Mixed Effects Models (Intro): Practice Problems

Crispin Jordan

30/11/2021

Please note that, given strong overlap between analyses presented here and those in previous documents, this document focuses explanations on aspects related to mixed effects models, per se. **Please see earlier documents related to, say, analysis by 1-Factor GLM, for explanation of analyses not strictly related to mixed-effects models.**

Question 1: Analysis of Height in Yellow Monkeyflower

Let's begin by importing the data:

```
mim <- read.table("HeightStress.csv", header = TRUE, sep = ',')
```

(I named the dataframe, `mim`, in honour of the species being studied, *Mimulus guttatus*.)

Before going further, let's check whether our variables are of the desired type:

```
str(mim)
```

```
## 'data.frame':    321 obs. of  3 variables:
##  $ Treatment: chr  "Control" "Control" "Control" "Control" ...
##  $ Height   : int  70 65 65 65 64 66 64 68 65 78 ...
##  $ Genotype : chr  "3 a" "9 a" "27 a" "1 a" ...
```

We see that `Treatment` and `Genotype` are type `chr` (i.e., character). (Please note that other versions of **R**, different from mine, sometimes recognize these variables as type `Factor`) Let's convert them to type, `Factor`:

```
mim$Treatment <- factor(mim$Treatment)
mim$Genotype <- factor(mim$Genotype)
```

Now, we'll double-check that they are the desired type:

```
str(mim)
```

```
## 'data.frame':    321 obs. of  3 variables:
##  $ Treatment: Factor w/ 3 levels "Control","H2Ostress",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Height   : int  70 65 65 65 64 66 64 68 65 78 ...
##  $ Genotype : Factor w/ 19 levels "1 a","11 a","13 a",..: 12 19 10 1 12 17 18 5 7 8 ...
```

Looks good!

Part (a)

Let's look at the top of the dataframe:

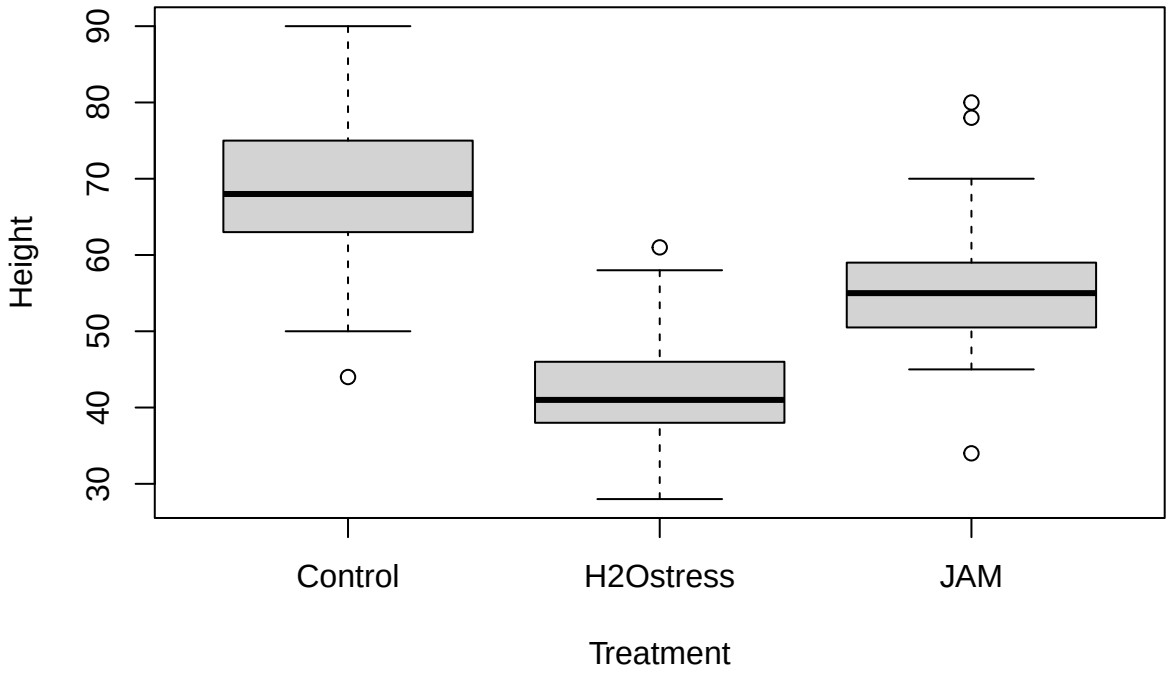
```
head(mim)
```

```
##   Treatment Height Genotype
## 1   Control     70      3 a
## 2   Control     65      9 a
```

```
## 3   Control     65     27 a
## 4   Control     65      1 a
## 5   Control     64      3 a
## 6   Control     66      5 a
```

We see three columns: `Treatment`, `Height` and `Genotype`. Our hypothesis aim to test whether the environment (specified by `Treatment`) affects `Height`. This thinking implies we expect `Height` to *depend* on `Treatment`; i.e., `Height` is the dependent variable and we will place it to the left of the tilda (`~`) in our code to plot the data as a boxplot:

```
boxplot(Height ~ Treatment, data = mim)
```



Recall that the 'boxes' in the boxplots show the range of the middle 50% of the data. Based on this we see little overlap between each distribution, suggesting that we might expect differences in `Height` between all three levels of `Treatment`. Also, note that the boxplots are relatively symmetrical, suggesting that the data may meet the assumptions of normality. However, this might change when we take a closer look at the data.

Part (b)

We will select `Treatment` as a **fixed** effect: we are specifically interested differences among the levels of `Treatment`. Moreover, and as a more subtle point, we expect that the information from one level of `Treatment` need not inform the value of another level of `Treatment`.

Part (c)

We will select `Genotype` as a **random** effect. We are not interested in the values of individual levels of this factor. (By 'level' of `Genotype`, we mean the identity of each genotype.) Moreover, the levels of this treatment were selected randomly from a population, and we aim to model our data in a way that allows us to make inferences about the population from which these genotypes come. Note that when we specify `Genotype` as a random effect, information from all levels of `Genotype` will be used to inform estimates of all other levels of `Genotype`.

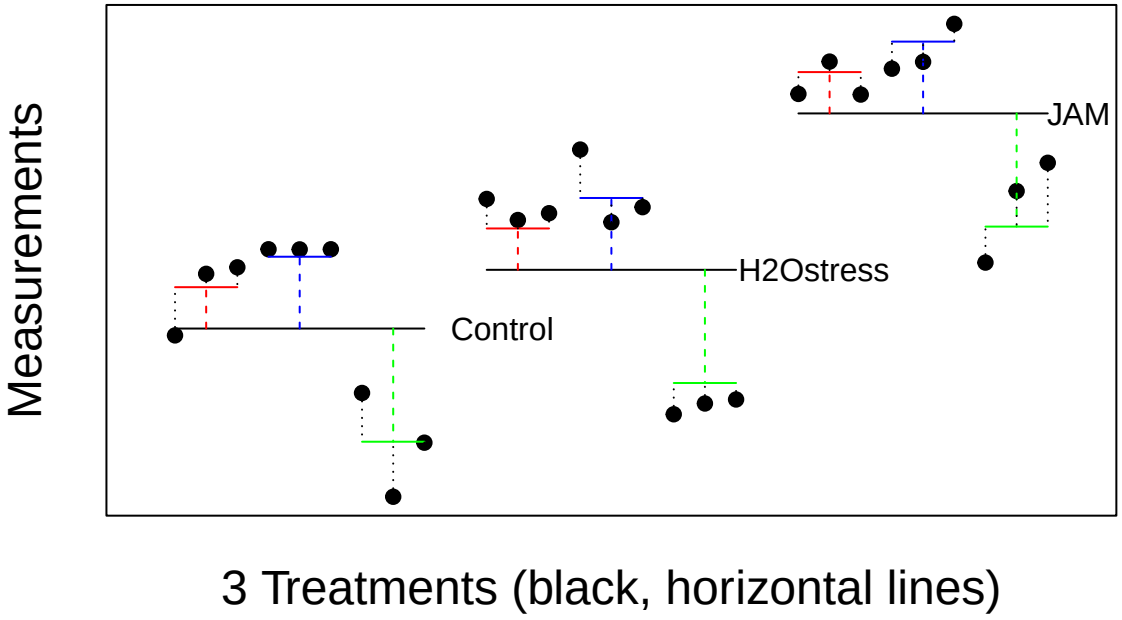
Part (d)

We will model the data in `Genotype` for two reasons:

- First, note that we often have more than one measurement of `Height` for a given genotype within a

level of `Treatment`. If we did not account for this fact in our analysis, we'd violate the assumption is independence (i.e., introduce pseudo-replication). By modelling `Genotype`, we avoid violating the assumption of independence. This is the point that has been emphasized in our introductory videos to mixed effects models.

- Our second reason is something we've not discussed in our introductory videos for mixed effects models (but we will do so in videos with a more advanced perspective). (*Consider the explanation here as 'bonus' for this question.*) We'll briefly introduce our second reason here as foreshadowing for a more thorough discussion in the future. In short, our analysis of `Genotype`, below, will allow us to model effects of `Genotype` *across* levels of `Treatment`, in addition to accounting for the multiple measurements per `Genotype` level per `Treatment` level to account for pseudo-replication. There are multiple ways of modeling how levels of `Genotype` can vary among levels of `Treatment`. Our analysis, below, will use the simplest approach: we will assume that an estimate of a given level of `Genotype` will deviate from the mean of each level of `Treatment` by the same amount, but the size of this deviation will vary among levels of `Genotype`. We illustrate this in the figure, below. This figure provides an imaginary illustration of the data in this experiment: we see three long, black horizontal lines, which represent the mean values of each of the three levels of `Treatment` (Control, H2Ostress and JAM). For each `Treatment` level, note three smaller, horizontal lines in three colours (red, blue, green): Each colour represents information for a given genotype (i.e., level of `Genotype`): the difference between each coloured horizontal line and the black line for a `Treatment` level represents the effect of a given genotype on `Height`. Notice that the blue horizontal line always has the same deviation from the mean of the three `Treatment` levels (this is true for all three colours): **this illustrates that we're modeling the effect of `Genotype` in a way that assumes that a given genotype (level of `Genotype`) always affects `Height` in the same way and by the same amount for all three levels of `Treatment`.** Finally, note that we have 3 measurements per `Genotype` level per `Treatment` level: the differences between these measurments (i.e., the 'points' on the plot) and the estimated effect of each `Genotype` level represents the residual variation.



Part (e)

We will use the `lmer()` function in the `lme4` library to run our analysis. Note that this function, on its own, does not provide p-values (for very good reasons that are beyond the scope of this exercise). There are multiple ways of calculating p-value; my preference is to use **parametric bootstrapping**, implemented by, for example, the `PBmodcomp()` function in the `pbkrtest` library. However, this approach deviates somewhat from what we're used to seeing so far and requires further explanation. Therefore, we'll instead use the `lmerTest` library to calculate p-values in a fashion that is more familiar to you (and this is also the approach

we use in introductory videos for mixed effects models). Note however, that the approaches used by `lmerTest` can be anti-conservative (i.e., can yield p-values smaller than they should be).

With all of this in mind, let's open the required libraries and create our model:

```
library(lme4)
```

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

```
library(lmerTest)
```

```
##
## Attaching package: 'lmerTest'
```

```
## The following object is masked from 'package:lme4':
##
##     lmer
```

```
## The following object is masked from 'package:stats':
##
##     step
```

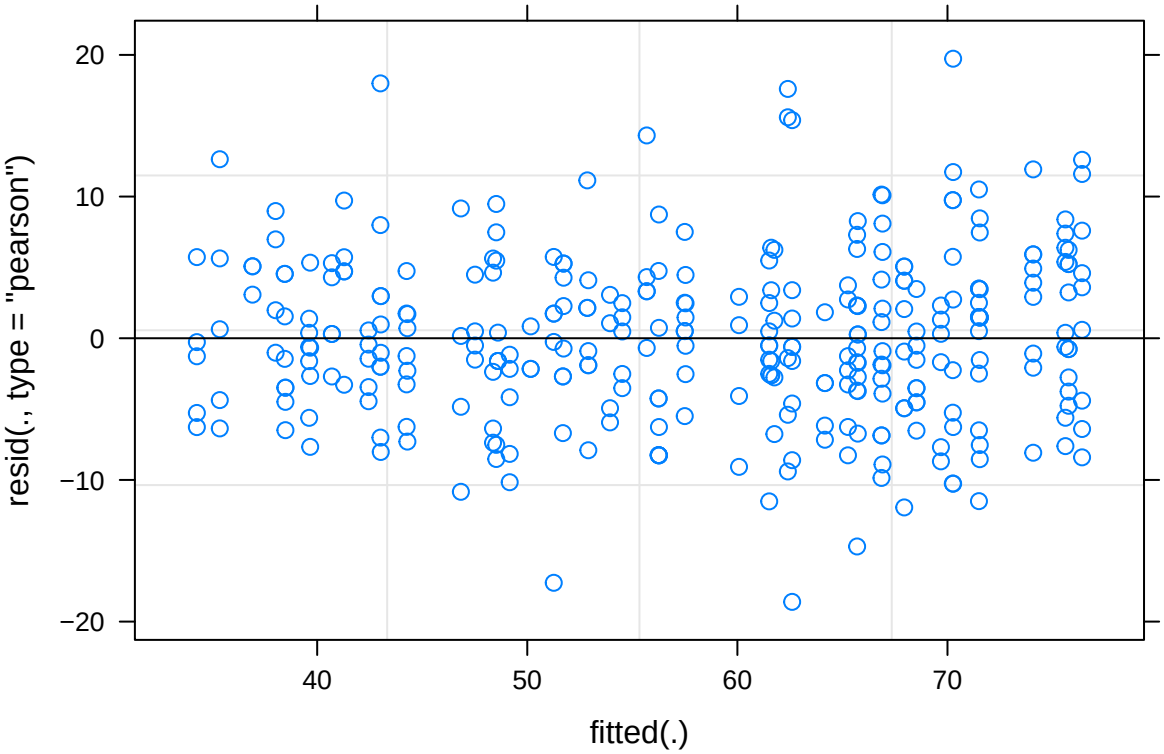
```
mim.lmer <- lmer(Height ~ Treatment + (1|Genotype), data = mim)
```

This model specifies `Height` and the dependent variable; for our independent variable, we model `Treatment` as a fixed effect (Factor) and `Genotype` as a random effect (also a Factor; see top of this document). Note that we modeled `Genotype` using the notation, `(1|Genotype)`: this notation reflects our assumption that each genotype affects `Height` the same way (and by the same amount) for all treatment levels, as illustrated in the Figure, above. There are alternative ways of modeling `Genotype` as a random effect, which we will address in more detailed discussion of mixed effects models. For now, we're happy to just use `(1|Genotype)`.

We can check our residual in a similar way as we did when using the `lm()` function. However, the output provides only a single plot (not four) when we submit the command, `plot(object.with.my.model.results)`.

We assess the assumption of equal variance like this:

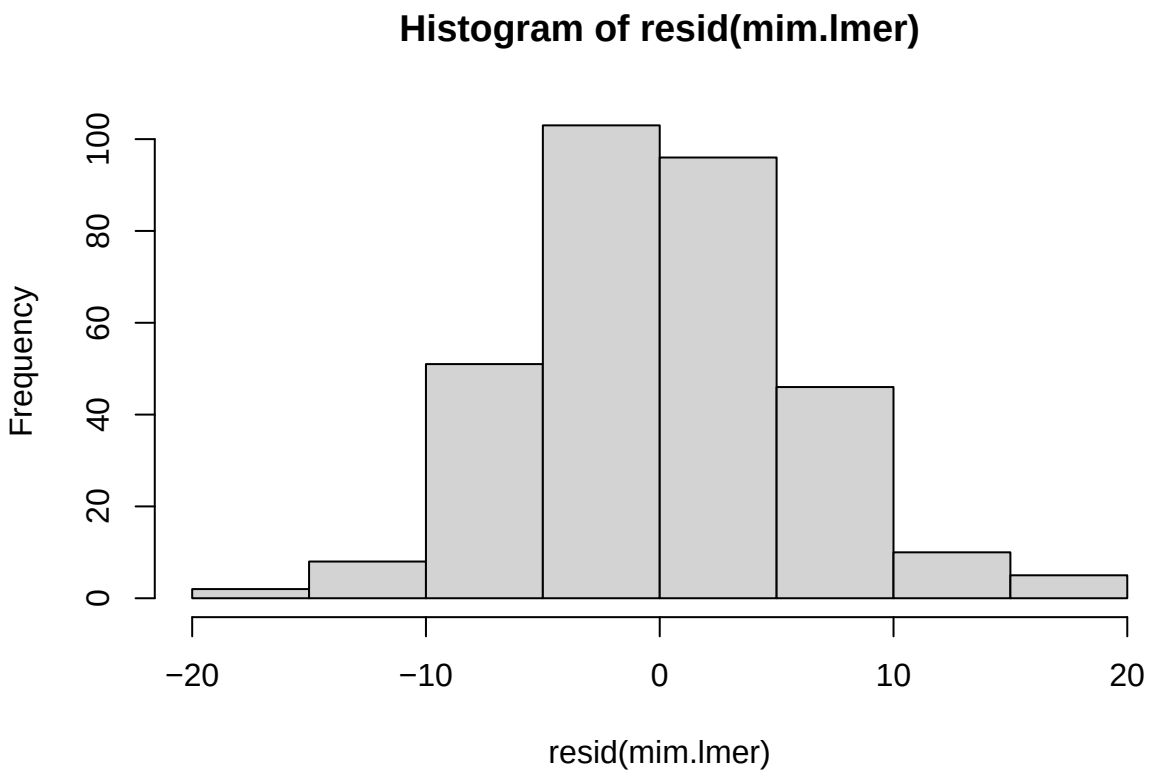
```
plot(mim.lmer)
```



This plot looks very nice. Ideally, we want this plot to look like a 'cloud'; i.e., with no obvious trends. This seems to be what we see. There may be a tiny tendency for the residuals to become more variable (i.e., more spread out, vertically) as we move from left to right (indicating that we violate the assumption of equal variance). However, if this occurs at all, the effect is slight. Moreover, log-transformation has little effect (try this yourself). **We would report the assumptions we checked, how we checked them, and whether they were met.** We'll proceed assuming the data meet the assumption of equal variance.

We can plot ths distribution of residual like this:

```
hist(resid(mim.lmer))
```



This looks beautiful. We're happy that the data meet the assumptions (recalling that the data meet the assumption of randomization and our analysis accounts for non-independence).

Part (f)

Let's look at our p-values. Having loaded the `lmerTest` library, we can check the p-values like this (note that this approach, using `lmerTest`, accounts for unbalanced data):

```
anova(mim.lmer)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##           Sum Sq Mean Sq NumDF  DenDF F value    Pr(>F)
## Treatment  41570   20785     2 300.48  572.78 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that our p-value is tiny, indicating strong evidence for an effect of `Treatment` on `Height`. **We would report this p-value, along with the F-value and the two types of degrees of freedom (`NumDF`, `DenDF`).**

Now, our job is to examine the effect sizes and obtain p-values for comparisons among `Treatment` levels. We can do this using `emmeans`, as we have in previous analyses using `lm()`.

Open the `emmeans` library:

```
library(emmeans)
```

Now, we want to calculate the means for each level of `Treatment`:

```
mim.emmeans <- emmeans(mim.lmer, "Treatment")
```

And we obtain our means, their SE's and 95% CI's here (**note that we would report all of this output**):

```
mim.emmeans
```

```
##  Treatment emmean   SE   df lower.CL upper.CL
##  Control     69.0 1.18 21.9     66.5     71.4
##  H2Ostress   41.7 1.25 27.7     39.2     44.3
##  JAM         55.0 1.25 28.3     52.4     57.5
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
```

We can perform comparisons among `Treatment` levels in a way that accounts for multiple comparisons (Tukey method) like this:

```
pairs(mim.emmeans)
```

```
##  contrast            estimate    SE  df t.ratio p.value
##  Control - H2Ostress     27.2 0.814 300  33.456 <.0001
##  Control - JAM           14.0 0.825 301  16.977 <.0001
##  H2Ostress - JAM        -13.2 0.923 300 -14.338 <.0001
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 3 estimates
```

We would report all of the output here (reporting `df` may not be necessary). What do we learn?

- We have strong evidence ($p < 0.0001$) for all differences among `Treatment` levels.
- We note that `H2Ostress` decreases height relative to the `Control` group by about twice as much that the `JAM` treatment does. This may be biologically interesting. Also, both `H2Ostress` and `JAM` reduce height by degree that may be biologically meaningful (by about 40% and 20%, respectively).

Finally, we can determine 95% CI's for these effect sizes:

```
confint(pairs(mim.emmeans))
```

```
##  contrast            estimate    SE  df lower.CL upper.CL
##  Control - H2Ostress     27.2 0.814 300     25.3     29.2
##  Control - JAM           14.0 0.825 301     12.1     15.9
##  H2Ostress - JAM        -13.2 0.923 300    -15.4    -11.1
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
## Conf-level adjustment: tukey method for comparing a family of 3 estimates
```

These 95% CI's indicate that we've estimated effect sizes fairly precisely. For example, the 95% CI's for the difference between `Control` and `H2Ostress` equal 25.3 and 29.2. Given that the mean of the `Control` group equals 69.0, these 95% CI's imply that `H2Ostress` plausibly decreases `Height` by 36% (i.e., `25.3 / 69.0`) to 42% (i.e., 29.2 / 69.0). **We would report this conclusion, and do similarly for the `JAM` level of `Treatment`.**

BONUS

Before we complete this question, let's look at the summary output from our model - there's something special to notice.

```
summary(mim.lmer)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: Height ~ Treatment + (1 | Genotype)
##    Data: mim
##
## REML criterion at convergence: 2101.5
```

```
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0888 -0.6139 -0.0850  0.6203  3.2755
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Genotype (Intercept) 21.57    4.644
##  Residual             36.29    6.024
## Number of obs: 321, groups:  Genotype, 19
##
## Fixed effects:
##                     Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)          68.9644     1.1751  22.0119   58.69   <2e-16 ***
## TreatmentH2Ostress -27.2422     0.8142 300.4328  -33.46   <2e-16 ***
## TreatmentJAM       -14.0031     0.8248 300.6445  -16.98   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) TrtH2O
## TrtmntH2Ost -0.256
## TreatmntJAM -0.253  0.365
```

First, notice that this output is very similar to what we see for a standard 1-Factor GLM implemented using `lm()`. However, notice that we now have estimates under the heading, `Random effects:`. We see an estimate of variation associated with `Genotype` and also associated with `Residual`. Examine the final column in this section, titled, `Std.Dev`: this column provides the estimated variation in our data explained by `Genotype` and `Residual` variation expressed as a standard deviation. These values are 4.644 and 6.024 for `Genotype` and `Residual`, respectively. This implies that there is almost as much variation among genotypes as there is within `Genotype/Treatment` combinations (i.e., residual variation). This is biologically interesting, because it implies that, potentially, `Genotype` levels (i.e., different genotypes) may have quite varied effects on `Height` (i.e., this may provide evidence for genetic variation for `Height`). Cool, eh? Biologists who study the (quantitative) genetics of traits use this kind of information very frequently to get a sense of the proportion of variation in a trait can be explained by genetics (i.e., 'heritability', broadly speaking). The point here is that, although our focal analysis called us to model `Genotype` as a random effect to control for pseudo-replication, **for some (biological) questions we may be specifically interested in quantifying the amount of variation explained by a random effect, and mixed effects models can help us here** - cool, right?!

Question 2

Before, we begin this analysis, let's consider the experimental design. This dataset includes multiple measurements from individual subjects because protein expression was measures more than once per mouse. However, the multiple measurements per mouse arose from different tissues: i.e., each tissue within a given mouse was measured only once. This experimental design does not violate the assumption of independence if were only interested in making comparisons among tissues (i.e., if 'tissue' was the only independent variable in our analysis). However, this is not true in our case. Instead, the researchers wanted to know whether the effect of tissue on protein expression differed over time: i.e., they wanted to look for an interaction between tissue and time. Note that each mouse was killed at a specific time point and we have multiple measurements per mouse at a given time. This means that if we include time in the analysis, we need to account for non-independence that arises due to having multiple measurments per mouse per time period. Overall, the point here is that we'll analyze these data with a mixed effects model because we want to account for the non-indepedence that arises when we include time in the model.

Let's import the data:

```
mys <- read.table("MysteryData.csv", header = TRUE, sep = ',')
```

Now, let's check our variables:

```
str(mys)
```

```
## 'data.frame':    54 obs. of  4 variables:
##  $ tissue : chr  "brain" "brain" "brain" "brain" ...
##  $ SMN_exp: num  1.202 1.282 1.122 1.064 0.984 ...
##  $ mouse.c: chr  "M1 5" "M2 5" "M3 5" "M1 10" ...
##  $ age.fac: chr  "Five" "Five" "Five" "Ten" ...
```

Note three variables listed as type `chr` (character), where we'd like them to be `Factor`: `tissue`, `mouse.c` (i.e., mouse subject identity) and `age.fac` (age). Notice that we will model age as a category / `Factor`, rather than as a continuous variable (i.e., covariate). Let's convert each to a `Factor` and check that our efforts have worked (note that this step may not be necessary for you, depending on your version of **R** and how it interprets data imported with the `read.table()` function).

```
mys$tissue <- factor(mys$tissue)
mys$mouse.c <- factor(mys$mouse.c)
mys$age.fac <- factor(mys$age.fac)
#Check our work:
str(mys)
```

```
## 'data.frame':    54 obs. of  4 variables:
##  $ tissue : Factor w/ 6 levels "brain","heart",..: 1 1 1 1 1 1 1 1 1 6 ...
##  $ SMN_exp: num  1.202 1.282 1.122 1.064 0.984 ...
##  $ mouse.c: Factor w/ 9 levels "M1 10","M1 15",..: 3 6 9 1 4 7 2 5 8 3 ...
##  $ age.fac: Factor w/ 3 levels "Fifteen","Five",..: 2 2 2 3 3 3 1 1 1 2 ...
```

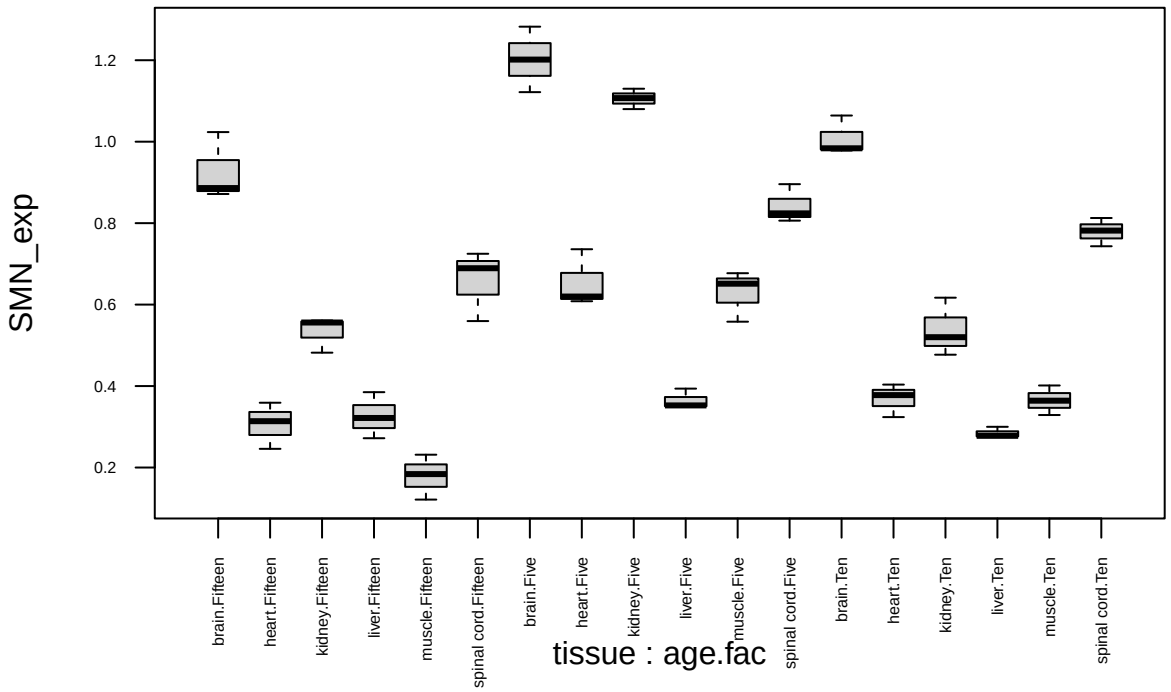
Excellent!

Let's carry on.

Part (a)

We're asked to plot the data in a boxplot. Our hypothesis is that `tissue` and age (`age.fac`) affect gene expression (column, `SMN_exp`). Therefore, we conclude that `SMN_exp` is the *dependent* variable (and gos to the left of the `~` in our model). Also, `tissue` and `age.fac` are the independent variables. If we ignore the effect of `mouse.c`, we can plot our data in the same way that we would for a standard 2-Factor glm:

```
boxplot(SMN_exp ~ tissue*age.fac, data = mys, cex.axis = 0.5, las = 2)
```



(In case you are wondering, the `las = 2` option rotated the orientation of the axis labels; `cex.axis = 0.5` made these labels only half their usual size.)

Hmmm... There's a lot to digest here. But, looking carefully, we might expect to find an interaction between `tissue` and `age.fac`. For example, compare the values of `SMN_exp` in the liver vs. muscle at times 'five' vs. 'fifteen': liver tends to be higher than muscle in the former but lower in the latter.

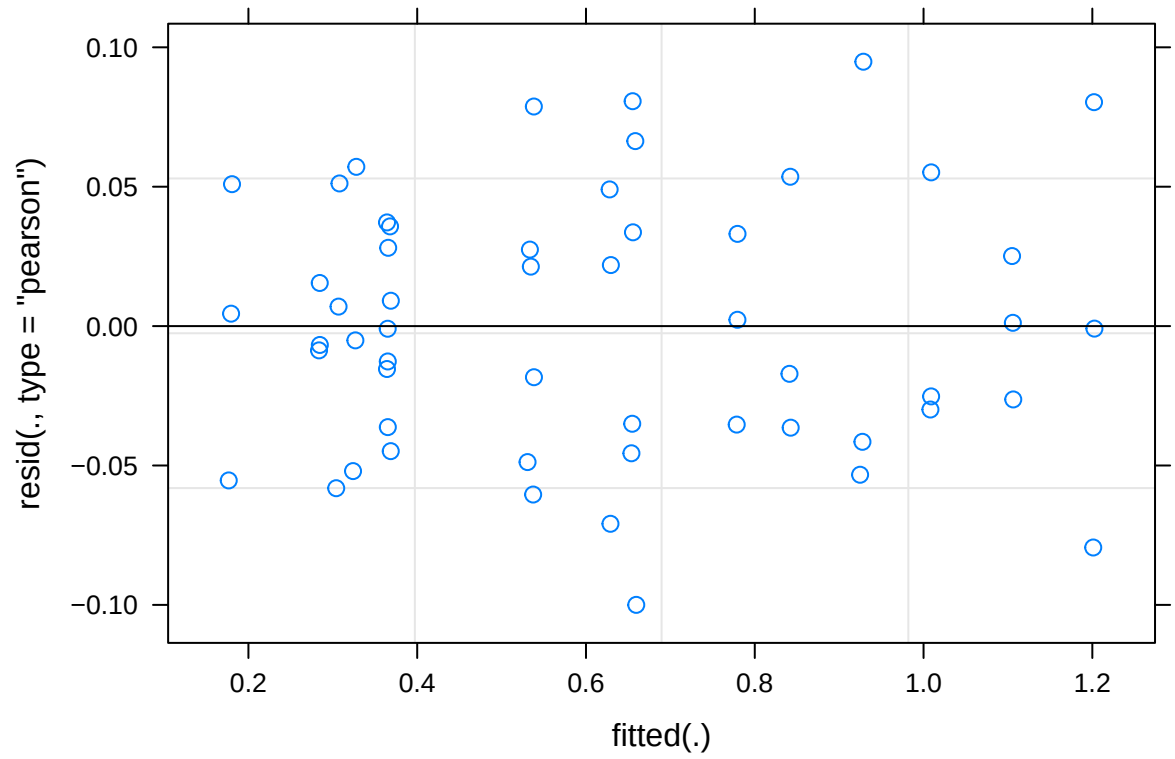
Part (b)

Let's model our data. Following our logic for the boxplot, above, we know that `SMN_exp` is the dependent variable, and `tissue` and `age.fac` will be our independent variables. Will we be able to model an interaction between `tissue` and `age.fac`? We answer that question in the same way we would for a normal 2-Factor glm: we ask ourselves whether we have at least 2 independent measurements in each combination of our two Factors (`tissue` and `age.fac`). In our case, we do, because the data include at least 2 mice per level of `age.fac` (and each mouse yield 1 measurement per level of `tissue`); therefore, our data meet the requirements to model an interaction. Finally, we'll model `mouse.c` as a random effect:

```
#Just a reminder to open the relevant libraries; see Question 1:
library(lme4)
library(lmerTest)
#Formulate our model:
mys.lmer <- lmer(SMN_exp ~ tissue*age.fac + (1|mouse.c), data = mys)
```

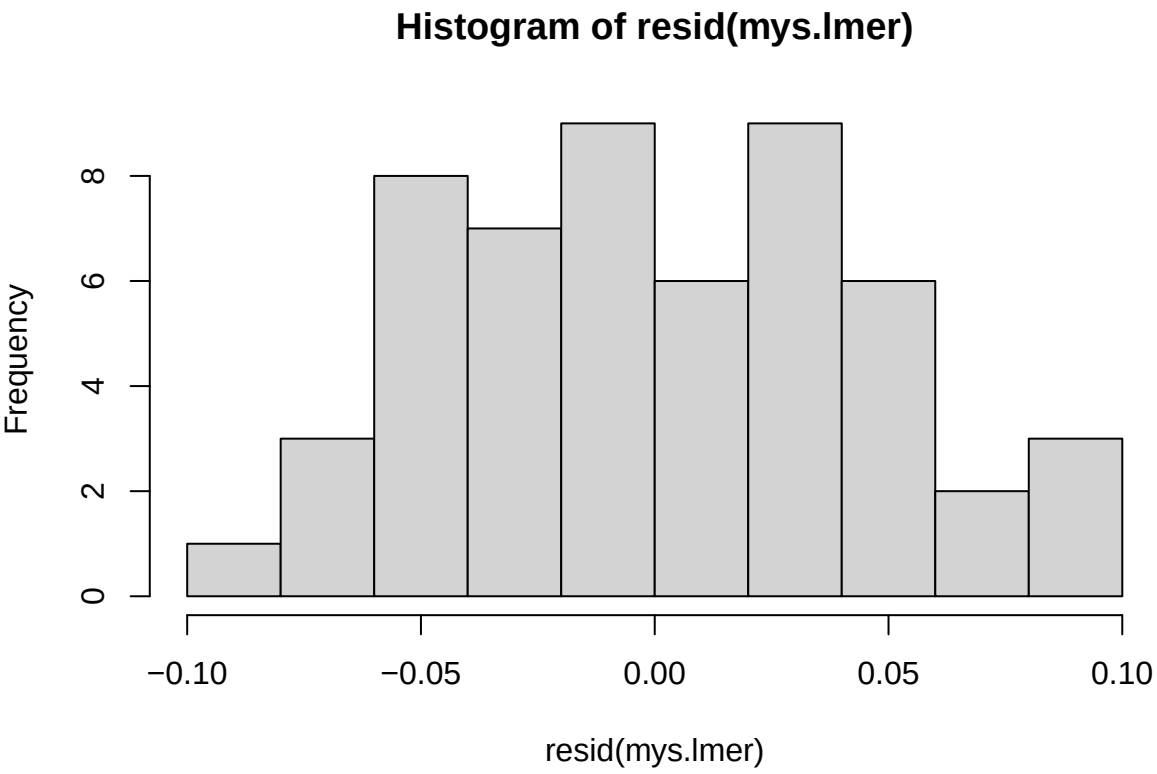
Now, let's check the assumptions:

```
plot(mys.lmer)
```



This plot looks ideal: the residuals form a nice 'cloud' with no apparent pattern.

```
hist(resid(mys.lmer))
```



This plot also looks great.

We're satisfied that the data meet the assumptions of the model. (Recall that our analysis accounts for

known sources of non-independence; also, to me knowledge the mice were allocated randomly to levels of `age.fac`; these two observations allow us to conclude the data also meet the assumptions of independence and randomization.)

With this in mind, let's look at our p-values:

```
anova(mys.lmer)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##                Sum Sq Mean Sq NumDF   DenDF F value    Pr(>F)
## tissue         3.4434 0.68869     5 30.0003 220.501 < 2.2e-16 ***
## age.fac        0.9111 0.45556     2  6.0002 145.861 8.182e-06 ***
## tissue:age.fac 0.3828 0.03828    10 30.0003  12.255 4.370e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(As a reminder, I prefer to implement `PBmodcomp` to calculate p-values, but the p-values shown here are likely to be calculated in a similar fashion to many other software packages.)

We find a very small p-value for the interaction, suggesting strong evidence for an interaction between `tissue` and `age.fac` for protein expression (`SMN_exp`). As a result, we focus our analysis not on intrpreting this interaction and we do not attempt to interpret the main effect of `tissue` and `age.fac`, following the exact same logic as we learned for standard 2-Factor GLMs (implemented with `lm()`). **We would report the results in the table above, as we would for a standard 2-Factor glm.**

We now wish to better understand the interaction. We will examine effect sizes to do so. As usual, we'll use `emmeans`. There are many ways to interrogate the effect sizes. We'll examine differences among `tissue` levels (e.g., liver vs. muscle), and we'll do so separately for each level of `age.fac`:

```
mys.emmeans <- emmeans(mys.lmer, "tissue", by = "age.fac")
mys.emmeans
```

```
## age.fac = Fifteen:
##  tissue      emmean     SE df lower.CL upper.CL
##  brain        0.927 0.0324 36    0.861    0.993
##  heart        0.306 0.0324 36    0.241    0.372
##  kidney       0.533 0.0324 36    0.467    0.599
##  liver        0.326 0.0324 36    0.260    0.392
##  muscle       0.179 0.0324 36    0.113    0.245
##  spinal cord  0.658 0.0324 36    0.592    0.724
##
## age.fac = Five:
##  tissue      emmean     SE df lower.CL upper.CL
##  brain        1.202 0.0324 36    1.136    1.268
##  heart        0.655 0.0324 36    0.589    0.720
##  kidney       1.106 0.0324 36    1.040    1.171
##  liver        0.365 0.0324 36    0.299    0.431
##  muscle       0.629 0.0324 36    0.563    0.695
##  spinal cord  0.842 0.0324 36    0.776    0.908
##
## age.fac = Ten:
##  tissue      emmean     SE df lower.CL upper.CL
##  brain        1.009 0.0324 36    0.943    1.074
##  heart        0.368 0.0324 36    0.303    0.434
##  kidney       0.538 0.0324 36    0.472    0.604
##  liver        0.284 0.0324 36    0.219    0.350
##  muscle       0.365 0.0324 36    0.299    0.431
```

```
##  spinal cord  0.779 0.0324 36    0.713    0.845
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
```

We would report these values.

Now, let's compare protein expression among tissues (by age):

```
pairs(mys.emmeans)
```

```
## age.fac = Fifteen:
##  contrast             estimate     SE df t.ratio p.value
##  brain - heart         0.62080 0.0456 30  13.605 <.0001
##  brain - kidney        0.39406 0.0456 30   8.636 <.0001
##  brain - liver         0.60085 0.0456 30  13.168 <.0001
##  brain - muscle        0.74815 0.0456 30  16.396 <.0001
##  brain - spinal cord   0.26915 0.0456 30   5.898 <.0001
##  heart - kidney       -0.22674 0.0456 30  -4.969 0.0003
##  heart - liver        -0.01995 0.0456 30  -0.437 0.9978
##  heart - muscle        0.12735 0.0456 30   2.791 0.0868
##  heart - spinal cord  -0.35165 0.0456 30  -7.706 <.0001
##  kidney - liver        0.20680 0.0456 30   4.532 0.0011
##  kidney - muscle       0.35409 0.0456 30   7.760 <.0001
##  kidney - spinal cord -0.12490 0.0456 30  -2.737 0.0971
##  liver - muscle        0.14730 0.0456 30   3.228 0.0325
##  liver - spinal cord  -0.33170 0.0456 30  -7.269 <.0001
##  muscle - spinal cord -0.47899 0.0456 30 -10.497 <.0001
##
## age.fac = Five:
##  contrast             estimate     SE df t.ratio p.value
##  brain - heart         0.54719 0.0456 30  11.992 <.0001
##  brain - kidney        0.09619 0.0456 30   2.108 0.3104
##  brain - liver         0.83686 0.0456 30  18.340 <.0001
##  brain - muscle        0.57301 0.0456 30  12.557 <.0001
##  brain - spinal cord   0.35992 0.0456 30   7.888 <.0001
##  heart - kidney       -0.45101 0.0456 30  -9.884 <.0001
##  heart - liver         0.28967 0.0456 30   6.348 <.0001
##  heart - muscle        0.02582 0.0456 30   0.566 0.9925
##  heart - spinal cord  -0.18728 0.0456 30  -4.104 0.0036
##  kidney - liver        0.74068 0.0456 30  16.232 <.0001
##  kidney - muscle       0.47682 0.0456 30  10.450 <.0001
##  kidney - spinal cord  0.26373 0.0456 30   5.780 <.0001
##  liver - muscle       -0.26386 0.0456 30  -5.782 <.0001
##  liver - spinal cord  -0.47695 0.0456 30 -10.452 <.0001
##  muscle - spinal cord -0.21309 0.0456 30  -4.670 0.0008
##
## age.fac = Ten:
##  contrast             estimate     SE df t.ratio p.value
##  brain - heart         0.64032 0.0456 30  14.033 <.0001
##  brain - kidney        0.47072 0.0456 30  10.316 <.0001
##  brain - liver         0.72444 0.0456 30  15.876 <.0001
##  brain - muscle        0.64389 0.0456 30  14.111 <.0001
##  brain - spinal cord   0.22952 0.0456 30   5.030 0.0003
##  heart - kidney       -0.16959 0.0456 30  -3.717 0.0097
##  heart - liver         0.08412 0.0456 30   1.843 0.4548
```

```
##  heart - muscle        0.00358 0.0456 30   0.078 1.0000
##  heart - spinal cord  -0.41080 0.0456 30  -9.003 <.0001
##  kidney - liver        0.25371 0.0456 30   5.560 0.0001
##  kidney - muscle       0.17317 0.0456 30   3.795 0.0080
##  kidney - spinal cord -0.24121 0.0456 30  -5.286 0.0001
##  liver - muscle       -0.08054 0.0456 30  -1.765 0.5021
##  liver - spinal cord  -0.49492 0.0456 30 -10.846 <.0001
##  muscle - spinal cord -0.41438 0.0456 30  -9.081 <.0001
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 6 estimates
```

(Note that this approach does not strictly account for all multiple comparisons, because the Tukey method is applied separately for each level of `age.fac` (14 comparisons) rather than for all 3*14 comparisons. This is not ideal, and I'm looking into an approach to address this within `emmeans`. Of course, this is not an issue if we do not control for multiple comparisons (the preferred approach for some scientists), by setting `adjust = "none"` in the `pairs()` function.)

There's a lot going on here. We'd report these results as usual (`estimate`, `SE`, `df`, `t.ratio`, `p.value` for each contrast). But, we may get a better sense of what's going on when we add confidence intervals:

```
confint(pairs(mys.emmeans))
```

```
## age.fac = Fifteen:
##  contrast              estimate     SE df lower.CL upper.CL
##  brain - heart         0.62080 0.0456 30   0.4820   0.7596
##  brain - kidney        0.39406 0.0456 30   0.2553   0.5328
##  brain - liver         0.60085 0.0456 30   0.4621   0.7396
##  brain - muscle        0.74815 0.0456 30   0.6094   0.8869
##  brain - spinal cord   0.26915 0.0456 30   0.1304   0.4079
##  heart - kidney       -0.22674 0.0456 30  -0.3655  -0.0880
##  heart - liver        -0.01995 0.0456 30  -0.1587   0.1188
##  heart - muscle        0.12735 0.0456 30  -0.0114   0.2661
##  heart - spinal cord  -0.35165 0.0456 30  -0.4904  -0.2129
##  kidney - liver        0.20680 0.0456 30   0.0680   0.3456
##  kidney - muscle       0.35409 0.0456 30   0.2153   0.4929
##  kidney - spinal cord -0.12490 0.0456 30  -0.2637   0.0139
##  liver - muscle        0.14730 0.0456 30   0.0085   0.2861
##  liver - spinal cord  -0.33170 0.0456 30  -0.4705  -0.1929
##  muscle - spinal cord -0.47899 0.0456 30  -0.6178  -0.3402
##
## age.fac = Five:
##  contrast              estimate     SE df lower.CL upper.CL
##  brain - heart         0.54719 0.0456 30   0.4084   0.6860
##  brain - kidney        0.09619 0.0456 30  -0.0426   0.2350
##  brain - liver         0.83686 0.0456 30   0.6981   0.9757
##  brain - muscle        0.57301 0.0456 30   0.4342   0.7118
##  brain - spinal cord   0.35992 0.0456 30   0.2211   0.4987
##  heart - kidney       -0.45101 0.0456 30  -0.5898  -0.3122
##  heart - liver         0.28967 0.0456 30   0.1509   0.4285
##  heart - muscle        0.02582 0.0456 30  -0.1130   0.1646
##  heart - spinal cord  -0.18728 0.0456 30  -0.3261  -0.0485
##  kidney - liver        0.74068 0.0456 30   0.6019   0.8795
##  kidney - muscle       0.47682 0.0456 30   0.3380   0.6156
##  kidney - spinal cord  0.26373 0.0456 30   0.1249   0.4025
##  liver - muscle       -0.26386 0.0456 30  -0.4026  -0.1251
```

```
##  liver - spinal cord  -0.47695 0.0456 30  -0.6157  -0.3382
##  muscle - spinal cord -0.21309 0.0456 30  -0.3519  -0.0743
##
## age.fac = Ten:
##  contrast             estimate     SE df lower.CL upper.CL
##  brain - heart         0.64032 0.0456 30   0.5015   0.7791
##  brain - kidney        0.47072 0.0456 30   0.3319   0.6095
##  brain - liver         0.72444 0.0456 30   0.5856   0.8632
##  brain - muscle        0.64389 0.0456 30   0.5051   0.7827
##  brain - spinal cord   0.22952 0.0456 30   0.0907   0.3683
##  heart - kidney       -0.16959 0.0456 30  -0.3084  -0.0308
##  heart - liver         0.08412 0.0456 30  -0.0547   0.2229
##  heart - muscle        0.00358 0.0456 30  -0.1352   0.1424
##  heart - spinal cord  -0.41080 0.0456 30  -0.5496  -0.2720
##  kidney - liver        0.25371 0.0456 30   0.1149   0.3925
##  kidney - muscle       0.17317 0.0456 30   0.0344   0.3120
##  kidney - spinal cord -0.24121 0.0456 30  -0.3800  -0.1024
##  liver - muscle       -0.08054 0.0456 30  -0.2193   0.0583
##  liver - spinal cord  -0.49492 0.0456 30  -0.6337  -0.3561
##  muscle - spinal cord -0.41438 0.0456 30  -0.5532  -0.2756
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
## Conf-level adjustment: tukey method for comparing a family of 6 estimates
```

This output helps us understand the nature of the interaction. For example, let's consider our observation from the boxplot, above, where we noticed that the difference between expression in liver vs. muscle might be different for ages 'Fifteen' vs. 'Five'. At age 'Fifteen', the 95% CI's for this difference (`contrast` is `liver - muscle`) are 0.0085 and 0.2861. This implies that expression in liver is **greater** than in muscle at this time point. But, at age 'Five' this contrast (`liver - muscle`) has 95% CI's of -0.4026 and -0.1251. These negative values imply that expression in liver is **less** than in muscle (opposite to what we found at time 'Fifteen'). This exemplifies why the interaction occurred: the effect of `tissue` depends on the level of `age.fac` (we could show that 'vice versa' is also true if we'd analyzed the data in `emmeans` **by** `tissue`, instead). Moreover, when we say that the the "**effect of** `tissue` depends on the level of `age.fac`", we refer to the effect sizes found for tissues: when effect sizes for comparisons among `tissue` levels differ among `age.fac` levels, this, by definition, implies an interaction. Take some time now to look through these results more, yourself, to find other possible causes of the interaction.

We would also report the 95% CI's for the effect sizes, above, and interpret them biologically as much as possible.