

Introduction to contrasts, and using emmeans in R

Ian Dworkin

15 Feb 2025

Contents

| | | |
|----------|---|----------|
| 1 | Introduction to contrasts, and using emmeans | 1 |
| 1.1 | Libraries | 1 |
| 1.2 | First example with the iris data set | 2 |
| 1.3 | Reviewing (or introducing) ourselves to the “design matrix” | 5 |
| 1.4 | So what’s the problem? | 6 |
| 1.5 | using emmeans | 8 |
| 1.6 | An example of interaction contrasts from a linear mixed effects model | 12 |
| 1.7 | links for other tutorials on contrasts | 21 |

1 Introduction to contrasts, and using emmeans

1.1 Libraries

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
library(emmeans)
```

```
## Welcome to emmeans.
```

```
## Caution: You lose important information if you filter this package’s results.
```

```
## See ’?’ untidy’
```

```
library(car)
```

```
## Loading required package: carData
```

```
library(ggplot2)
```

```
library(ggbeeswarm)
```

1.2 First example with the iris data set

```
data(iris)
```

1.2.1 The iris data

```
head(iris)
```

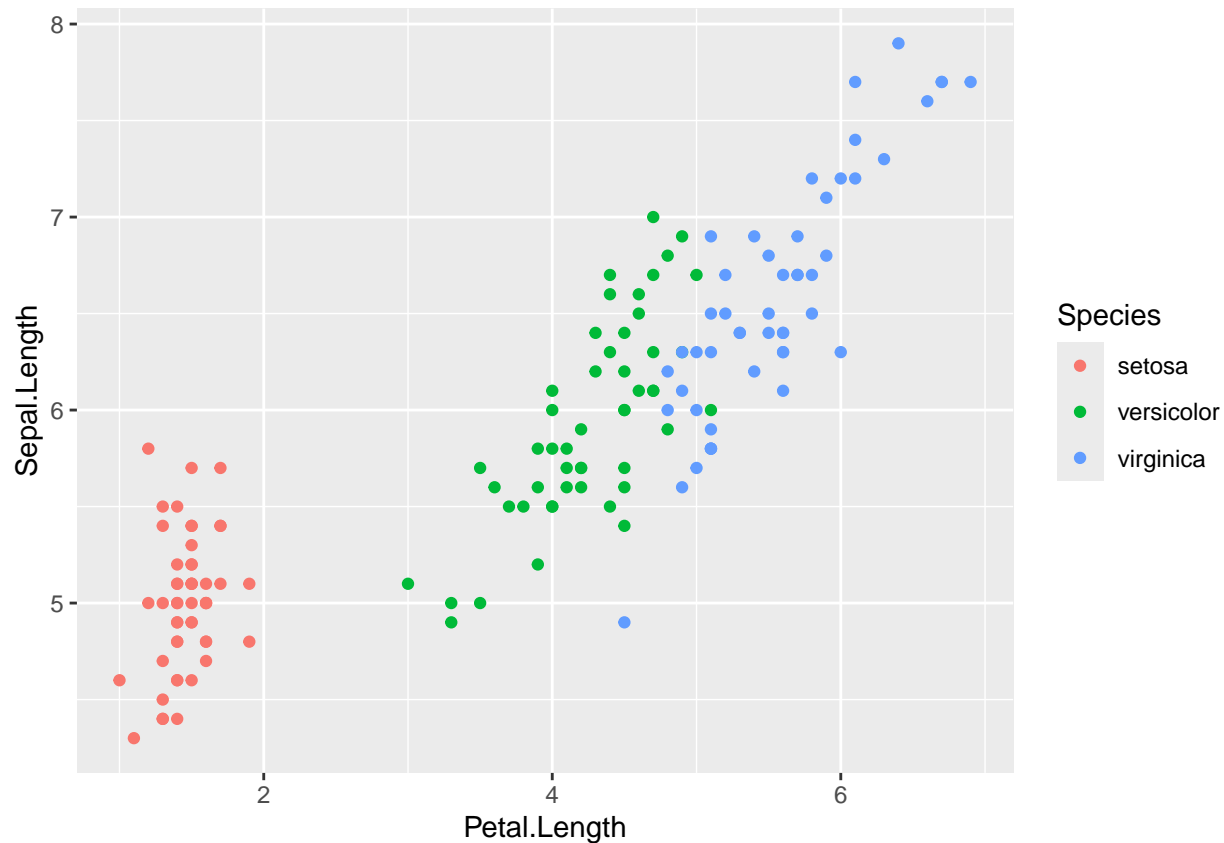
```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1         5.1         3.5         1.4         0.2   setosa
## 2         4.9         3.0         1.4         0.2   setosa
## 3         4.7         3.2         1.3         0.2   setosa
## 4         4.6         3.1         1.5         0.2   setosa
## 5         5.0         3.6         1.4         0.2   setosa
## 6         5.4         3.9         1.7         0.4   setosa
```

```
with(iris, table(Species))
```

```
## Species
##      setosa versicolor  virginica
##         50         50         50
```

1.2.2 quick plot of the iris data

```
ggplot(iris, aes(y = Sepal.Length, x = Petal.Length, col = Species)) +  
  geom_point()
```



1.2.3 Let's consider a situation

(this “question” is made up to facilitate the analysis!)

While we want to compare the differences between species for these morphological traits, we are specifically interested in comparisons of *versicolor* to the other two species. So how may we do this?

1.2.4 What if we just fit a linear model (simple one way ANOVA)?

```
mod1 <- lm(Sepal.Length ~ Species,
           data = iris)
```

```
anova(mod1)
```

```
## Analysis of Variance Table
##
## Response: Sepal.Length
##          Df Sum Sq Mean Sq F value Pr(>F)
## Species    2   63.2   31.61    119 <2e-16
## Residuals 147   39.0    0.27
```

```
summary(mod1)
```

```
##
## Call:
## lm(formula = Sepal.Length ~ Species, data = iris)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.688 -0.329 -0.006  0.312  1.312
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.0060     0.0728   68.76 < 2e-16
## Speciesversicolor  0.9300     0.1030    9.03 8.8e-16
## Speciesvirginica   1.5820     0.1030   15.37 < 2e-16
##
## Residual standard error: 0.515 on 147 degrees of freedom
## Multiple R-squared:  0.619, Adjusted R-squared:  0.614
## F-statistic: 119 on 2 and 147 DF, p-value: <2e-16
```

By default this shows the differences of each of *versicolor* and *virginica* to *setosa*. The intercept is the mean Sepal length for *setosa*, and the other two coefficients are the treatment contrasts (differences) for the other two species with *setosa*.

1.2.5 A couple of ways of thinking about this that may help

Compute the species means

```
SL_means_Species <- with(iris,
  tapply(Sepal.Length, Species, mean))

SL_means_Species
```

```
##      setosa versicolor  virginica
##      5.01      5.94      6.59
```

As this is such a simple model (just a single factor) these means will be the same as the predicted values from the models:

```
species_predictedVals <- unique(predict(mod1))

names(species_predictedVals) <- c("setosa", "versicolor", "virginica")

species_predictedVals
```

```
##      setosa versicolor  virginica
##      5.01      5.94      6.59
```

and the differences between the means can be estimated simply:

```
SL_means_Species["versicolor"] - SL_means_Species["setosa"]

## versicolor
##      0.93
```

```
SL_means_Species["virginica"] - SL_means_Species["setosa"]
```

```
## virginica
##      1.58
```

1.2.6 These are not the difference(s) we are looking for

But we are not interested in the difference between setosa and the other two species, but in comparisons to versicolor.

1.3 Reviewing (or introducing) ourselves to the “design matrix”

One easy fix for this without having to think about custom contrasts at all (yet!) is to re-level our factor so that the base level is versicolor:

```
iris$Species2 <- relevel(iris$Species, "versicolor")
```

Now if we fit the same model, just with *versicolor* representing the intercept

```
mod1_alt <- lm(Sepal.Length ~ Species2,
               data = iris)
```

```
summary(mod1_alt)
```

```
##
## Call:
## lm(formula = Sepal.Length ~ Species2, data = iris)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.688 -0.329 -0.006  0.312  1.312
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.9360     0.0728  81.54 < 2e-16
## Species2setosa    -0.9300     0.1030  -9.03 8.8e-16
## Species2virginica  0.6520     0.1030   6.33 2.8e-09
##
## Residual standard error: 0.515 on 147 degrees of freedom
## Multiple R-squared:  0.619, Adjusted R-squared:  0.614
## F-statistic: 119 on 2 and 147 DF, p-value: <2e-16
```

Why did this work? Because we re-organized the design matrix a bit.

Our initial design matrix had this format

```
unique(model.matrix(mod1))
```

```
##      (Intercept) Speciesversicolor Speciesvirginica
## 1              1              0              0
## 51             1              1              0
## 101            1              0              1
```

When we changed the reference level, the intercept was now *versicolor* instead of *setosa*.

```
unique(model.matrix(mod1_alt))
```

```
##      (Intercept) Species2setosa Species2virginica
## 1              1              1              0
## 51             1              0              0
## 101            1              0              1
```

1.4 So what's the problem?

This solution is OK for this really really simple case. It is not particularly useful for more complex models, or complex contrasts.

What if I wanted to compare the difference between *versicolor* and the other two species? How should I go about doing that?

Here are the predicted values again:

```
species_predictedVals
```

```
##      setosa versicolor  virginica
##      5.01      5.94      6.59
```

One way of thinking about this is comparing the predicted value of *setosa* to the mean of the predicted values of the other two species

```
species_predictedVals["versicolor"] - mean(species_predictedVals[c("setosa", "virginica")])
```

```
## versicolor
##      0.139
```

Mathematically,

$$1 \times \hat{\mu}_{versicolor} - \frac{1}{2} \times \hat{\mu}_{setosa} - \frac{1}{2} \times \hat{\mu}_{virginica}$$

i.e

$$1 \times \hat{\mu}_{versicolor} - \frac{1}{2}(\hat{\mu}_{setosa} + \hat{\mu}_{virginica})$$

with coefficients

$$(1, -\frac{1}{2}, -\frac{1}{2})$$

```
1*species_predictedVals["versicolor"] - (1/2)*species_predictedVals["setosa"] - (1/2)*species_predictedVals["virginica"]
```

```
## versicolor
##      0.139
```

That is, the difference between the predicted value of *versicolor* and half the predicted value of each of the other two species.

```
contrast_vector_example <- c(1, -0.5, -0.5)

sum(contrast_vector_example)
```

```
## [1] 0
```

This is the same as saying if the predicted value of *setosa* was the same as the mean of the predicted values of the other two species, the difference should be zero.

$$\hat{\mu}_{versicolor} \cong \frac{1}{2}(\hat{\mu}_{setosa} + \hat{\mu}_{virginica})$$

This is an example of a custom contrast. Our $(1, -\frac{1}{2}, -\frac{1}{2})$ represents the contrasts coefficients.

1.4.1 Contrast coefficients

We can construct our contrast coefficients with a few basic “rules” (See Crawley pg 368-369)

- Treatments to be lumped together (like *setosa* and *virginica*) get the same sign (+ or −)
- Groups of means to be contrasted get opposite signs
- a contrast coefficient of 0 is used if that factor level is to be excluded from the comparison
- The sum of the contrasts coefficients add up to zero

1.4.2 OK, great?

Well a couple of issues. First the way I just wrote this, is annoying for anything but a toy problem like this. More importantly I want to be able to assess the uncertainty in this difference I computed. That means some fiddly algebra with the standard errors for each estimate.

1.4.3 custom contrasts in base R.

Not shockingly in base R, we can use the `contrasts` function to help set up our custom contrasts

We can see what the default contrasts are

```
contrasts(iris$Species)
```

```
##           versicolor virginica
## setosa             0          0
## versicolor         1          0
## virginica           0          1
```

```
contrasts(iris$Species2)
```

```
##           setosa virginica
## versicolor     0          0
## setosa         1          0
## virginica      0          1
```

Say in addition to the contrast above, I planned to also contrast the difference between *setosa* and *virginica*. How could we set this up?

```
levels(iris$Species)

## [1] "setosa"      "versicolor" "virginica"

setosa_virginica <- c(1, 0, -1) # setosa VS virginica

versicolor_VS_others <- c(-0.5, 1, -0.5)
```

Unfortunately using these effectively with base R requires a few additional algebraic steps, which we want to avoid for now. So let's use emmeans which can take care of all of this for us.

1.5 using emmeans

While you can write out custom contrasts, I find it really a pain in the butt, so use emmeans!

1.5.1 Getting the estimated means and their confidence intervals with emmeans

```
summary(mod1)

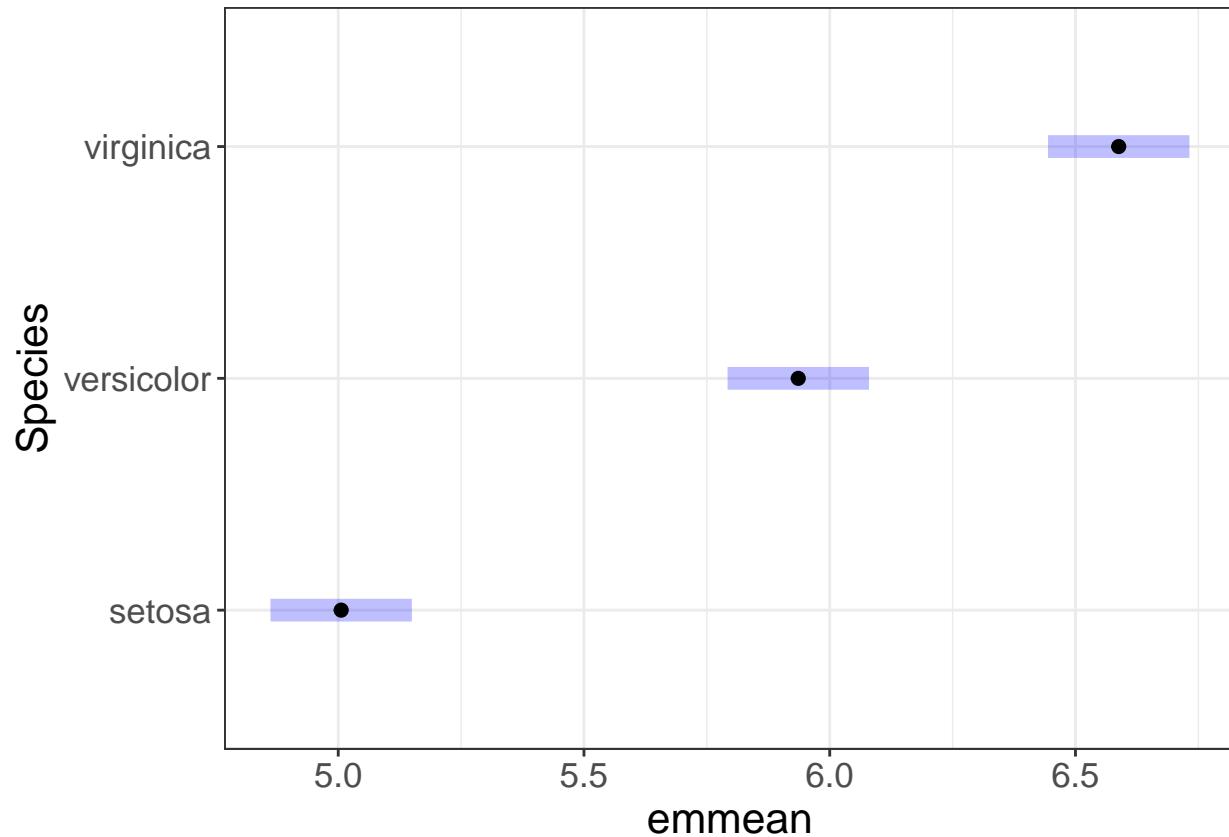
##
## Call:
## lm(formula = Sepal.Length ~ Species, data = iris)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.688 -0.329 -0.006  0.312  1.312
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.0060     0.0728   68.76 < 2e-16
## Speciesversicolor  0.9300     0.1030    9.03 8.8e-16
## Speciesvirginica   1.5820     0.1030   15.37 < 2e-16
##
## Residual standard error: 0.515 on 147 degrees of freedom
## Multiple R-squared:  0.619, Adjusted R-squared:  0.614
## F-statistic: 119 on 2 and 147 DF, p-value: <2e-16

spp_em <- emmeans(mod1, ~Species) # means
spp_em

## Species      emmean      SE df lower.CL upper.CL
## setosa         5.01 0.0728 147     4.86     5.15
## versicolor     5.94 0.0728 147     5.79     6.08
## virginica      6.59 0.0728 147     6.44     6.73
##
## Confidence level used: 0.95
```



```
plot(spp_em) +
  theme_bw() +
  theme(text = element_text(size = 16))
```



1.5.2 Setting up our custom contrasts in emmeans

We can use the `contrast` function (note, not `contrasts` with an s at the end) and provide these to get the contrasts we are interested in.

```
iris_custom_contrasts <- contrast(spp_em,
  list(versicolor_VS_others = versicolor_VS_others,
        virginica_VS_setosa = c(1, 0, -1)))

iris_custom_contrasts
```

```
## contrast      estimate      SE df t.ratio p.value
## versicolor_VS_others  0.139 0.0892 147  1.560  0.1212
## virginica_VS_setosa   -1.582 0.1030 147 -15.370 <.0001
```

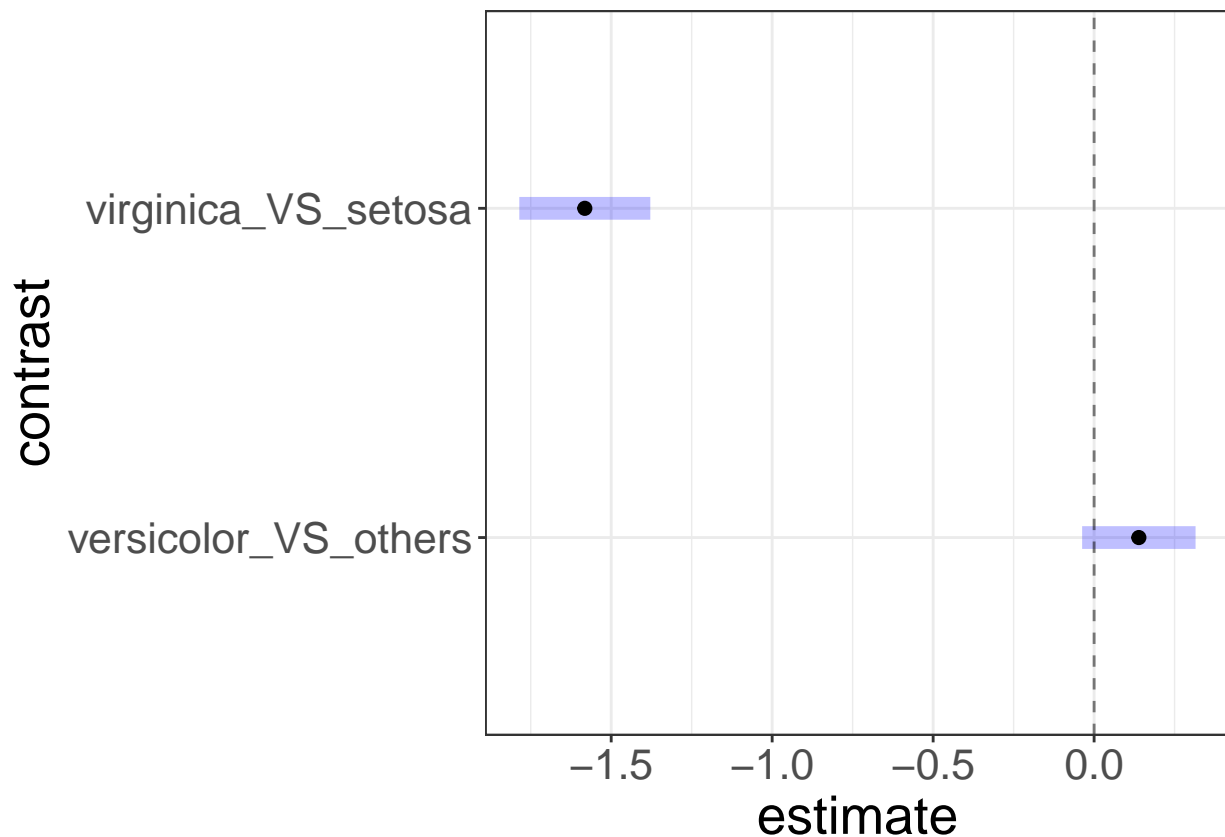
Even more useful we can also get the confidence intervals on the contrasts!

```
confint(iris_custom_contrasts )
```

```
## contrast      estimate      SE  df lower.CL upper.CL
## versicolor_VS_others    0.139 0.0892 147  -0.037   0.315
## virginica_VS_setosa    -1.582 0.1030 147  -1.785  -1.379
##
## Confidence level used: 0.95
```

Which can be plotted easily (ggplot2 object)

```
plot(iris_custom_contrasts) +
  geom_vline(xintercept = 0, lty = 2 , alpha = 0.5) +
  theme_bw() +
  theme(text = element_text(size = 20))
```



1.5.3 Flexibility with emmeans for many types of contrasts

This approach allows much more flexibility. see `?contrast-methods`

Importantly and helpfully, for broader sets of contrasts emmeans does much automatically

If we had *a priori* (really important!), planned to compare all species to each other we could use this

```
contrast(spp_em, method = "pairwise")
```

```
## contrast      estimate      SE  df t.ratio p.value
## setosa - versicolor    -0.930 0.103 147  -9.030  <.0001
```

```
## setosa - virginica      -1.582 0.103 147 -15.370 <.0001
## versicolor - virginica -0.652 0.103 147  -6.330 <.0001
##
## P value adjustment: tukey method for comparing a family of 3 estimates
```

```
pairs(spp_em) # same as above, just a shortcut function.
```

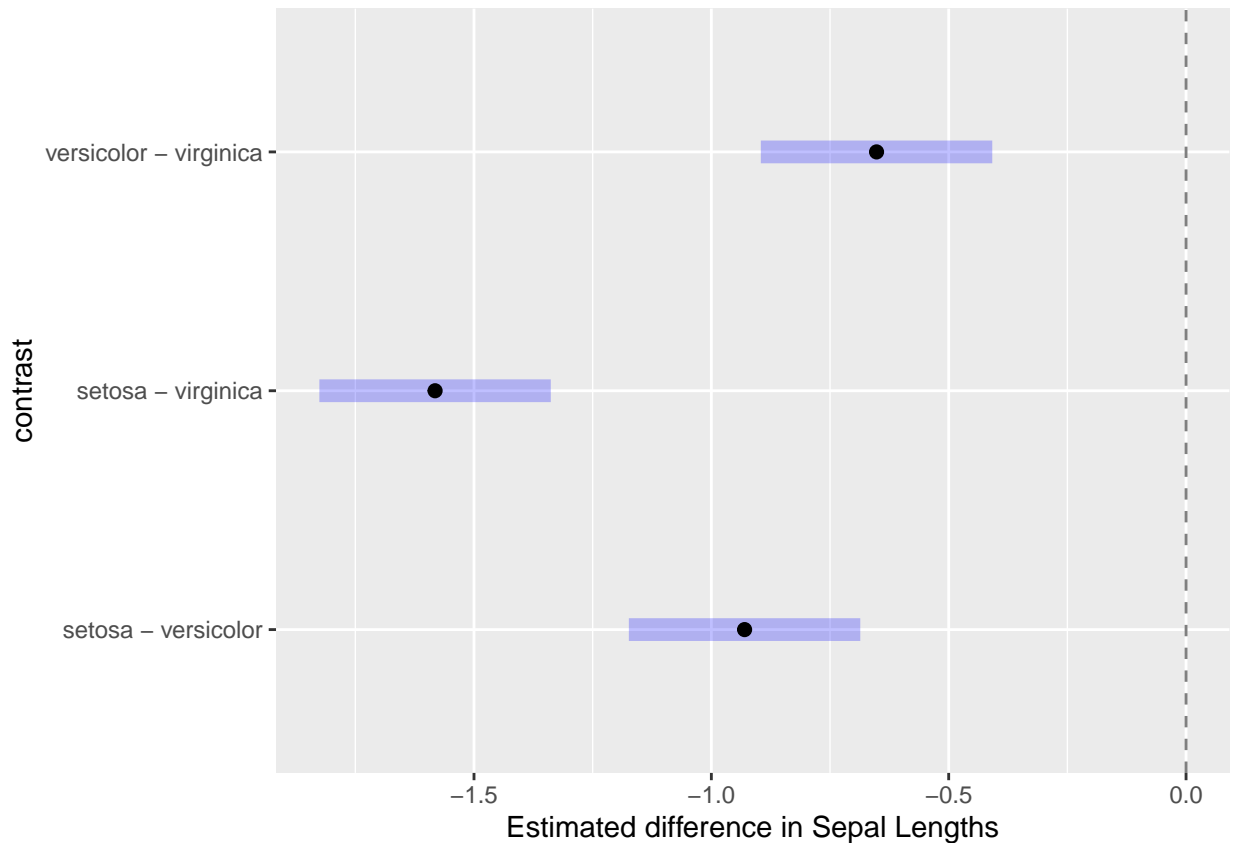
```
## contrast      estimate    SE  df t.ratio p.value
## setosa - versicolor    -0.930 0.103 147  -9.030 <.0001
## setosa - virginica     -1.582 0.103 147 -15.370 <.0001
## versicolor - virginica -0.652 0.103 147  -6.330 <.0001
##
## P value adjustment: tukey method for comparing a family of 3 estimates
```

```
confint(pairs(spp_em))
```

```
## contrast      estimate    SE  df lower.CL upper.CL
## setosa - versicolor    -0.930 0.103 147  -1.174  -0.686
## setosa - virginica     -1.582 0.103 147  -1.826  -1.338
## versicolor - virginica -0.652 0.103 147  -0.896  -0.408
##
## Confidence level used: 0.95
## Conf-level adjustment: tukey method for comparing a family of 3 estimates
```

Note how it automatically adjusts for the multiple comparisons in this case!

```
plot(pairs(spp_em)) +
  geom_vline(xintercept = 0, lty = 2 , alpha = 0.5) +
  xlab("Estimated difference in Sepal Lengths")
```



and, usefully we could have gotten our setosa VS virginica (i.e. excluding versicolor) comparison this way

```
pairs(spp_em, exclude = 2)
```

```
## contrast      estimate    SE  df t.ratio p.value
## setosa - virginica    -1.58 0.103 147 -15.370 <.0001
```

1.6 An example of interaction contrasts from a linear mixed effects model

Here is an example from a much more complex linear mixed model.

The data is from an artificial selection experiment in *Drosophila melanogaster*, where, over the course of more than 350 generations (when this data was generated) flies were selected sex concordantly or discordantly for body size.

We have been investigating trait specific changes in sexual size dimorphism, and how it compares to what is observed in the control lineages (where the ancestral pattern of female biased size dimorphism occurs). So we are interested in particular in the effects of the interaction between sex and selective treatment. How best to examine this?

A subset of the data from this experiment is available in the same github repository as “contrast_tutorial_dat.RData”

1.6.1 The data

```
load("./contrast_tutorial_dat.RData")

size_dat <- contrast_tutorial_dat
```

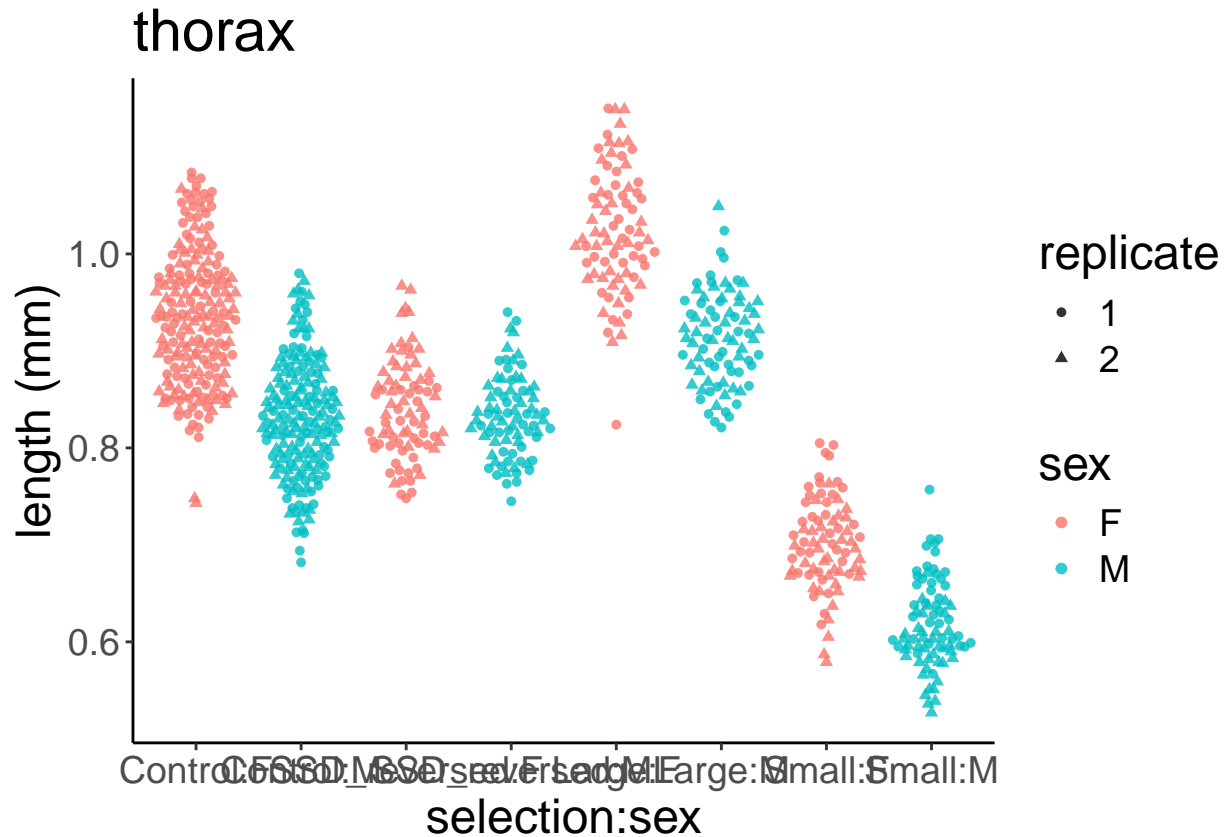
What the data frame looks like

```
head(size_dat)
```

| | ## | sex | selection | replicate | sampling | trait | individual_id | length | repeat_measure |
|----|-----|-----|-----------|-----------|----------|--------|---------------|--------|----------------|
| ## | 100 | F | Control | 1 | R | thorax | 1 | 0.934 | 1 |
| ## | 101 | F | Control | 1 | R | thorax | 2 | 1.057 | 1 |
| ## | 102 | F | Control | 1 | R | thorax | 3 | 0.908 | 1 |
| ## | 103 | F | Control | 1 | R | thorax | 4 | 1.016 | 1 |
| ## | 104 | F | Control | 1 | R | thorax | 5 | 0.854 | 1 |
| ## | 105 | F | Control | 1 | R | thorax | 6 | 0.973 | 1 |

1.6.2 A quick visual summary

```
ggplot(size_dat,
  aes( y = length, x = selection:sex, col = sex, shape = replicate)) +
  geom_quasirandom(alpha = 0.8, size = 1.4) +
  labs( y = "length (mm)" ) +
  ggtitle("thorax") +
  theme_classic() +
  theme(text = element_text(size = 18))
```



Here is the model we used for the study for this trait. I will briefly discuss it with you, but importantly you will see it is more complicated than our toy example above.

1.6.3 review: Why log transform the response variable?

Note: with the model that I am multiplying thorax length by 1000 (to convert to μm) and then using a \log_2 transformation on it.

Why am I doing the log transformation in the first place?

I am doing the transformation directly in the model call itself. This is not necessary, but emmeans will recognize this, and facilitates backtransformation of our estimates.

1.6.4 The model

```
mod1_thorax <- lmer(log2(length*1000) ~ (sex + selection + sampling)^2 + (0 + sex | replicate:selection),
  data = size_dat,
  subset = repeat_measure == "1")
```

1.6.5 Limited value of ANOVAs for interaction effect

So how do we make sense of how sexual size dimorphism is changing among the selective treatments?

An Anova (putting aside limitations on ANOVA in general, and for mixed models in particular for the moment) is not particularly informative

```
Anova(mod1_thorax)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: log2(length * 1000)
##               Chisq Df Pr(>Chisq)
## sex           140.67  1    < 2e-16
## selection     242.72  3    < 2e-16
## sampling       18.90  1    1.4e-05
## sex:selection   38.24  3    2.5e-08
## sex:sampling     5.29  1    0.02141
## selection:sampling 20.28  3    0.00015
```

We see a significant effect of the interaction between sex and sampling, but so what? What selective treatment is this a result of and what are the magnitudes of the change? An Anova is not useful for this.

1.6.6 The model coefficients (in standard treatment contrast form) are not super helpful either

For a model this complicated we also see that the summary table of coefficients is not simple to parse for our needs (we could do lots of adding up of various terms, but how about the standard errors for these...)

```
summary(mod1_thorax)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: log2(length * 1000) ~ (sex + selection + sampling)^2 + (0 + sex |
##   replicate:selection)
##   Data: size_dat
##   Subset: repeat_measure == "1"
##
## REML criterion at convergence: -1270
##
## Scaled residuals:
##   Min      1Q  Median      3Q      Max
## -3.476 -0.663 -0.013  0.646  2.870
##
## Random effects:
##   Groups                Name Variance Std.Dev. Corr
##   replicate:selection sexF 0.00128  0.0358
##                      sexM 0.00165  0.0406  0.80
##   Residual                0.00759  0.0871
## Number of obs: 668, groups:  replicate:selection, 8
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)      9.883101   0.027436  360.22
## sexM             -0.179839   0.022417  -8.02
## selectionSSD_reversed -0.154340   0.039091  -3.95
## selectionLarge      0.118572   0.039091   3.03
## selectionSmall     -0.386438   0.039091  -9.89
## samplingS         -0.039407   0.014452  -2.73
## sexM:selectionSSD_reversed 0.152674   0.031050   4.92
```

```
## sexM:selectionLarge      0.000473  0.031050  0.02
## sexM:selectionSmall     -0.015909  0.031050 -0.51
## sexM:samplingS          0.031150  0.013539  2.30
## selectionSSD_reversed:samplingS 0.004112  0.018862  0.22
## selectionLarge:samplingS 0.029080  0.018862  1.54
## selectionSmall:samplingS -0.055738  0.018862 -2.96
```

```
##
## Correlation matrix not shown by default, as p = 13 > 12.
## Use print(x, correlation=TRUE) or
##      vcov(x)          if you need it
```

1.6.7 Using emmeans for this model

This is where contrasts become SO useful!

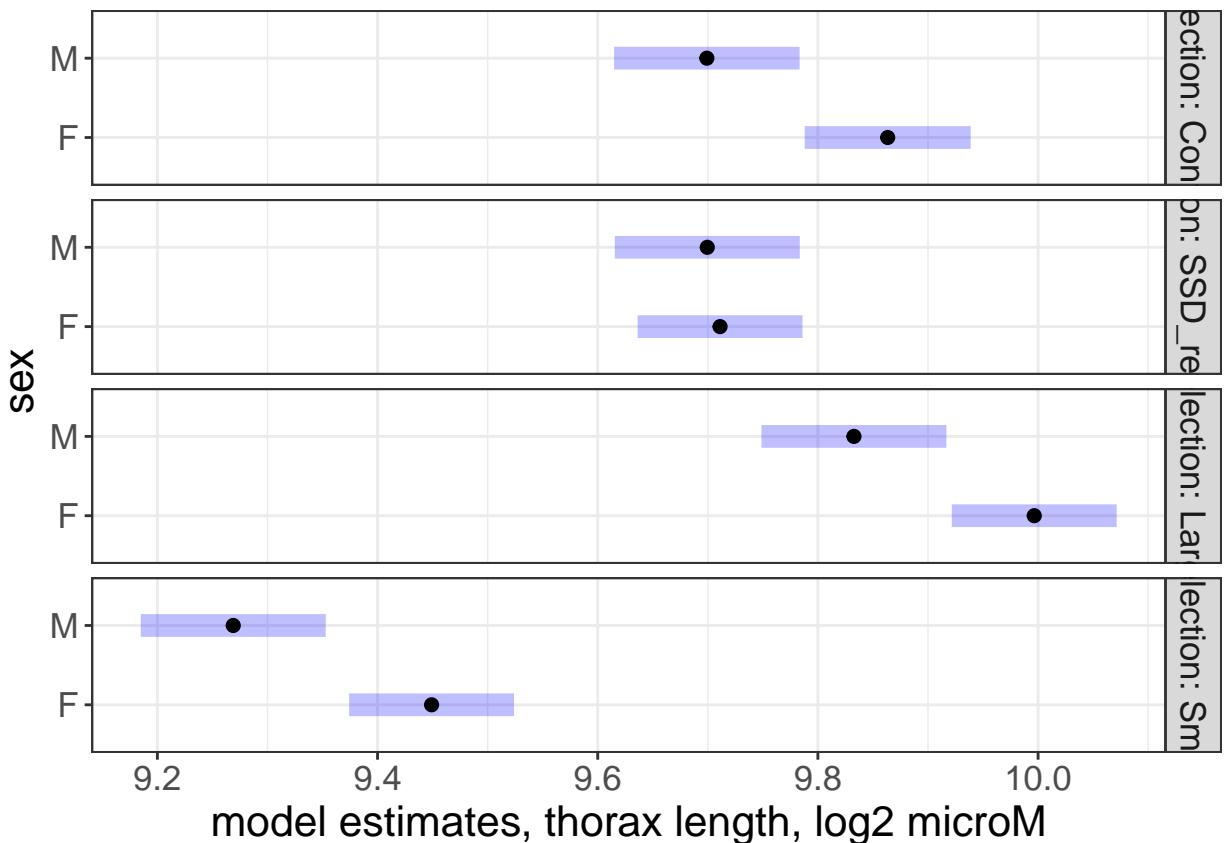
1.6.7.1 Estimated marginal means First, just the estimated means, for each sex, by selective regime

```
thorax_emm <- emmeans(mod1_thorax, specs = ~ sex | selection)
```

```
thorax_emm
```

```
## selection = Control:
## sex emmean      SE    df lower.CL upper.CL
## F      9.86 0.0268 3.88      9.79      9.94
## M      9.70 0.0302 3.94      9.61      9.78
##
## selection = SSD_reversed:
## sex emmean      SE    df lower.CL upper.CL
## F      9.71 0.0271 4.05      9.64      9.79
## M      9.70 0.0303 4.02      9.62      9.78
##
## selection = Large:
## sex emmean      SE    df lower.CL upper.CL
## F     10.00 0.0271 4.05      9.92     10.07
## M      9.83 0.0303 4.02      9.75      9.92
##
## selection = Small:
## sex emmean      SE    df lower.CL upper.CL
## F      9.45 0.0271 4.05      9.37      9.52
## M      9.27 0.0303 4.02      9.18      9.35
##
## Results are averaged over the levels of: sampling
## Degrees-of-freedom method: kenward-roger
## Results are given on the log2 (not the response) scale.
## Confidence level used: 0.95
```

```
plot(thorax_emm,
      xlab = "model estimates, thorax length, log2 microM") +
  theme_bw() +
  theme(text = element_text(size = 16))
```

1.6.7.2 side note: Backtransforming in emmeans Like I mentioned emmeans can recognize the log2 transformation, so if you prefer the measures or plots in μm response scale.

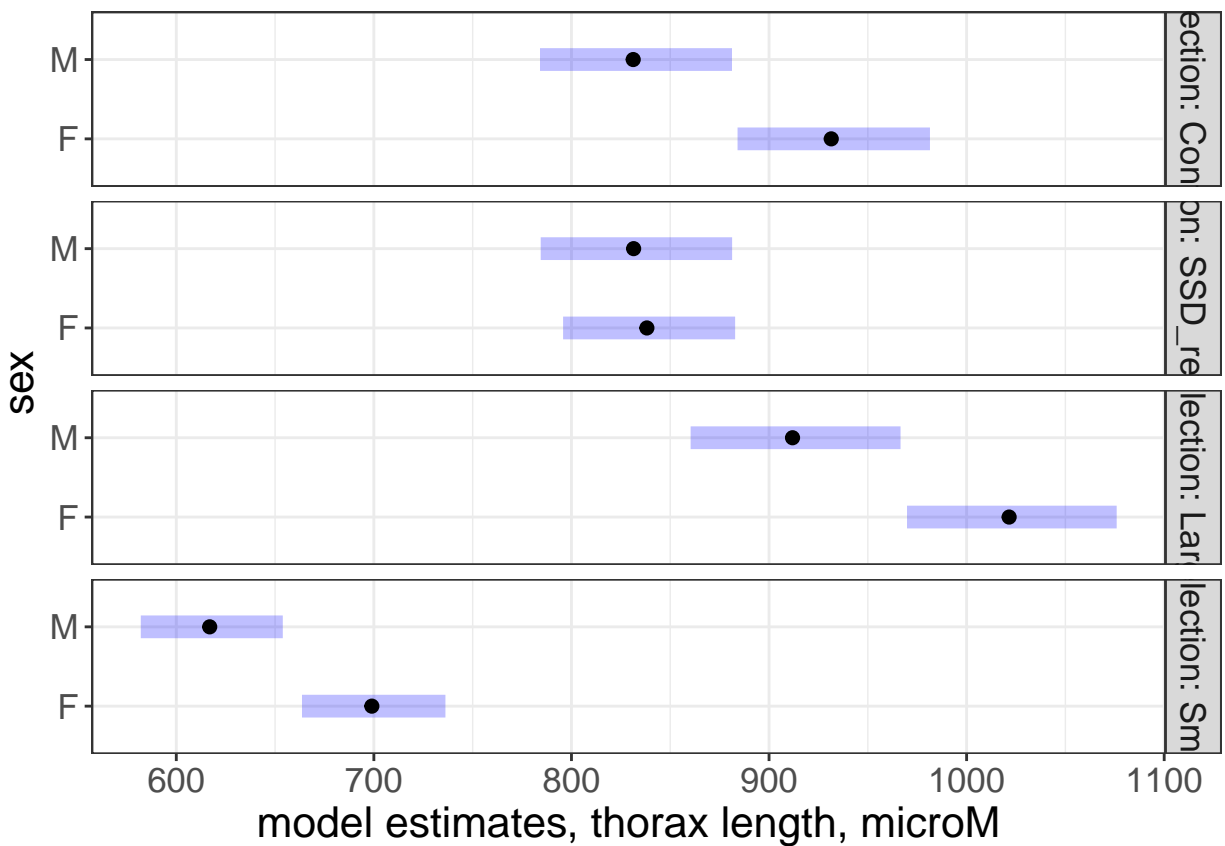
```
thorax_emm_response <- emmeans(mod1_thorax, specs = ~ sex | selection, type = "response")
```

```
thorax_emm_response
```

```
## selection = Control:
## sex response SE df lower.CL upper.CL
## F 931 17.3 3.88 884 981
## M 831 17.4 3.94 784 881
##
## selection = SSD_reversed:
## sex response SE df lower.CL upper.CL
## F 838 15.8 4.05 796 883
## M 831 17.5 4.02 784 881
##
## selection = Large:
## sex response SE df lower.CL upper.CL
## F 1022 19.2 4.05 970 1076
## M 912 19.2 4.02 860 967
##
## selection = Small:
## sex response SE df lower.CL upper.CL
## F 699 13.1 4.05 664 736
```

```
## M      617 13.0 4.02      582      654
##
## Results are averaged over the levels of: sampling
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
## Intervals are back-transformed from the log2 scale
```

```
plot(thorax_emm_response,
     xlab = "model estimates, thorax length, microM") +
  theme_bw() +
  theme(text = element_text(size = 16))
```



1.6.8 contrasts for sexual dimorphism

While it is not the hypothesis we are examining, for purposes of clarity of taking you through the steps, let's examine treatment specific patterns of sexual dimorphism. We will set this up as so:

```
thorax_vals <- emmeans(mod1_thorax,
  specs = ~ sex | selection)

SSD_contrasts_treatment <- pairs(thorax_vals)

SSD_contrasts_treatment
```

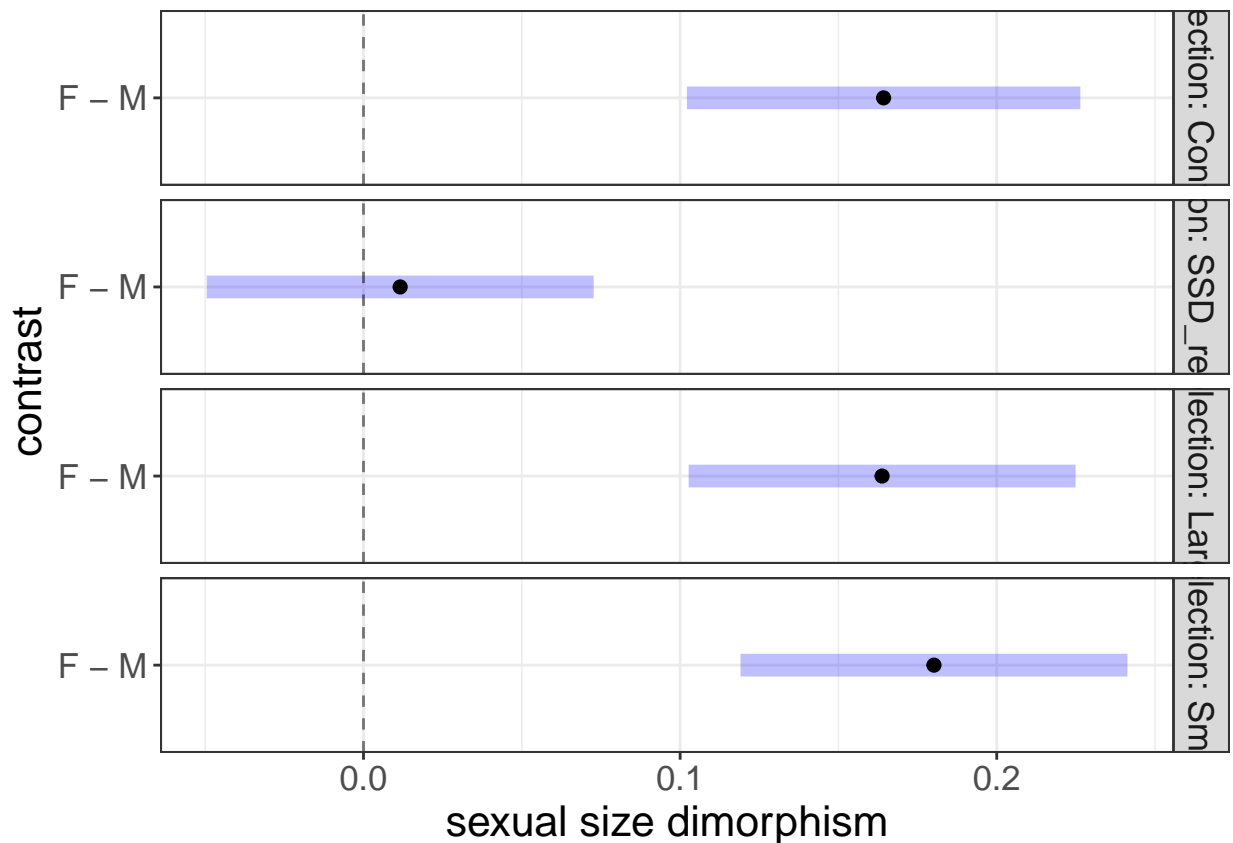
```
## selection = Control:
```

```
## contrast estimate      SE    df t.ratio p.value
## F - M      0.1643 0.0217 3.70   7.580 0.0020
##
## selection = SSD_reversed:
## contrast estimate      SE    df t.ratio p.value
## F - M      0.0116 0.0222 4.11   0.520 0.6290
##
## selection = Large:
## contrast estimate      SE    df t.ratio p.value
## F - M      0.1638 0.0222 4.11   7.360 0.0020
##
## selection = Small:
## contrast estimate      SE    df t.ratio p.value
## F - M      0.1802 0.0222 4.11   8.100 0.0010
##
## Results are averaged over the levels of: sampling
## Degrees-of-freedom method: kenward-roger
## Results are given on the log2 (not the response) scale.
```

```
confint(SSD_contrasts_treatment)
```

```
## selection = Control:
## contrast estimate      SE    df lower.CL upper.CL
## F - M      0.1643 0.0217 3.70   0.1021 0.2264
##
## selection = SSD_reversed:
## contrast estimate      SE    df lower.CL upper.CL
## F - M      0.0116 0.0222 4.11  -0.0495 0.0727
##
## selection = Large:
## contrast estimate      SE    df lower.CL upper.CL
## F - M      0.1638 0.0222 4.11   0.1027 0.2249
##
## selection = Small:
## contrast estimate      SE    df lower.CL upper.CL
## F - M      0.1802 0.0222 4.11   0.1191 0.2413
##
## Results are averaged over the levels of: sampling
## Degrees-of-freedom method: kenward-roger
## Results are given on the log2 (not the response) scale.
## Confidence level used: 0.95
```

```
plot(SSD_contrasts_treatment) +
  geom_vline(xintercept = 0, lty = 2, alpha = 0.5) +
  labs(x = "sexual size dimorphism") +
  theme_bw() +
  theme(text = element_text(size = 16))
```



1.6.9 The interaction contrast

But what we really want is to see how dimorphism changes in the selected treatments, relative to the controls. This is the interaction contrast (contrast of contrasts)

```
thorax_ssd <- emmeans(mod1_thorax, pairwise ~ sex*selection) # warning is letting you know these are n
thorax_ssd_contrasts <- contrast(thorax_ssd[[1]],
                                interaction = c(selection = "trt.vs.ctrl1", sex = "pairwise"),
                                by = NULL)
```

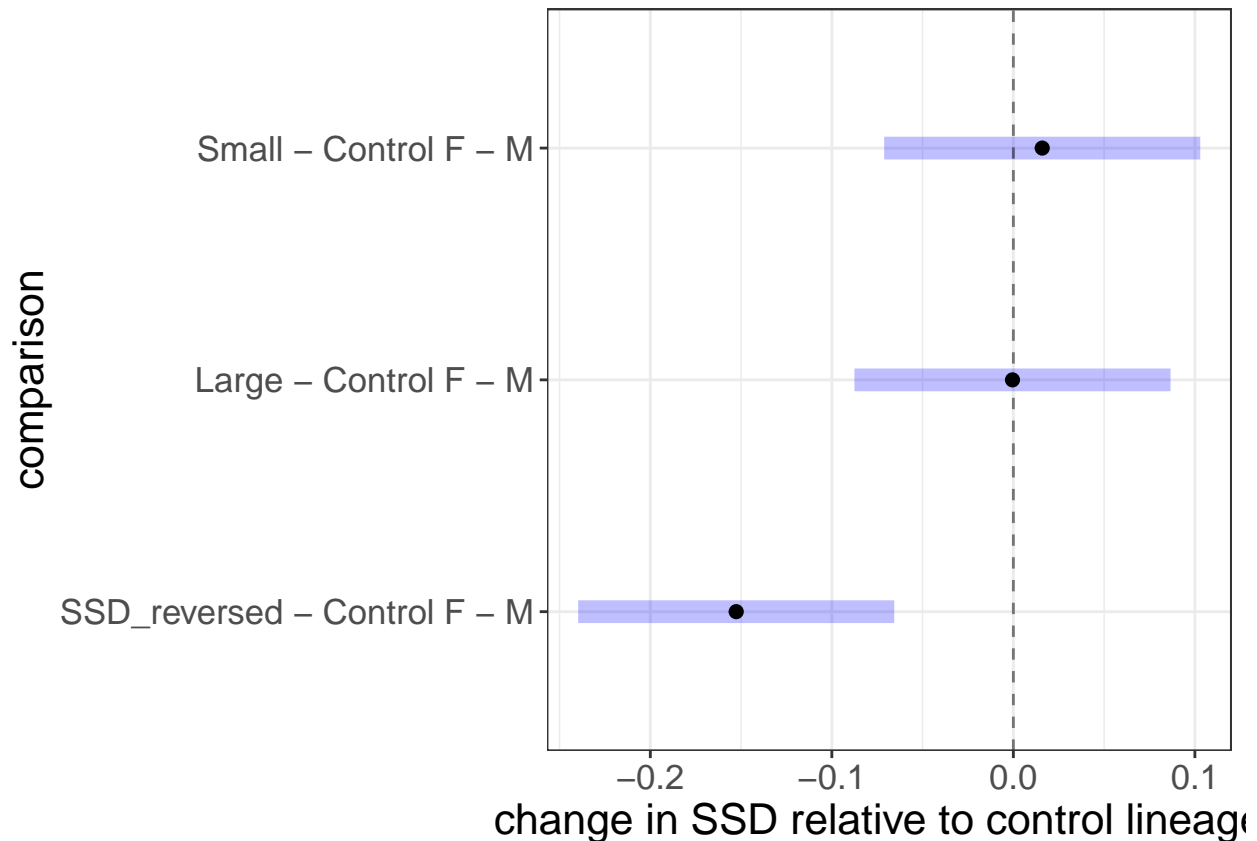
```
thorax_ssd_contrasts
```

```
## selection_trt.vs.ctrl1 sex_pairwise estimate SE df t.ratio p.value
## SSD_reversed - Control F - M -0.1527 0.031 3.9 -4.920 0.0080
## Large - Control F - M -0.0005 0.031 3.9 -0.020 0.9890
## Small - Control F - M 0.0159 0.031 3.9 0.510 0.6360
##
## Results are averaged over the levels of: sampling
## Degrees-of-freedom method: kenward-roger
## Results are given on the log2 (not the response) scale.
```

```
confint(thorax_ssd_contrasts)
```

```
## selection_trt.vs.ctrl1 sex_pairwise estimate SE df lower.CL upper.CL
## SSD_reversed - Control F - M -0.1527 0.031 3.9 -0.2397 -0.0656
## Large - Control F - M -0.0005 0.031 3.9 -0.0875 0.0866
## Small - Control F - M 0.0159 0.031 3.9 -0.0712 0.1030
##
## Results are averaged over the levels of: sampling
## Degrees-of-freedom method: kenward-roger
## Results are given on the log2 (not the response) scale.
## Confidence level used: 0.95
```

```
plot(thorax_ssd_contrasts) +
  geom_vline(xintercept = 0, lty = 2, alpha = 0.5) +
  labs(x = "change in SSD relative to control lineages", y = "comparison") +
  theme_bw() +
  theme(text = element_text(size = 16))
```



1.7 links for other tutorials on contrasts

https://rpubs.com/timflutre/tuto_contrasts586f394d8eb84f84b1baaf56ffb6b47f.html

https://rstudio-pubs-static.s3.amazonaws.com/65059_