FES 524 Winter 2018 Lab 5

Contents

Additional random effects due to variation at multiple scales

In Lab 5, you will learn to fit models with multiple, nested random effects as well as multiple fixed effects. The random effects are based on elements of the study design, which involves blocking on watersheds and then measuring the two factors of interest on distinct physical units. The distinct physical units, which we will discuss later in the lab, vary in size. The dataset this week is an example of a *blocked split-plot* design, although it is possible to work with physical units of varying sizes and not have a split plot design.

Load R packages needed today

```
library(dplyr)
library(ggplot2)
library(nlme)
library(gmodels)
```

Read in the dataset

We will be working with the dataset `lab5.example.biomass.txt`, so make sure you have this file and have changed your working directory appropriately. As usual when we read in a dataset we'll take a look at the structure and make any necessary changes. Our two factors of interest today are the overstory type (`overstory`) and the species of tree the litter came from (`litterspp`). `biomass` is the response variable.

```
dbiomass = read.table("lab5.example.biomass.txt", header = TRUE)

head(dbiomass) # Look at the first 6 lines of the data set
```

```
  watershed overstory litterspp biomass
1         A        RA       ACMA       6
2         G        RA       ACMA       7
3         A        RA       ALRU      10
4         C        RA       ALRU      11
5         C        RA       ACMA      18
6         F        RA       ALRU      18
```

Check the structure of the dataset.

```
'data.frame':   64 obs. of  4 variables:
 $ watershed: Factor w/ 8 levels "A","B","C","D",..: 1 7 1 3 3 6 4 7 1 6 ...
 $ overstory: Factor w/ 2 levels "DF","RA": 2 2 2 2 2 2 2 2 1 2 ...
 $ litterspp: Factor w/ 4 levels "ACMA","ALRU",..: 1 1 2 2 1 2 1 2 1 1 ...
 $ biomass  : int  6 7 10 11 18 18 24 25 29 29 ...
```

A nested study design

There were three different sizes of physical units in this study design. The largest physical units were *watersheds*. The researchers picked two different *stands* in each watershed, one with a primarily Douglas-fir overstory and one with a red alder overstory. The stands are the middle-sized physical units. Within each stand, the researchers put out four bags of litter, one for each litter type of interest, and these litter bags represent the *subplots*. The subplots are the smallest physical units in the study. The measurement of the response was done at the level of the subplots (one measurement of biomass for each litter bag in each stand in each watershed). One factor of interest was measured at the stand level (type of overstory), and the other factor of interest was applied at the subplot level (type of litter).

Based on the description of the physical units, this study has a *nested* structure, with stands nested in watersheds and litter bags nested in stands. We recognize a nesting structure when we see that, e.g., the stand in one watershed is distinct from a stand in another watershed.

Implicit vs explicit names for physical units

The current dataset has *implicit* names for stands and plots. Every unique watershed is represented by a unique letter. However, the stands within watersheds do not have unique names. Instead, stands are represented by the overstory species factor. This can lead to confusion between the factor of interest that we will be testing as a fixed effect and the physical units that cause random variation and should be treated as random effects. To avoid this confusion and any mistakes it might cause, we'll be making a new variable called `stand` where we'll give each stand present in the study a unique name. This is called *explicit* naming. See https://bbolker.github.io/mixedmodels-misc/glmmFAQ.html#nested-or-crossed for more discussion.

Because we will be working with linear mixed models, we don't have to uniquely name the subplots. The subplots are our observation-level units and we know that we will get the residual error term (the observation-level random effect) by default in `lme`. However, in your own work you might consider unique naming of all physical units just to help you keep factors of interest vs physical units straight.

We will use the `interaction` function to make unique names for stands. This function works well for perfectly crossed variables, where every level of one variable occurs with every level of the second variable. Another option is `paste`, which we will see next week.

This work will be done in `mutate` to avoid dollar sign notation.

```
dbiomass = mutate(dbiomass, stand = interaction(watershed, overstory))
head(dbiomass)
```

```
  watershed overstory litterspp biomass stand
1         A        RA       ACMA       6  A.RA
2         G        RA       ACMA       7  G.RA
```

```
3         A         RA       ALRU       10  A.RA
4         C         RA       ALRU       11  C.RA
5         C         RA       ACMA       18  C.RA
6         F         RA       ALRU       18  F.RA
```

Initial data exploration

Summary statistics

We'll be looking at our usual summary statistics. It would be appropriate to also look at summary statistics for each factor separately as well as for the factor combinations, which is not shown here.

Several things to notice this week:

1. Biomass is strictly positive (doesn't include 0).
2. Standard deviations vary wildly among the combined factor groups.
3. The highest value for biomass is more than 80 times higher than the lowest value.

```
( sumdat = dbiomass %>%
    group_by(overstory, litterspp) %>%
    summarise(n = n(),
              mean = mean(biomass),
              sd = sd(biomass),
              min = min(biomass),
              max = max(biomass) ) )
```

```
# A tibble: 8 x 7
# Groups: overstory [?]
  overstory litterspp     n  mean    sd   min   max
  <fct>     <fct>     <int> <dbl> <dbl> <dbl> <dbl>
1 DF        ACMA          8  80.4  71.1 29.0  249
2 DF        ALRU          8 127   107   50.0  298
3 DF        PSME          8 189   152   53.0  500
4 DF        TSHE          8 163    89.3 51.0  316
5 RA        ACMA          8  29.5  23.5  6.00  80.0
6 RA        ALRU          8  28.2  14.5 10.0   45.0
7 RA        PSME          8 154    94.1 56.0  295
8 RA        TSHE          8 103    65.2 30.0  204
```

```
# Examine the number of observations in the groups
    # We're looking for balanced vs unbalanced data
xtabs(~overstory + litterspp, dbiomass )
```

```
          litterspp
overstory ACMA ALRU PSME TSHE
       DF    8    8    8    8
       RA    8    8    8    8
```

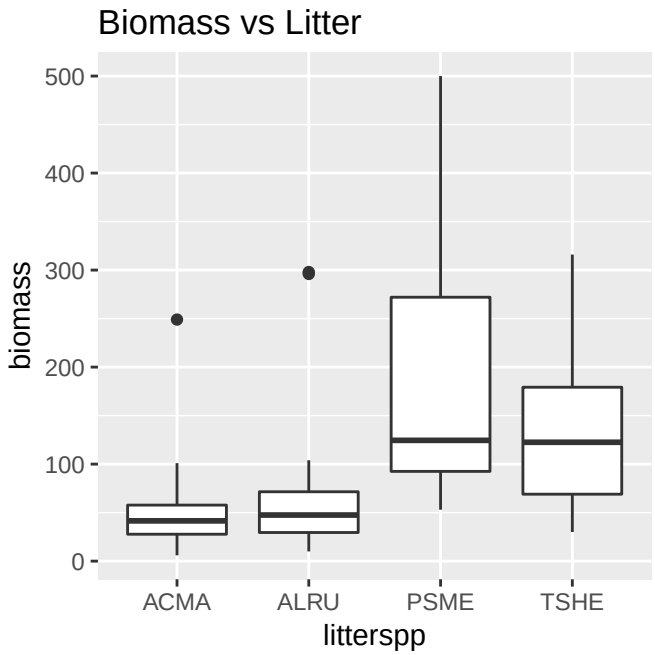
Graphical data exploration

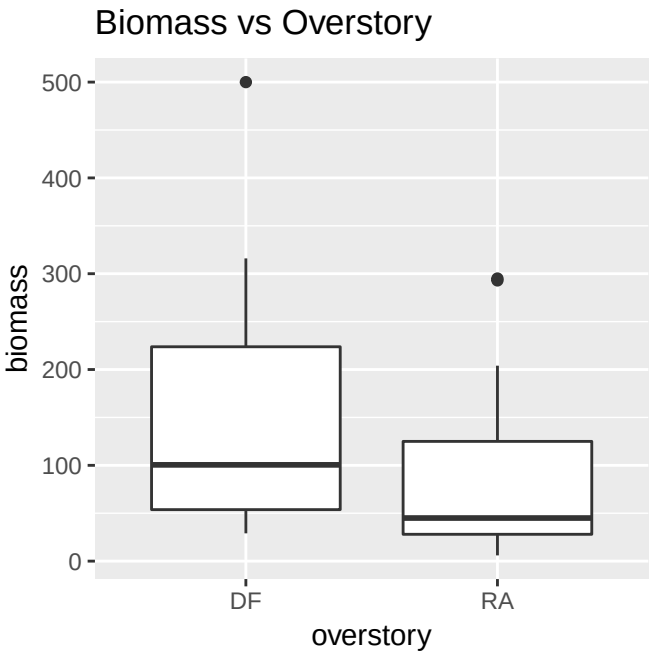
Boxplots

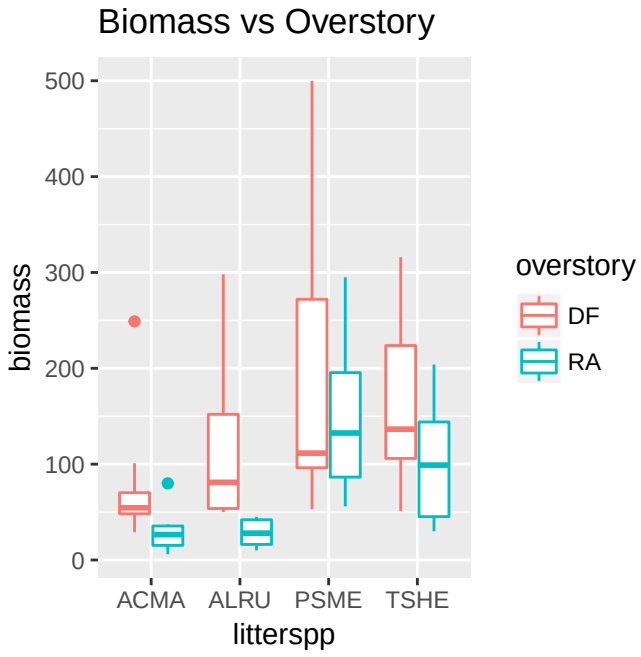
Because we are working with only categorical variables and have a bit more data this week, we can use boxplots in our initial data exploration. Boxplots can help us understand the shape of the observed distribution as long as it is unimodal.

```
# Graphical exploration

# Plot the raw data as boxplots
# First biomass vs each explanatory
```

```r
qplot(litterspp, biomass, data = dbiomass, geom = "boxplot",
    main = "Biomass vs Litter")

qplot(overstory, biomass, data = dbiomass, geom = "boxplot",
    main = "Biomass vs Overstory")

# Factor combination: color by overstory, litterspp on x axis
qplot(litterspp, biomass, color = overstory, data = dbiomass, geom = "boxplot",
    main = "Biomass vs Overstory")
```





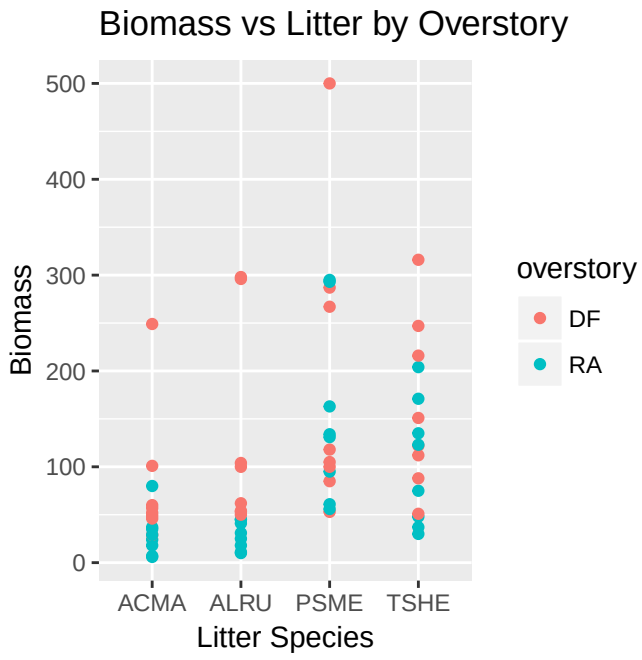


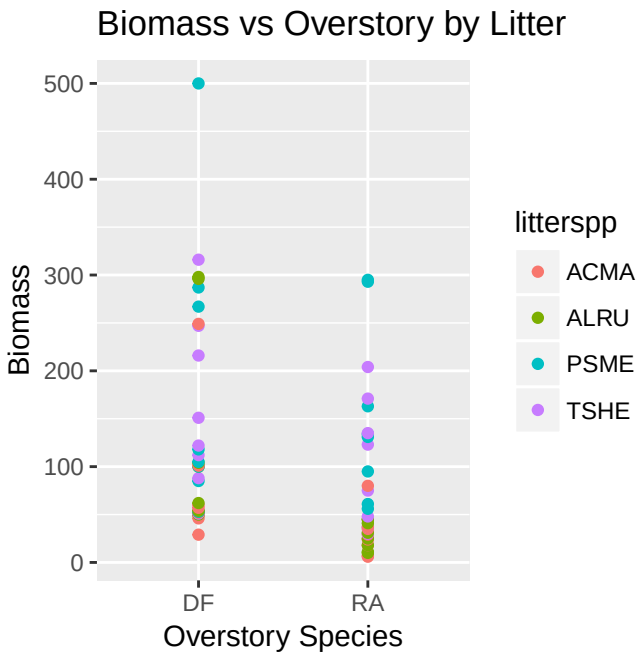
Scatterplots and jittering

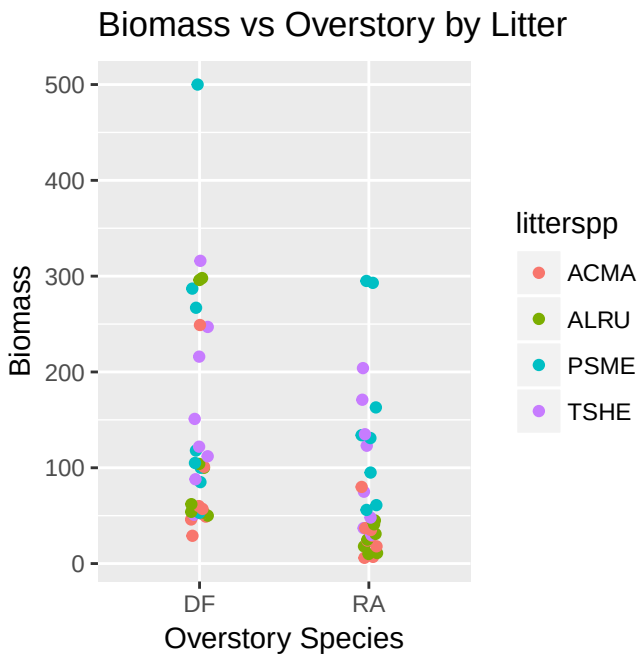
Here are the scatterplots we've been making each week. Like in Lab 4, we will use color to add dimensions to the graphics. As our datasets get larger, it can be more difficult to see individual points in a scatterplot. To solve that problem, we can

jitter the points apart. Below you will see the use of `geom_jitter`. Setting the `width` tells how much to jitter the points. I tend to jitter only a small amount in a scatterplot of groups like this. Notice we have to switch to using `ggplot` directly when we making graphics like this as they are too complicated for `qplot`.

```
# scatter plot of biomass vs litter species
qplot(litterspp, biomass, color = overstory, data = dbiomass,
      xlab = "Litter Species",
      ylab = "Biomass",
      main = "Biomass vs Litter by Overstory")


# scatter plot of biomass vs overstory
qplot(overstory, biomass, color = litterspp, data = dbiomass,
      xlab = "Overstory Species",
      ylab = "Biomass",
      main = "Biomass vs Overstory by Litter")

# I'm having a hard time seeing the colors in the above plot
  # because of dot overlap
# I will add some "jitter" to spread the points around
  # by using the position argument with position_jitter
# The spread of the jitter is controlled using "width"
# To use jitter I have to switch to using ggplot directly
ggplot(dbiomass, aes(overstory, biomass, color = litterspp)) +
    geom_jitter(width = .05, height = 0) +
    xlab("Overstory Species") +
    ylab("Biomass") +
    ggtitle("Biomass vs Overstory by Litter")
```





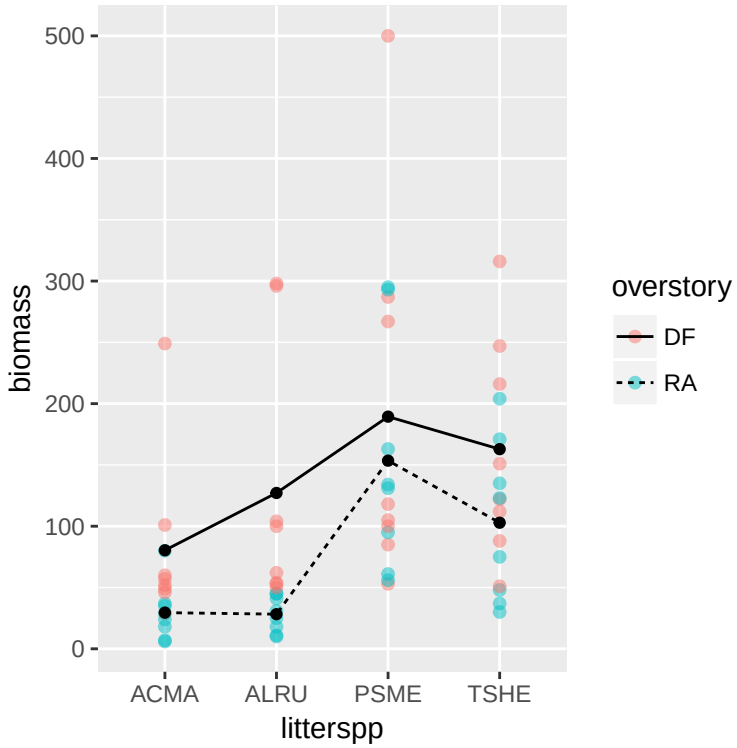


Interaction plot

Because we have factors that are perfectly crossed, we need to think about the interaction. We'll explore the interaction with an interaction plot like we made in Lab 4.

What do you think, do you think there could be a detectable interaction?

```
ggplot(dbiomass, aes(x = litterspp, y = biomass,
                     group = overstory, linetype = overstory)) +
    geom_point(alpha = .5, size = 1.75,
               aes(color = overstory) ) + # Add raw data
    stat_summary(fun.y = mean, geom = "point") + # Add points for means
    stat_summary(fun.y = mean, geom = "line") # Connect points with lines
```



Fitting a linear mixed model with nested random effects in `lme`

We will fit a linear mixed model using `lme` from package **nlme**, where `watershed` and `stand` are random effects and the two factors of interest, `overstory` and `litterspp`, are fixed effects. We will include a term for the interaction between `overstory` and `litterspp`.

This week I use the short-cut coding for the fixed effects. Using the symbol `*` with two variables indicates I am putting each variable plus the interaction between the variables into the model. So `litterspp*overstory` coding expands to `litterspp + overstory + litterspp:overstory`.

Notice the use of the forward slash, `/`, in the random effects. The forward slash represents nesting in `lme`. Below we are stating that `stand` is nested in `watershed` by our coding in the `random` argument. The random effect of subplot is the observation-level random effect, so this is the residual error from the model.

```
model1 = lme(biomass ~ litterspp*overstory, random = ~1|watershed/stand,
             data = dbiomass)
```

Checking that the model structure is coded correctly

Before we check our assumptions, I wanted to take a moment and review how R and other software packages do what we tell them to even if what we are doing is wrong. If we take a look at `model1`, we can check the structure of the random effects by examining the `Number of Groups` section. This tells us we have 8 watersheds in our data, which is true. It also tells us we have 16 stands - as there are 2 stands per 8 watersheds, this is also true. The number in the line above, `Number of Observations`, matches the number of rows in our dataset. The structure of the model reflects the structure of our data, which makes us confident that we fit our random effects correctly.

```
model1
```

```
Linear mixed-effects model fit by REML
  Data: dbiomass
  Log-restricted-likelihood: -337.7844
  Fixed: biomass ~ litterspp * overstory
            (Intercept)            littersppALRU            littersppPSME
                 80.375                   46.750                  109.000
```

```
           littersppTSHE               overstoryRA littersppALRU:overstoryRA
                  82.500                   -50.875                   -48.000
littersppPSME:overstoryRA littersppTSHE:overstoryRA
                  15.000                    -9.125

Random effects:
 Formula: ~1 | watershed
        (Intercept)
StdDev:    24.87125

 Formula: ~1 | stand %in% watershed
        (Intercept) Residual
StdDev:  0.02601692 84.03225

Number of Observations: 64
Number of Groups:
           watershed stand %in% watershed
                   8                  16
```

Look at what happens if we were to put our variable in `random` backwards, essentially saying that watersheds are nested in stands. This happens a lot, especially for folks trained in SAS before they started learning R.

```
lme(biomass ~ litterspp*overstory, random = ~1|stand/watershed,
             data = dbiomass)
```

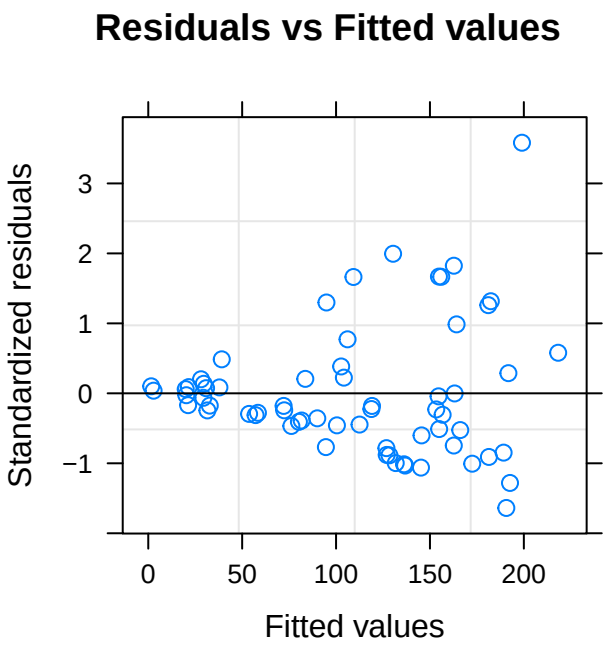
```
Linear mixed-effects model fit by REML
  Data: dbiomass
  Log-restricted-likelihood: -338.058
  Fixed: biomass ~ litterspp * overstory
             (Intercept)             littersppALRU             littersppPSME
                  80.375                    46.750                   109.000
           littersppTSHE               overstoryRA littersppALRU:overstoryRA
                  82.500                   -50.875                   -48.000
littersppPSME:overstoryRA littersppTSHE:overstoryRA
                  15.000                    -9.125

Random effects:
 Formula: ~1 | stand
        (Intercept)
StdDev:    17.01507

 Formula: ~1 | watershed %in% stand
        (Intercept) Residual
StdDev:    17.01547 84.26719

Number of Observations: 64
Number of Groups:
               stand watershed %in% stand
                  16                  16
```

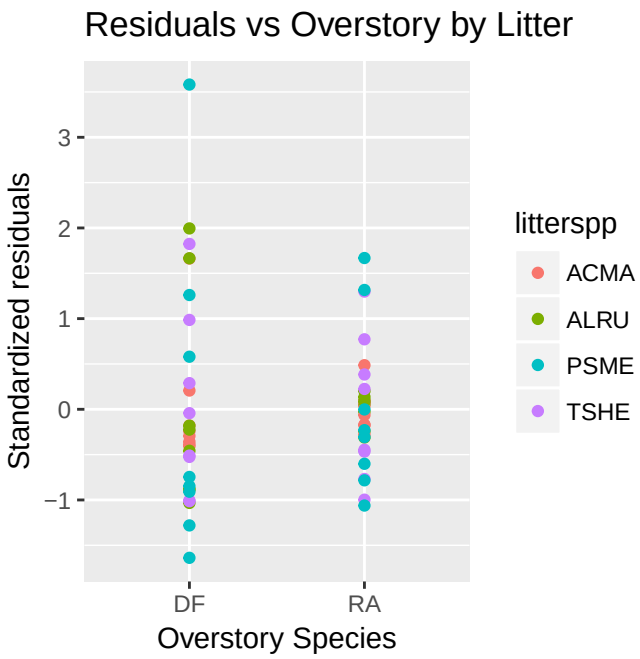
The model fit without complaint, and if we weren't paying attention we might go on and use this model for inference. But if we check the `Number of Groups` we see that the model assumes the wrong number of watersheds (16 instead of 8). This would alert us that we defined the model incorrectly.

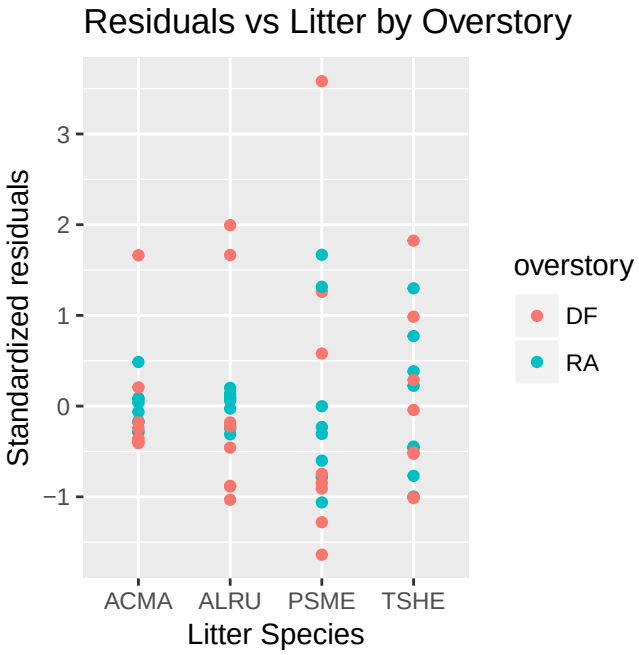
Checking model assumptions

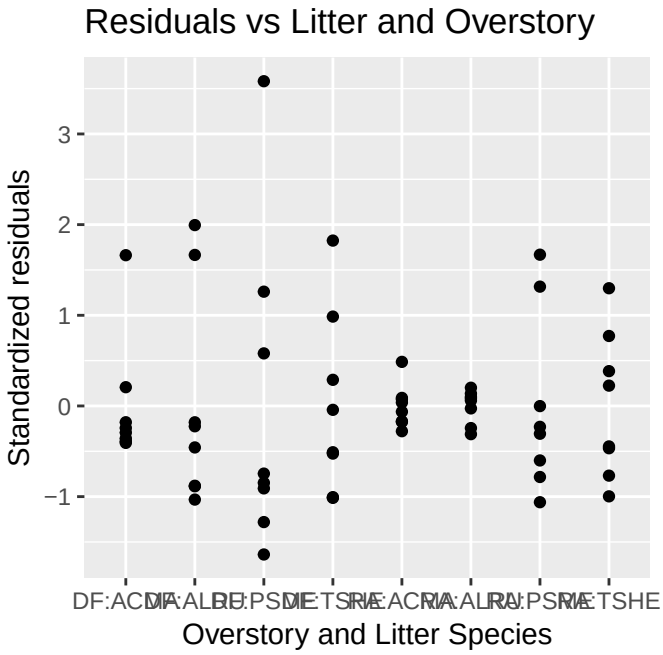
As always, we'll need to check the assumptions of the model using residual plots. We can add the residuals to the dataset `dbiomass`, and then plot the residuals vs the fitted values, the residuals vs the explanatory variables, and check the normality/symmetry of the residuals with a quantile-quantile plot and a boxplot.

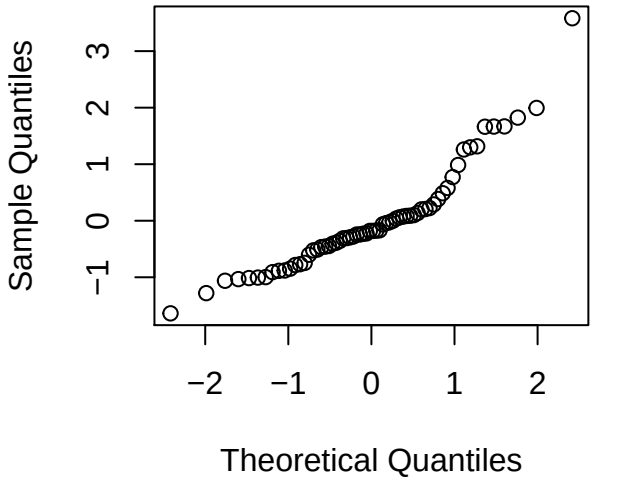
```r
# Save the residual values for assumption checking.
dbiomass$res = resid(model1, type = "pearson")

# Plot residuals vs fitted values
plot(model1, main = "Residuals vs Fitted values")

# Make scatter plots of residuals vs explanatory variables
# overstory
qplot(overstory, res, color = litterspp, data = dbiomass,
      xlab = "Overstory Species",
      ylab = "Standardized residuals",
      main = "Residuals vs Overstory by Litter")

# litter species
qplot(litterspp, res, color = overstory, data = dbiomass,
      xlab = "Litter Species",
      ylab = "Standardized residuals",
      main = "Residuals vs Litter by Overstory")

# combination of overstory and litter species
qplot(x = overstory:litterspp, res, data = dbiomass,
      xlab = "Overstory and Litter Species",
      ylab = "Standardized residuals",
      main = "Residuals vs Litter and Overstory")

# Check normality of residuals with normal probability plot or boxplot
qqnorm(dbiomass$res, main = "Normal Q-Q Plot of Residuals")

qplot(x = "res", y = res, data = dbiomass,
     geom = "boxplot",
     main = "Boxplot of standardized residuals")
```

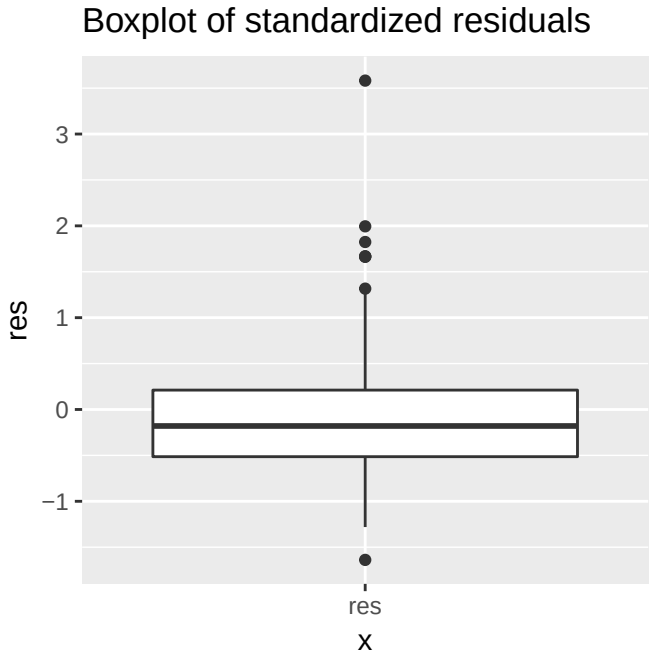

Residuals vs Fitted values
Standardized residuals
Fitted values
3
2
1
0
−1
0
50
100
150
200


Residuals vs Overstory by Litter
Standardized residuals
Overstory Species
DF
RA
3
2
1
0
−1
litterspp
ACMA
ALRU
PSME
TSHE


Residuals vs Litter by Overstory
Standardized residuals
Litter Species
ACMA
ALRU
PSME
TSHE
3
2
1
0
−1
overstory
DF
RA


Residuals vs Litter and Overstory
Standardized residuals
Overstory and Litter Species
3
2
1
0
−1





Refitting the model when the assumptions are not met

The residual plots from `model1` indicate a problem. In the first plot we saw that the variance of the residuals increased with the fitted values, and further plots showed us that the groups don't appear to have homogeneous variances in either factor variable.

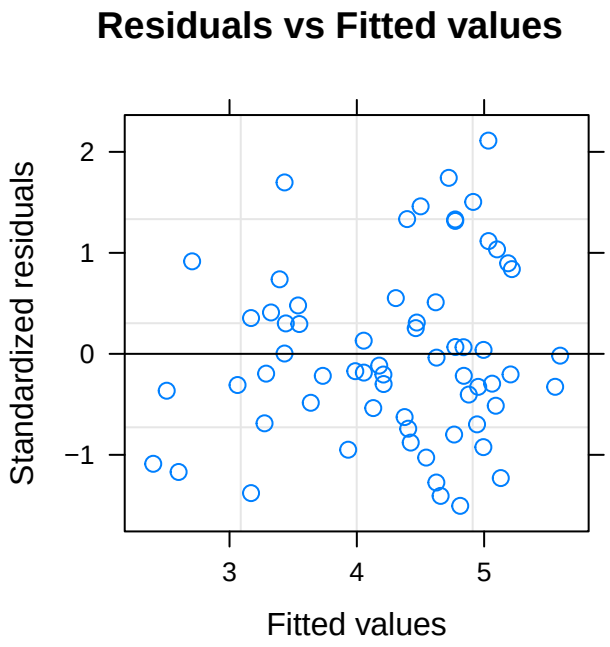
We could try to address this by allowing variances to be different among the levels of both of the factors, much like we saw last week. However, this would be a very complicated model. In addition, allowing for variances to differ among groups does not address the long right tail we see in the boxplot and the quantile-quantile plot.

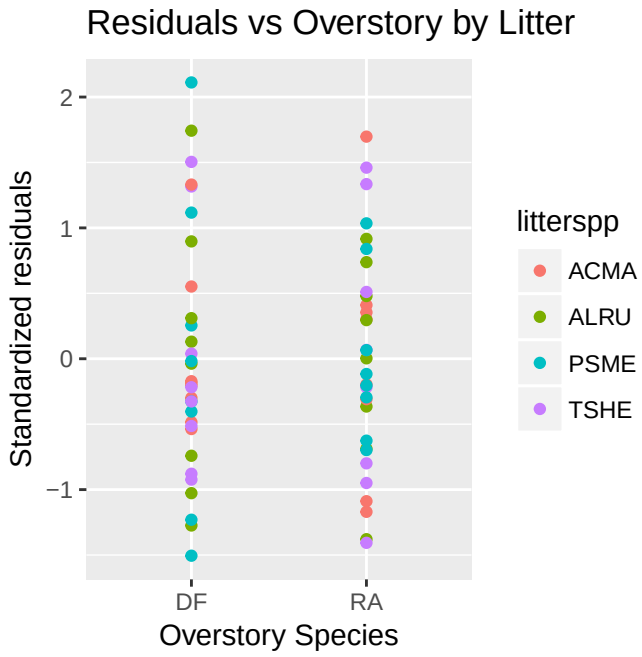
Let's think about our observed data and the residuals a little more. The values of our response, biomass, are strictly positive (so don't include zero). The residuals are right-skewed and the residual variance increases with the mean. A dataset like this is a good candidate for modeling the response with either a log-normal or a gamma distribution. Using the gamma distribution would mean we would have to switch to using a generalized linear mixed model, which we are not covering in this class. To fit a log-normal model, though, we can simply do a natural logarithm transformation on our response and stick with a linear mixed model.

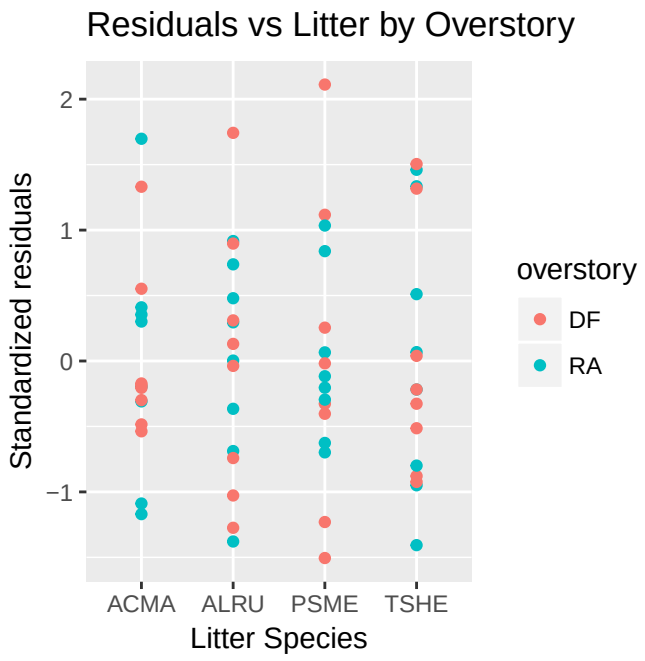
Let's transform `biomass` and use `log(biomass)` as the response in a new model. We would need to go back and remake our exploratory plots with the transformed response variable, but we are not going to take the time today. Instead, we'll fit a second model, `model2`, using the newly transformed response. We'll still need to check our assumptions for this new model before we can use it to make inference.

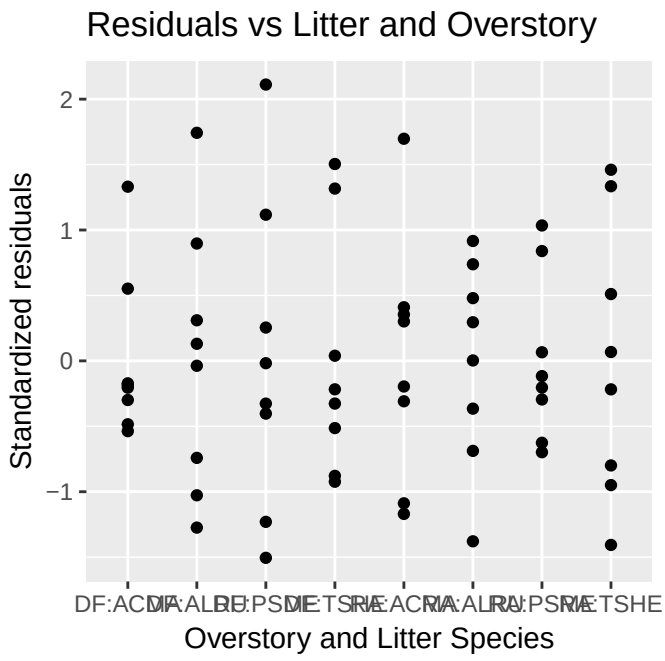
```
# Make a new response variable
dbiomass$lbio = log(dbiomass$biomass)

# Fit a model with the transformed response
model2 = lme(lbio ~ litterspp*overstory, random = ~1|watershed/stand,
             data = dbiomass)

# Compute and save residual values
dbiomass$res2 = resid(model2, type = "pearson")

# Plot residuals vs fitted values
plot(model2, main = "Residuals vs Fitted values")

# Make scatter plots of residuals vs explanatory variables
# overstory
```

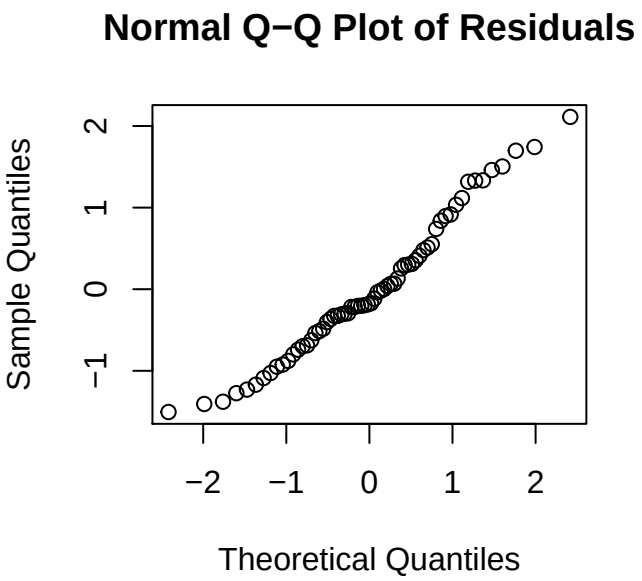
```r
qplot(overstory, res2, color = litterspp, data = dbiomass,
      xlab = "Overstory Species",
      ylab = "Standardized residuals",
      main = "Residuals vs Overstory by Litter")

# litter species
qplot(litterspp, res2, color = overstory, data = dbiomass,
      xlab = "Litter Species",
      ylab = "Standardized residuals",
      main = "Residuals vs Litter by Overstory")

# interaction of overstory and litter species
qplot(x = overstory:litterspp, res2, data = dbiomass,
      xlab = "Overstory and Litter Species",
      ylab = "Standardized residuals",
      main = "Residuals vs Litter and Overstory")

# Check normality of residuals with normal probability plot and boxplot
qqnorm(dbiomass$res2, main = "Normal Q-Q Plot of Residuals")

qplot(x = "res", y = res2, data = dbiomass,
     geom = "boxplot",
     main = "Boxplot of standardized residuals")
```

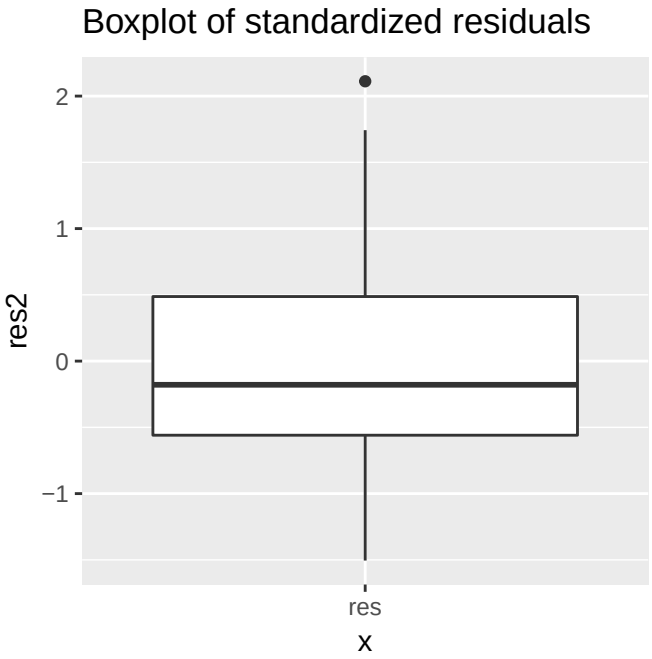












Model results

If the assumptions are now reasonably met, we can report any model results of interest from `anova` and/or `summary`. Make note of that there is statistical evidence that the effect of the litter species is different among the overstory canopy types.

```
anova(model2)
```

```
                   numDF denDF  F-value p-value
(Intercept)            1    42 905.6619  <.0001
litterspp              3    42  19.8215  <.0001
overstory              1     7  12.5848  0.0094
litterspp:overstory    3    42   4.0325  0.0131
```

```
summary(model2)
```

```
Linear mixed-effects model fit by REML
```

```
 Data: dbiomass
       AIC      BIC    logLik
  148.7095 170.9884 -63.35475

Random effects:
 Formula: ~1 | watershed
        (Intercept)
StdDev:   0.2572113

 Formula: ~1 | stand %in% watershed
        (Intercept) Residual
StdDev:   0.3366683 0.559389

Fixed effects: lbio ~ litterspp * overstory
                              Value Std.Error DF   t-value p-value
(Intercept)                4.163621 0.2480976 42 16.782193  0.0000
littersppALRU              0.415810 0.2796945 42  1.486656  0.1446
littersppPSME              0.824391 0.2796945 42  2.947469  0.0052
littersppTSHE              0.785050 0.2796945 42  2.806813  0.0076
overstoryRA               -1.073411 0.3264435  7 -3.288197  0.0133
littersppALRU:overstoryRA -0.309694 0.3955477 42 -0.782949  0.4380
littersppPSME:overstoryRA  0.950903 0.3955477 42  2.404017  0.0207
littersppTSHE:overstoryRA  0.546995 0.3955477 42  1.382881  0.1740
 Correlation:
                           (Intr) ltALRU ltPSME ltTSHE ovrsRA lALRU: lPSME:
littersppALRU              -0.564
littersppPSME              -0.564  0.500
littersppTSHE              -0.564  0.500  0.500
overstoryRA                -0.658  0.428  0.428  0.428
littersppALRU:overstoryRA   0.399 -0.707 -0.354 -0.354 -0.606
littersppPSME:overstoryRA   0.399 -0.354 -0.707 -0.354 -0.606  0.500
littersppTSHE:overstoryRA   0.399 -0.354 -0.354 -0.707 -0.606  0.500  0.500

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-1.5051788 -0.5593704 -0.1786940  0.4870866  2.1116289

Number of Observations: 64
Number of Groups:
           watershed stand %in% watershed
                   8                  16
```

Estimating group differences

This study was designed to answer four specific questions, listed below.

1. Is the total microbial biomass under a Douglas-fir overstory the same as the total microbial biomass under red alder?
2. Is the total microbial biomass in the decomposing litter of the two conifer species under Douglas-fir the same as the total microbial biomass in the decomposing litter of the two conifer species under the red alder over story?
3. Is the total microbial biomass in the decomposing red alder litter under red alder overstory the same as the total microbial biomass in the decomposing Douglas-fir litter under Douglas-fir?
4. Did the decomposing Douglas-fir litter under Douglas-fir overstory have more than twice the total microbial biomass compared to any of the other factor level combinations?

Because of the presence of an interaction between overstory canopy species and litter species, we'll need to figure out appropriate comparisons to answer each of these questions. We will be doing a total of 11 unique comparisons that answer the four questions. You'll notice that one comparison is part of both question 1 and question 4 and another comparison is part of both question 3 and question 4. I will do the Bonferroni correction for the family of 11 comparisons, although this may be so conservative that I won't be able to detect differences that really exist. This means I increase the possibility of a Type II error while controlling for the familywise Type I error rate.

Part of the decision about correcting for multiple comparisons and what correction to make has to do with the research question and whether we are more concerned with making a Type I or Type II error.

```
1 - .05/11
```

```
[1] 0.9954545
```

If you are going to report p-values for the comparisons, you need to make sure you interpret them compared to a Bonferroni-adjusted alpha. If we are using an alpha of .05, then the familywise alpha is .05 divided by the number of comparisons. The Bonferroni-adjusted alpha for this example, with 11 comparisons, is shown below.

```
.05/11
```

```
[1] 0.004545455
```

As usual, we'll start by writing out the linear combinations of coefficients for each factor combination group mean.

```
df_acma = c(1, 0, 0, 0, 0, 0, 0, 0)
df_alru = c(1, 1, 0, 0, 0, 0, 0, 0)
df_psme = c(1, 0, 1, 0, 0, 0, 0, 0)
df_tshe = c(1, 0, 0, 1, 0, 0, 0, 0)
ra_acma = c(1, 0, 0, 0, 1, 0, 0, 0)
ra_alru = c(1, 1, 0, 0, 1, 1, 0, 0)
ra_psme = c(1, 0, 1, 0, 1, 0, 1, 0)
ra_tshe = c(1, 0, 0, 1, 1, 0, 0, 1)
```

We will use `estimable` from package **gmodels** to make estimates of differences in mean log biomass that answer our questions of interest, starting with our first research question. Because of the statistically detectable interaction, we will compare overstory effects (DF vs RA) within each litter species.

```
acma_dfra = df_acma - ra_acma
alru_dfra = df_alru - ra_alru
psme_dfra = df_psme - ra_psme
tshe_dfra = df_tshe - ra_tshe

# Use "estimable" from package gmodels as in the past

( overdiff = estimable(model2, rbind(acma_dfra, alru_dfra,
                               psme_dfra, tshe_dfra),
    conf.int = .996) )
```

```
Warning in estimable.default(model2, rbind(acma_dfra, alru_dfra, psme_dfra, : Degrees of freedom
vary among parameters used to construct linear contrast(s): 2, 3, 4. Using the smallest df among the
set of parameters.
```

```
           Estimate Std. Error   t value DF    Pr(>|t|)     Lower.CI Upper.CI
acma_dfra 1.0734109  0.3264435 3.2881975  7 0.013335594 -0.299977816 2.446800
alru_dfra 1.3831044  0.3264435 4.2368869  7 0.003855125  0.009715726 2.756493
psme_dfra 0.1225074  0.3264435 0.3752791  7 0.718565598 -1.250881225 1.495896
tshe_dfra 0.5264155  0.3264435 1.6125775  7 0.150871268 -0.846973164 1.899804
```

Warnings from `estimable`

Note the warning message from `estimable`. When working with `lme` objects where the denominator degrees of freedom vary depending on the factor, `estimable` gives conservative confidence intervals even for balanced, classic study designs like this. If you have a model like this in your own work, you can use the results from `estimable` to calculate tests and confidence

intervals using the known degrees of freedom. I will not show you how to do that today in lab, but ask me if this occurs in your own work and I can help you figure it out.

Back-transforming estimates and CI limits

Our estimates of the differences in means and confidence intervals are on the natural logarithm scale. We will need to exponentiate our results so we can make inference and interpret the multiplicative differences in median biomass on the original, untransformed scale. When we write statements using multiplicative results, we will use language that one group's median was estimated to be "X times" or "X percent larger/smaller" than another group's median.

```
( eoverdiff = exp(overdiff[ , c("Estimate", "Lower.CI", "Upper.CI")]) )
```

```
          Estimate  Lower.CI  Upper.CI
acma_dfra 2.925340 0.7408347 11.551318
alru_dfra 3.987260 1.0097631 15.744531
psme_dfra 1.130328 0.2862524  4.463334
tshe_dfra 1.692853 0.4287106  6.684585
```

We will go on to answer the other three questions in much the same manner.

Taking the average of linear combinations of coefficients

The second research question is one about comparing total microbial biomass among overstory canopy types overall for the two conifer litters. This is a little different than other comparisons we've done, because we'll need to calculate a "conifer" vector for each overstory by averaging the vectors that represent that two conifer litter species together within each overstory. Once we have done the averaging we can do the comparisons via subtraction as we usually do.

```
# Comparisons for Question 2
# Compute the mean of the two litter types of interest under DF overstory
dfconif = (df_psme + df_tshe)/2
raconif = (ra_psme + ra_tshe)/2

# Define comparison vector
diffconif = dfconif - raconif

# We want an estimate of the difference in the means
( conifdiff = estimable(model2, rbind(diffconif),
    conf.int = .996) )
```

```
           Estimate Std. Error  t value DF  Pr(>|t|)   Lower.CI Upper.CI
diffconif 0.3244615  0.2597131 1.249307  7 0.2517062 -0.7681838 1.417107
```

```
# Back transform
( econifdiff = exp(conifdiff[,c("Estimate", "Lower.CI", "Upper.CI")]) )
```

```
          Estimate  Lower.CI Upper.CI
diffconif 1.383286 0.4638547 4.125168
```

```
# Comparisons for Question 3
# Define comparison vector
diffdfdf_rara = df_psme - ra_alru

# Make estimate of differences
( dfdfraradiff = estimable(model2, rbind(diffdfdf_rara),
    conf.int = .996) )
```

```
              Estimate Std. Error  t value DF     Pr(>|t|) Lower.CI Upper.CI
diffdfdf_rara 1.791686  0.3264435 5.488501  7 0.0009176548 0.418297 3.165074
```

```
# Back transform
( edfdfraradiff = exp(dfdfraradiff [ ,c("Estimate", "Lower.CI", "Upper.CI")]) )
```

```
              Estimate Lower.CI Upper.CI
diffdfdf_rara 5.999557 1.519372  23.6905
```

```
# Comparisons for Question 4

# Define comparison vectors
# Compare all overstory/litter combos to Doug-fir overstory with Doug-fir litter.
df_acma_psme = df_psme - df_acma
df_alru_psme = df_psme - df_alru
df_tshe_psme = df_psme - df_tshe
ra_acma_psme = df_psme - ra_acma
ra_alru_psme = df_psme - ra_alru
ra_psme_psme = df_psme - ra_psme
ra_tshe_psme = df_psme - ra_tshe


# Make estimates
( psmediff = estimable(model2, rbind(df_acma_psme, df_alru_psme, df_tshe_psme,
                             ra_acma_psme, ra_alru_psme, ra_psme_psme,
                             ra_tshe_psme),
                 conf.int = .996) )
```

```
               Estimate Std. Error   t value DF     Pr(>|t|)    Lower.CI  Upper.CI
df_acma_psme 0.82439085  0.2796945 2.9474691 42 0.0052116241 -0.02746701 1.6762487
df_alru_psme 0.40858124  0.2796945 1.4608126 42 0.1515080362 -0.44327662 1.2604391
df_tshe_psme 0.03934077  0.2796945 0.1406562 42 0.8888139725 -0.81251710 0.8911986
ra_acma_psme 1.89780171  0.3264435 5.8135678  7 0.0006544160  0.52441304 3.2711904
ra_alru_psme 1.79168564  0.3264435 5.4885007  7 0.0009176548  0.41829697 3.1650743
ra_psme_psme 0.12250745  0.3264435 0.3752791  7 0.7185655981 -1.25088122 1.4958961
ra_tshe_psme 0.56575628  0.3264435 1.7330907  7 0.1266782595 -0.80763240 1.9391449
```

```
# Back transform
( epsmediff = exp(psmediff[ ,c("Estimate", "Lower.CI", "Upper.CI")]) )
```

```
              Estimate  Lower.CI  Upper.CI
df_acma_psme 2.280491 0.9729068  5.345466
df_alru_psme 1.504681 0.6419296  3.526970
df_tshe_psme 1.040125 0.4437397  2.438050
ra_acma_psme 6.671213 1.6894669 26.342678
ra_alru_psme 5.999557 1.5193718 23.690504
ra_psme_psme 1.130328 0.2862524  4.463334
ra_tshe_psme 1.760779 0.4459126  6.952803
```

Wrapping up the analysis

Graphic

Once we have our results, we can make tables and figures to include in our write-up. Here I will work on a table of results and a graphic for the comparisons that answer question 1 and then make a table of summary statistics. I will leave any further graphics, which I likely need for my example write-up, to the "Bonus graphics" portion of the lab.

```
# Make nicer names for comparisons and add to dataset
eoverdiff$estdiffnames = factor( c("DF over RA overstory, ACMA litter",
                           "DF over RA overstory, ALRU litter",
                           "DF over RA overstory, PSME litter",
                           "DF over RA overstory, TSHE litter") )

( g1 = ggplot(eoverdiff, aes(x = Estimate, y = estdiffnames) ) + # Put response on x axis
    geom_errorbarh(height = .2, lwd=.75, aes(xmin = Lower.CI, xmax = Upper.CI) ) + # Add error bar
    geom_point(size=2.5) + # Add points
```

```
    labs(x = "Ratio of Total Microbial Biomass",
        y = NULL) + # Label axes
    geom_vline(xintercept = 1, lty = 2) + # add a horiz line when ratio at 1
    geom_rect(alpha = .0625, ymin = 0, ymax = 5, xmin = .5, xmax = 2) +  # add rect
    scale_x_continuous(breaks = seq(0, 18, by = 3) ) + # Add more breaks on x
    theme_bw() + # Make graph black and white for printing
     theme(axis.ticks.y = element_blank(),
         axis.text.y = element_blank(), # Remove axis labels and tick marks
         panel.grid.major.y = element_blank() ) + # Remove y gridlines
    geom_text(aes(x = rep(1.5, 4), y = as.numeric(estdiffnames)  + .2,
                label = estdiffnames), size = 4, hjust = 0) + # Place text label
    annotate("text", x = 1.25, y = .5, label = "No difference", size = 3) ) # Label no difference
```

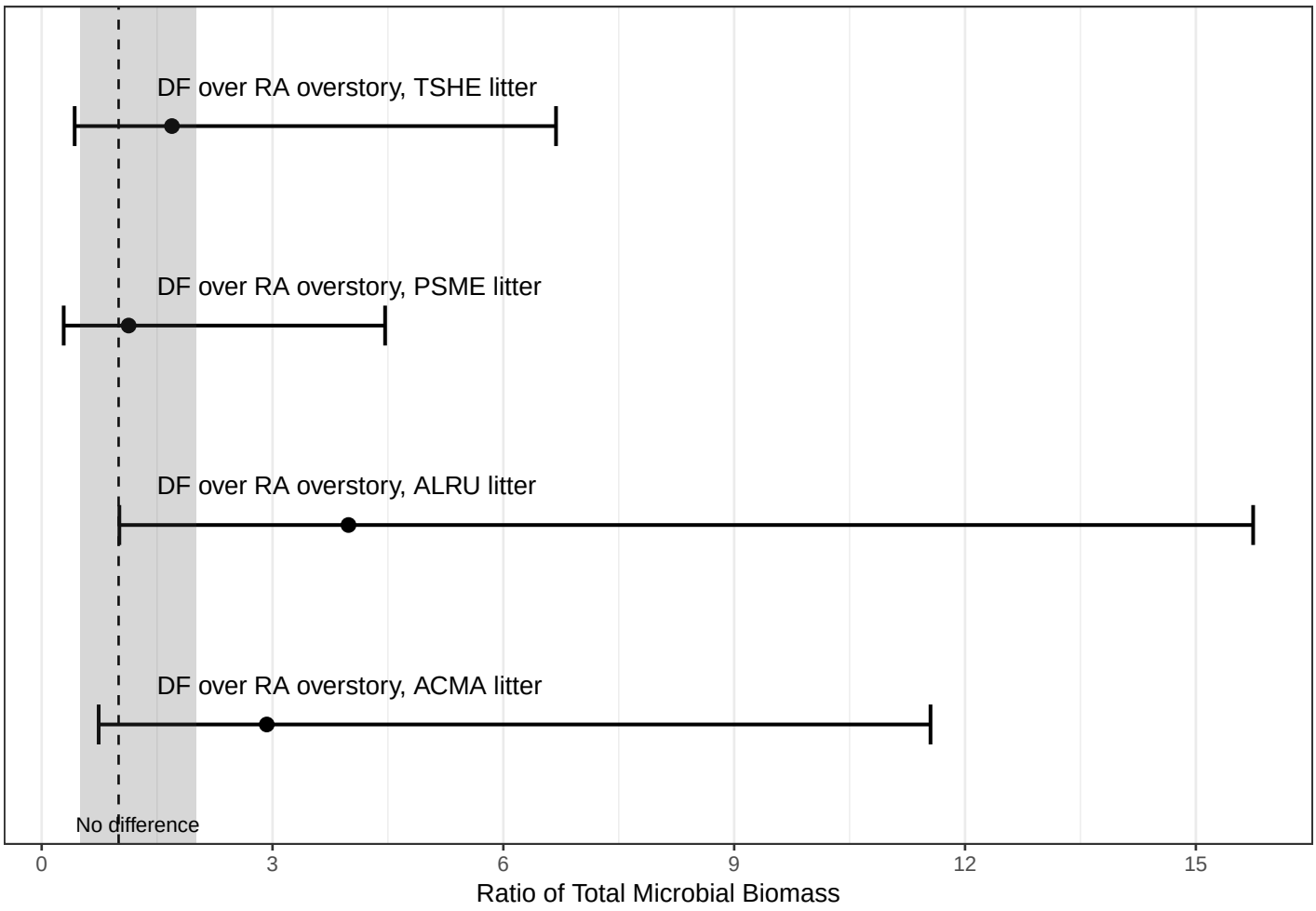


Table of results

```
# Make a table for estimates of
    # ratios of medians and confidence intervals
    # to answer question 1
# Start by rounding everything to 1 digit
eoverdifftab = eoverdiff %>%
    select_if(is.numeric) %>%
    mutate_all(round, 1)

# I am going to use paste to make a single column for the confidence intervals
    # Using sprintf() to force 1 decimal place in Lower.CI
```

```
eoverdifftab = mutate(eoverdifftab, ci = paste(sprintf("%.1f", Lower.CI),
                                    Upper.CI, sep = ", ") )

# I need to make the row names and column names looks nicer
rownames(eoverdifftab) = c("DF over RA overstory, ACMA litter",
                       "DF over RA overstory, ALRU litter",
                       "DF over RA overstory, PSME litter",
                       "DF over RA overstory, TSHE litter")
colnames(eoverdifftab) = c("Ratio of medians", "Lower", "Upper", "99.6% CI")

# This is a table of results I could put in my write up
eoverdifftab[,c("Ratio of medians", "99.6% CI")]
```

	Ratio of medians	99.6% CI
DF over RA overstory, ACMA litter	2.9	0.7, 11.6
DF over RA overstory, ALRU litter	4.0	1.0, 15.7
DF over RA overstory, PSME litter	1.1	0.3, 4.5
DF over RA overstory, TSHE litter	1.7	0.4, 6.7

Summary table

Below I create a summary table of descriptive statistics. Notice that I summarize the data using the median as the measure of center and the interquartile range as a measure of spread. The median and interquartile range are generally a better way to describe skewed data like this compared to the mean and standard deviation. Also, our results are about ratios of medians, so showing the means of the observed data didn't make a lot of sense.

```
# A table of summary statistics by combined factor group, if desired
( sumtable = dbiomass %>%
        group_by("Overstory" = overstory,
                 "Litter" = litterspp) %>%
        summarise(n = n(),
                  Median = round( median(biomass) ),
                  "1st quartile" = round( quantile(biomass, .25) ),
                  "3rd quartile" = round( quantile(biomass, .75) ) ) )
```

Overstory	Litter	n	Median	1st quartile	3rd quartile
DF	ACMA	8	54	48	70
DF	ALRU	8	81	54	152
DF	PSME	8	112	96	272
DF	TSHE	8	136	106	224
RA	ACMA	8	26	15	36
RA	ALRU	8	28	16	42
RA	PSME	8	132	86	196
RA	TSHE	8	99	45	144