ adoptive.f + biological.f, data=adopt)
```




```
boxplot(IQ ~ ab, data=adopt)
```




```
# interaction plot
with(adopt,
  interaction.plot(adoptive.f, biological.f, IQ))
```




You can provide information to boxplot() various ways. The easiest is using a formula (has the tilde in the middle). The left-hand side is the response variable. The right-hand side specifies how to define groups. When you give two variables, you get all combinations (not the same behavior as when fitting a model). The two uses of boxplot() differ only in the ordering of groups, which you could change by the order of additive and biological in either the boxplot() call or the creating of the ab variable.

interaction.plot() is part of base R stats. The three arguments are, in order: the X axis variable, the "trace" variable, and the response variable. The trace variable defines a line in the interaction plot. Both the X axis and trace variables must be factors. A function is applied to the response variables to give one number per cell. The default function is mean, but you can change that with function=.

I recommend you put the variable with the larger # levels on the X axis. It is possible to reorder groups on the X axis by specifying the order of levels. See factor(... , levels=) if you want to do that. The interaction plot is especially easy to read when the groups on the X axis are sorted by increasing mean response. Here's one way to do that. It uses the raw averages for each level of adoptive. You could also use the marginal means. Sorting is not really necessary when there are only two levels.

```
levels(adopt$adoptive.f)
```

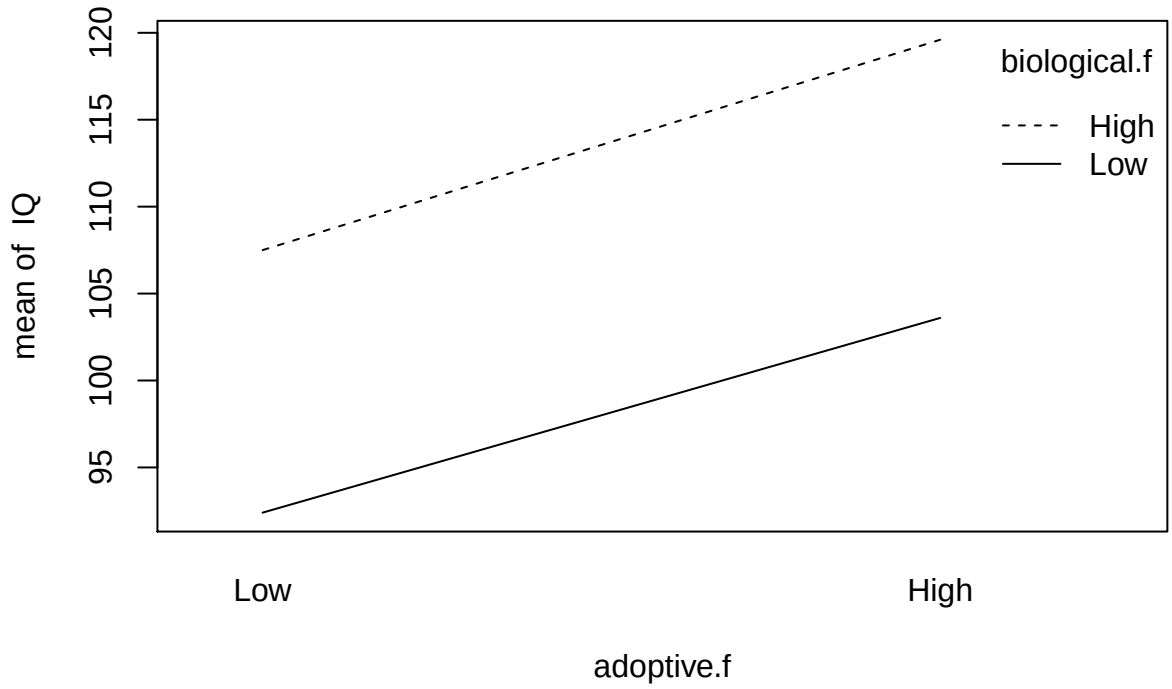
```
## [1] "High" "Low"
```

```
adopt.mean <- tapply(adopt$IQ, adopt$adoptive.f, mean)

adopt$adoptive.f <- factor(adopt$adoptive.f,
  levels=levels(adopt$adoptive.f)[order(adopt.mean)])
levels(adopt$adoptive.f)
```

```
## [1] "Low"  "High"
```

```r
with(adopt,
  interaction.plot(adoptive.f, biological.f, IQ))
```

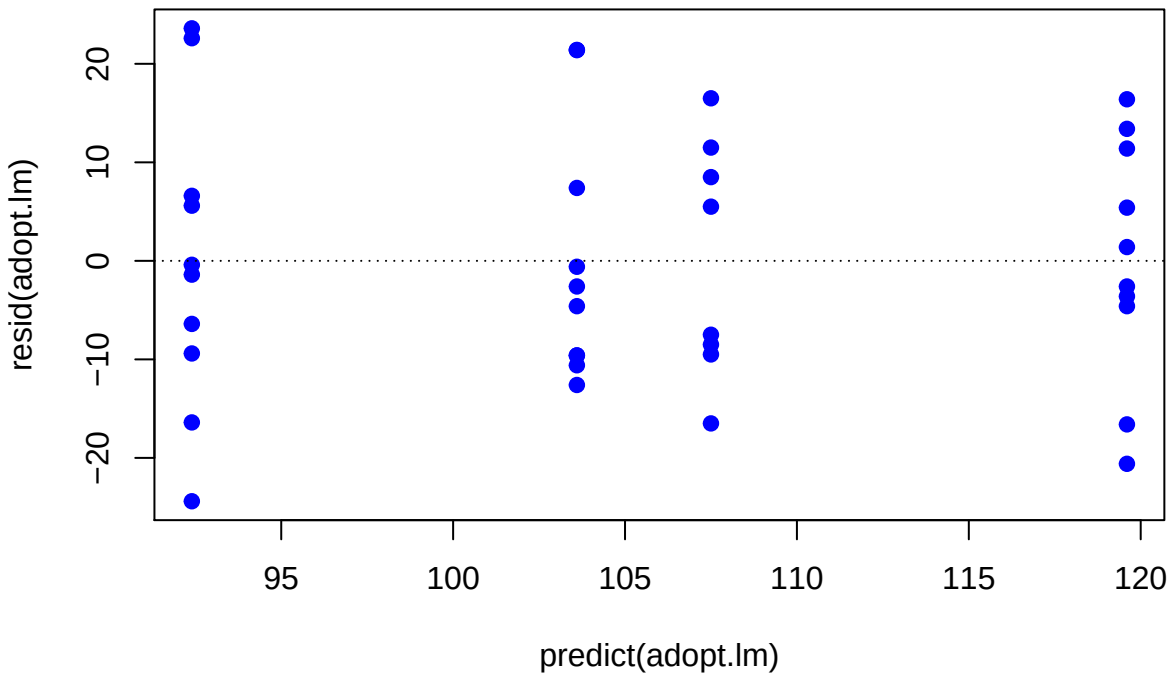



**1 way ANOVA followed by contrasts**

Much of this will be familiar, because it is no different from what we did with 1 way ANOVA for treatments with different types of structure.

You can provide multiple contrasts at the same time to contrasts(). Each contrast is specified by name = vector of coefficients. Different contrasts are separated by commas.

```r
adopt.lm <- lm(IQ ~ ab.f, data=adopt)
plot(predict(adopt.lm), resid(adopt.lm),
  pch=19, col=4)
abline(h=0, lty=3)
```




```
adopt.emm <- emmeans(adopt.lm, 'ab.f')
adopt.emm
```

```
##  ab.f      emmean   SE df lower.CL upper.CL
##  High/High  119.6 4.18 34    111.1      128
##  High/Low   103.6 4.18 34     95.1      112
##  Low/High   107.5 4.67 34     98.0      117
##  Low/Low     92.4 4.18 34     83.9      101
##
## Confidence level used: 0.95
```

```
adopt.contrasts <- contrast(adopt.emm, list(
  adopt = c(0.5, 0.5, -0.5, -0.5),
  biol = c(0.5, -0.5, 0.5, -0.5),
  ab = c(1, -1, -1, 1) ) )
```

Printing the contrasts result gives estimates, se, and T tests for each contrast. Feeding the result into summary() and adding infer=c(T,F) gives you confidence intervals. The first component of the infer= vector controls confidence intervals (TRUE or FALSE); the second controls tests.

```
adopt.contrasts
```

```
##  contrast estimate   SE df t.ratio p.value
##  adopt        11.6 4.31 34 2.704   0.0106
##  biol         15.6 4.31 34 3.609   0.0010
##  ab            0.9 8.62 34 0.104   0.9174
```

```
summary(adopt.contrasts, infer=c(T,F))
```

```
##  contrast estimate   SE df lower.CL upper.CL
##  adopt        11.6 4.31 34     2.89     20.4
##  biol         15.6 4.31 34     6.79     24.3
##  ab            0.9 8.62 34   -16.61     18.4
##
## Confidence level used: 0.95
```

Adding joint=T gives an F test that all contrasts = 0 simultaneously. When specify k-1 contrasts for k groups, this is equivalent to the 1 way ANOVA F test of equality of all groups.

```
test(adopt.contrasts, joint=T)
```

```
##  df1 df2 F.ratio p.value
##    3  34   7.194 0.0007
```

**Interpretation**

No evidence of an interaction, so report main effects. You do need to make sure how you coded the contrast (High - Low or Low - High). In other words, when the estimate is positive, is High larger or is Low larger? Definitely affects your conclusion. If I'm not sure, I print the emmeans. Here, both contrasts were High - Low

If the Interaction was significant, you might want simple effects. Don't currently have those. Write contrasts for the difference of A within each level of B.

Can use spaces in the contrast name if enclose the name in quotes (single or double).

An alternative, when there are not too many cells is to get all pairwise differences and extract those you care about. pairs() applied to an emmeans object gives you all pairwise differences. Tukey multiple comparisons adjustment applied by default. Want to turn that off. Even if you want to adjust, it won't be for all pairs. I would use Bonferroni, or no adjustment.

```
contrast(adopt.emm, list(
  'adopt in High Biol'=c(1, 0, -1,0),
  'adopt in Low Biol'=c(0, 1, 0, -1) ) )
```

```
##  contrast           estimate   SE df t.ratio p.value
##  adopt in High Biol     12.1 6.27 34 1.930   0.0620
##  adopt in Low Biol      11.2 5.91 34 1.895   0.0667
```

```
pairs(adopt.emm, adjust='none')
```

```
##  contrast             estimate   SE df t.ratio p.value
##  High/High - High/Low     16.0 5.91 34  2.706  0.0106
##  High/High - Low/High     12.1 6.27 34  1.930  0.0620
##  High/High - Low/Low      27.2 5.91 34  4.601  0.0001
##  High/Low - Low/High      -3.9 6.27 34 -0.622  0.5381
##  High/Low - Low/Low       11.2 5.91 34  1.895  0.0667
##  Low/High - Low/Low       15.1 6.27 34  2.408  0.0216
```

**using a factor effects model:**

This is specified in terms of main effects (each factor by itself) and their interaction. The : indicates the interaction. A shortcut is to use *, which expands into the interaction and all the component main effects. These two models are identical, except perhaps for the order of the terms. I usually write out all the terms (first lm call) so I can control the order.

```
adopt.lm2 <- lm(IQ ~ adoptive.f + biological.f + adoptive.f:biological.f,
  data=adopt)

adopt.lm2 <- lm(IQ ~ adoptive.f*biological.f,
  data=adopt)
```

You can then get marginal means and contrasts of marginal means by telling emmeans() the names of the factors you care about. This could be both, or just one.

Specifying both adoptive and biological effects as a vector gives you the cell means. Specifying them as a list gives you both sets of marginal means. Specifying just one factor name gives you just the marginal means for that factor.

The note sounds scary, but it is just a reminder that you are averaging simple effects. That's very appropriate here because of the apparent absence of an interaction.

```
adopt.emm2 <- emmeans(adopt.lm2, c('adoptive.f','biological.f'))
adopt.emm2
```

```
##  adoptive.f biological.f emmean   SE df lower.CL upper.CL
##  Low        High          107.5 4.67 34     98.0      117
##  High       High          119.6 4.18 34    111.1      128
##  Low        Low            92.4 4.18 34     83.9      101
##  High       Low           103.6 4.18 34     95.1      112
##
## Confidence level used: 0.95
```

```
adopt.emm3 <- emmeans(adopt.lm2, list('adoptive.f','biological.f'))
```

```
## NOTE: Results may be misleading due to involvement in interactions
## NOTE: Results may be misleading due to involvement in interactions
```

```
adopt.emm3
```

```
## $`emmeans of adoptive.f`
##  adoptive.f emmean   SE df lower.CL upper.CL
##  Low           100 3.14 34     93.6      106
##  High          112 2.96 34    105.6      118
##
## Results are averaged over the levels of: biological.f
## Confidence level used: 0.95
##
## $`emmeans of biological.f`
##  biological.f emmean   SE df lower.CL upper.CL
##  High            114 3.14 34      107      120
##  Low              98 2.96 34       92      104
##
## Results are averaged over the levels of: adoptive.f
## Confidence level used: 0.95
```

The nice thing about the effects parameterization is that you can write contrasts in terms of the marginal means or request all pairwise differences. When you use the list form to get multiple sets of marginal means, contrast does the expected.

```
contrast(adopt.emm3, list(highlow = c(1, -1)))
```

```
## $`emmeans of adoptive.f`
##  contrast estimate   SE df t.ratio p.value
```

```
##  highlow     -11.6 4.31 34 -2.704  0.0106
##
## Results are averaged over the levels of: biological.f
##
## $`emmeans of biological.f`
##  contrast estimate   SE df t.ratio p.value
##  highlow      15.6 4.31 34 3.609   0.0010
##
## Results are averaged over the levels of: adoptive.f
```

```
pairs(adopt.emm3)
```

```
## $`emmeans of adoptive.f`
##  contrast   estimate   SE df t.ratio p.value
##  Low - High    -11.6 4.31 34 -2.704  0.0106
##
## Results are averaged over the levels of: biological.f
##
## $`emmeans of biological.f`
##  contrast   estimate   SE df t.ratio p.value
##  High - Low     15.6 4.31 34 3.609   0.0010
##
## Results are averaged over the levels of: adoptive.f
```

You get simple effects easily by calculating a contrast "by" the other factor, starting with the "all combinations" emmeans result (the vector, not the list).

```
contrast(adopt.emm2, list(highlow = c(1, -1)),
  by='biological.f')
```

```
## biological.f = High:
##  contrast estimate   SE df t.ratio p.value
##  highlow     -12.1 6.27 34 -1.930  0.0620
##
## biological.f = Low:
##  contrast estimate   SE df t.ratio p.value
##  highlow     -11.2 5.91 34 -1.895  0.0667
```

The most reliable way to get type III tests is to use joint_tests() applied to the "all combinations" (i.e. the vector with 2 variables) emmeans object. anova() gives sequential (type I tests), which are not the same for this data set. You should make sure you understand why they are not the same here, and why they don't match contrast results.

```
joint_tests(adopt.emm2)
```

```
##  model term              df1 df2 F.ratio p.value
##  adoptive.f                1  34   7.310 0.0106
##  biological.f              1  34  13.024 0.0010
##  adoptive.f:biological.f   1  34   0.011 0.9174
```

```
anova(adopt.lm2)
```

```
## Analysis of Variance Table
##
## Response: IQ
##                          Df Sum Sq Mean Sq F value    Pr(>F)
## adoptive.f                1 1477.6 1477.63  8.4561 0.0063663 **
## biological.f              1 2291.5 2291.47 13.1135 0.0009445 ***
```

```
## adoptive.f:biological.f  1    1.9    1.91  0.0109 0.9174370
## Residuals               34 5941.2  174.74
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

**Other useful things to know:**

Many commonly used contrasts have pre-defined functions. See ¿contrast-methods' for that list.