Lecture 19. Mixed models I.

Reasons why mixed models are important

Mixed models (aka mixed-effects models, hierarchical models, multilevel models) are an extremely important technique in biology. In some ways mixed models are the culmination of this course, although we will cover a few additional topics. Mixed models are so useful (and often necessary) because they allow us to deal with two related issues:

1. From a statistical perspective, biological data are often structured in a way that violates the assumption of independently distributed data. Examples include:

- multiple measurements on the same organism
- experiments organized into spatial blocks
- observational data where multiple observations were made at each of many locations
- observational data where multiple observations were made at each of many timepoints
- community data where multiple observations (e.g. abundance) were made on each of many species
- data syntheses of similar experiments that were performed by many different researchers

If we analyze such data as if each observation was independent, we are likely to have the *pseudoreplication* problem that we discussed earlier.

2. From a biological perspective, the processes we measure can be affected by multiple sources of variation, which often occur at different spatial or temporal scales. We would like to use statistical methods that can model multiple sources of stochasticity, over multiple scales, so that we can ask: what is the relative magnitude of different sources of variation? and what predictors explain variation at different scales?

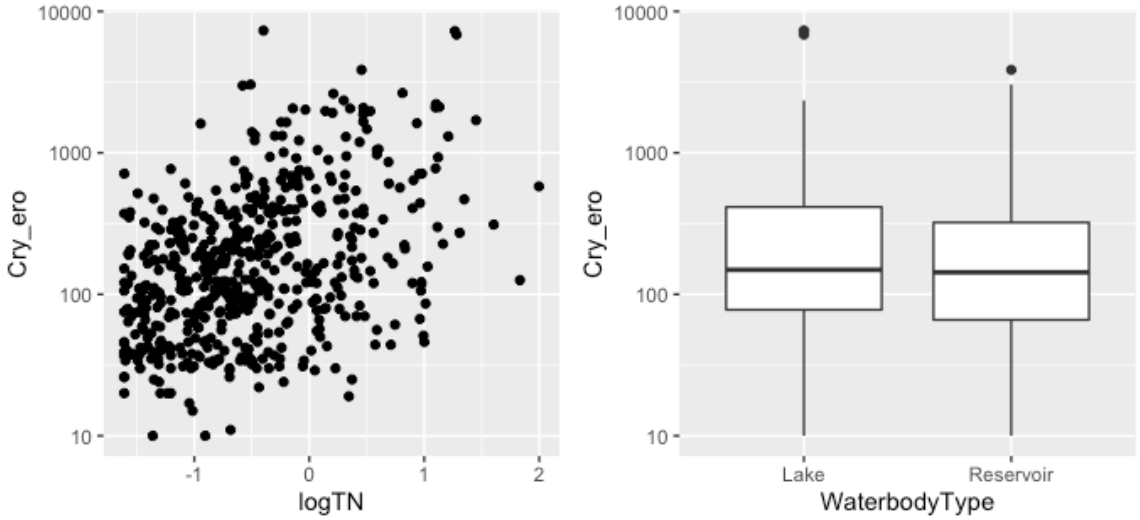
Points 1 and 2 are essentially two sides of the same coin, and because these issues are so often applicable in both experimental and observational studies, mixed models are very often the appropriate method for analysis.

An example to introduce random effects

Let's start with a relatively simple example to learn how mixed models work, before moving on to more interesting scenarios. In the mid 1970s the EPA did a large survey of US lakes, measuring the phytoplankton community and limnological variables in more than 500 lakes, with 3-4 samples per lake (roughly in spring, summer, and fall of one year). Let's say I want to quantify the relationship

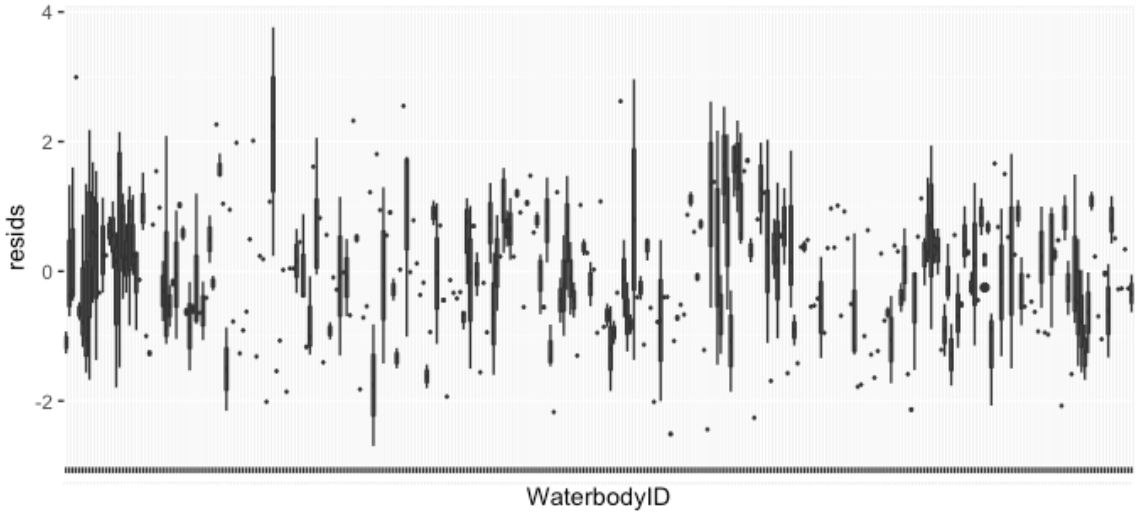
between the abundance of a common cryptomonad phytoplankter, *Cryptomonas erosa*, and some environmental predictors.

With the raw data, these are the relationships between log(algal density) and total nitrogen (TN; mg L^{-1}), and waterbody type (lake or reservoir):



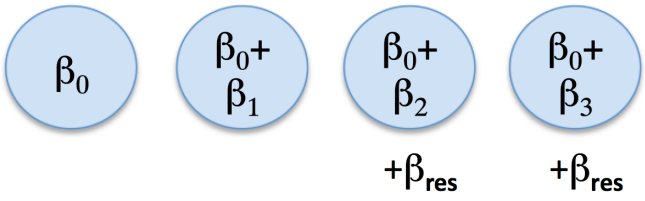
Here I'm only looking at the data where *Crytpmonas* (Cry_ero) is >0 (recall the challenges of modeling positive continuous data that includes zeros). How should we model these relationships? There is enough data here (535 observations of 322 waterbodies) that the answer is probably robust to whatever method we use, but that will not always be the case, and even in this case the effect of waterbody type may depend on whether we account for other predictors like TN. For the effect of TN we could just use a linear regression with log(Cry_ero) as the response; i.e., we would be assuming *Cryptomonas* density is lognormally distributed. However, we would be ignoring the fact that waterbodies were sampled multiple times (1-4 samples per waterbody), and it is likely that *Cryptomonas* abundance varies systematically between waterbodies. Therefore, we would likely violate the assumption of independently distributed data. To visualize this we can fit a simple linear regression and plot the residuals by waterbody:

```
crysub$resids = resid(lm(log(Cry_ero) ~ logTN, data = crysub))
ggplot(crysub, aes(WaterbodyID, resids)) + geom_boxplot() + theme(axis.
text.x = element_text(angle = 45, vjust = -1, hjust=1, size = 0.5))
```



Plotting boxplots for 322 different waterbodies is ugly, but it's clear that the waterbodies differ in residual abundance (if you want to be doubly sure, you could do an anova on the residuals).

How can we account for variation among waterbodies? We could add a factor for Waterbody in the model, but that would add 321 parameters to the model, resulting in a model with 323 parameters for 535 samples. This is mathematically possible, but generally a bad idea, and it will result in reduced power due to estimation of a large number of parameters relative to the sample size. In addition, if we want to include another factor in the model that varies at the level of waterbody, such as WaterbodyType (lake vs. reservoir), it won't work. We cannot simultaneously estimate 1) a factor that includes a separate mean for each Waterbody, and 2) the effect of WaterbodyType. This is impossible because these predictors would be perfectly collinear. This is a little tricky to explain, but imagine we had four waterbodies, and we tried to include a factor for WaterbodyID as well as a factor for lake vs. reservoir:

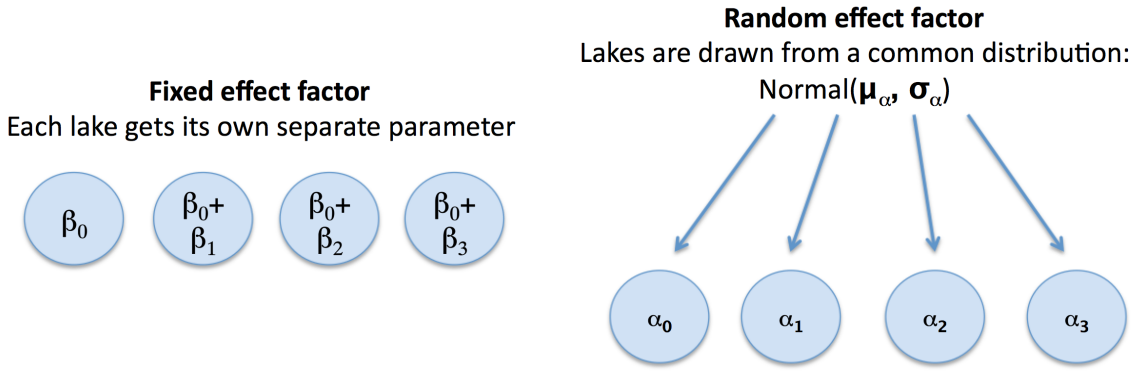


Here β_0 is the intercept (baseline), β_1-β_3 are the differences between each waterbody and the baseline waterbody, and β_{res} is the difference between reservoirs and lakes (I'm assuming that the first two waterbodies are lakes and the second two are reservoirs). The problem here is that we can't separately estimate β_2, β_3, and β_{res}. Maybe estimates of $\beta_2 = 1$, $\beta_3 = 2$, and $\beta_{res} = -1$ seem like good estimates. But I can change these to $\beta_2 = 2$, $\beta_3 = 3$, and $\beta_{res} = -2$, and when we add up the parameters we will end up with the same exact mean for each waterbody. In other words, we are trying to estimate more parameters than we have data for. The

general issue here is that there are many situations where we would like to model the variability among groups, while also asking what predicts the variation among those groups, and you typically can't do that with the standard LM/GLM framework.

The logic of random effects

Modeling variation in *Cryptomonas* abundance across waterbodies is a perfect case for using *random effects*. To start simple, imagine that all we care about is modeling the mean density of *Cryptomonas* in each waterbody. The logic of a random effect is this: rather than fitting a separate parameter for each waterbody, we will assume that the mean density for each waterbody is drawn from a common distribution (a normal distribution). *Then the parameters the model estimates are the mean and variance of the underlying distribution, rather than a separate mean for each waterbody*. Here is a schematic illustrating the idea:



The model on the left is the type we've been using so far, and is typically called a 'fixed effects' model, to distinguish it from a random effects model. It can be written:

$$\mu_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i}$$
$$Y_i \sim \text{Normal}(\mu_i, \sigma)$$

Where μ_i is the expected value for observation *i*; X_{ji} is an indicator variable for whether observation *i* is in waterbody *j* ($X_{ji} = 1$) or not ($X_{ji} = 0$); and the observation Y_i is drawn from a normal distribution with standard deviation σ.

The model on the right is a random effects model. There are different ways to write a random effects model mathematically, which can create confusion, but it is also helpful to look at these models from several angles to understand how they work. One way to write the model on the right is with similar syntax to a fixed effects model:

$$\mu_i = \beta_0 + b_1 Z_{1i} + b_2 Z_{2i} + b_3 Z_{3i}$$
$$b_j \sim \text{Normal}(0, \sigma_b)$$
$$Y_i \sim \text{Normal}(\mu_i, \sigma_Y)$$

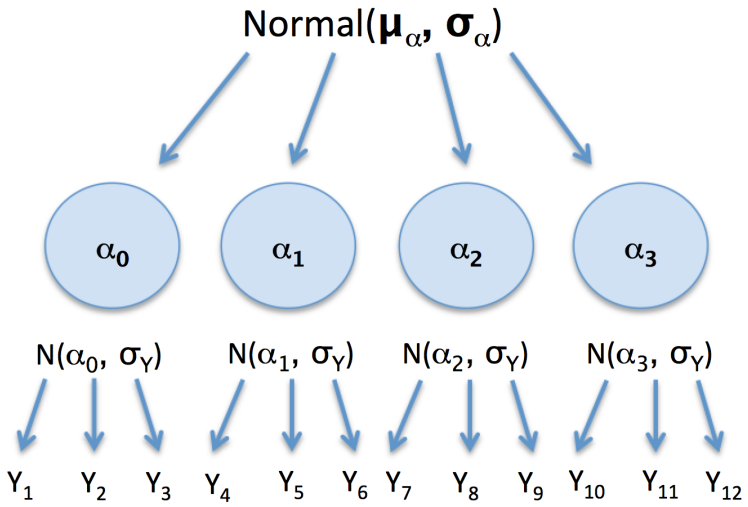
Now we use *Z*'s as the indicator variables, just to clarify that we are talking about random effects. The main difference here is that the b_j's are now constrained to come from a common normal distribution with mean 0 and standard deviation σ_b. Alternatively, we could write the model like this:

$$\alpha_j \sim \text{Normal}(\mu_\alpha, \sigma_\alpha)$$
$$Y_i \sim \text{Normal}(\alpha_{j[i]}, \sigma_Y)$$

This form says that each waterbody has its own mean density α_j, and these means are themselves normally distributed, with an underlying mean μ_α and standard deviation σ_α. Each observation within a waterbody is in turn randomly distributed with a mean $\alpha_{j[i]}$ and standard deviation σ_Y. The syntax $\alpha_{j[i]}$ just means "the group mean α_j that corresponds to observation *i*". So now we have parameters that correspond to multiple levels in the data (see the following figure). The parameters μ_α and σ_α are sometimes called "hyperparameters", because these upper-level parameters determine the distribution of lower-level parameters (the random effects α_j).

This kind of model is often called a *varying-intercept model* or *random intercept model*. The use of 'intercept' is a bit confusing here, because this model does not have any continuous predictors. Later we will see examples where we have a regression and the intercept of the line varies randomly among groups; for the current example the group means are also called 'intercepts' in R to maintain consistent terminology.

I've shown two ways (not the only ways) to write a simple random effects model, and they show different perspectives on what such a model is doing. It is important to note that these two versions are mathematically equivalent and would be fit with the same R syntax. The first version (with indicator variables) emphasizes that this is just like a linear model, but with a special constraint on the random effects parameters (they are randomly drawn from a common distribution). The second version emphasizes that we are now modeling stochastic variation on multiple levels. This can be further clarified with a diagram like this:



This helps clarify how we are thinking about stochasticity in this model: we are assuming that there is a hierarchy of random processes, where the group means (waterbodies) are randomly drawn from some distribution, and then the observations of phytoplankton abundance (Y_i) are themselves randomly drawn from a distribution with the corresponding waterbody mean. For this reason, models with random effects are often called *hierarchical models* or *multilevel models*. In biology the term *mixed-effects models* is more common, because the perspective is that random effects are typically added to a model that has other "fixed effects" predictors.

How to fit a varying-intercepts model with lmer

Now that we have some sense for the logic of a random effects model, let's fit one in R. To start, we'll just fit a model where *Cryptomonas* density varies between waterbodies. To make things easier to look at, I will use a subset of 100 observations from this dataset. If we were fitting a model where the factor WaterbodyID was a fixed effect, the syntax would look like this:

```
fixed.mod = lm(log(Cry_ero) ~ WaterbodyID, data = crysub1)
```

For mixed models we will use the function 'lmer' in the package lme4, and the syntax looks like this:

```
library(lme4)
rand.mod = lmer(log(Cry_ero) ~ 1 + (1|WaterbodyID), data = crysub1)
```

The random effect term for WaterbodyID is specified as (1|WaterbodyID). This model is saying "fit an intercept, i.e. the mean of the data, and then let that intercept vary randomly between waterbodies". As I mentioned earlier, this is called a varying-intercepts model because sometimes we will add in slope terms

that also vary by group. But for now the model only contains intercepts, i.e. terms that model the mean of the data. Let's see what summary says:

```
summary(rand.mod)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: log(Cry_ero) ~ 1 + (1 | WaterbodyID)
##    Data: crysub1
##
## REML criterion at convergence: 326.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.6857 -0.7575 -0.0365  0.6161  2.1561
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  WaterbodyID (Intercept) 0.352    0.593
##  Residual                1.213    1.101
## Number of obs: 100, groups:  WaterbodyID, 48
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)    5.035      0.142    35.5
```

At the top it says 'linear mixed model fit by REML'. We'll learn about REML later. For now I'll note that this is called a linear mixed model (LMM), because it is a linear model (i.e., linear terms with a normally distributed response) with a random effects term. Later we'll see how the same kind of model can be used with non-normal responses, and these are called generalizd linear mixed models (GLMMs).

Compared to fitting a linear model, we now have two sections for the model parameters, one for Random effects and one for Fixed effects. The random effects section will report the variance estimates for any random effects terms in the model. The fixed effects section will report the parameter estimates for any fixed effects terms in the model. In this case the model is the simplest possible mixed model; we just have one random effect term for WaterbodyID. The output says that we have 100 observations in 48 waterbodies. The variance estimate for WaterbodyID is 0.35; the standard deviation, which is just the square root of the variance, is 0.593. The random effects section also contains the estimate for the residual variance, which is 1.21.

Let's think about what these numbers mean. The standard deviation for the WaterbodyID term is 0.59. This means that the response, log(Cry_ero), is normally distributed across waterbodies with a standard deviation of 0.59. In other words, the typical difference between waterbodies in log(Cry_ero) is 0.59. The 'residual' variance in a mixed model is the variance among observations, after all the fixed and random effects have been accounted for. So in this model the residual variance

is the amount of variation between observations from the same waterbody (note that the model assumes each waterbody has the same within-waterbody variance). The residual variation has a standard deviation of 1.1, so the typical difference in log(Cry_ero) between observations taken in the same waterbody is 1.1. In the syntax I used earlier, the standard deviation for WaterbodyID is σ_α and the standard deviation for the residual variation is σ_Y.

With this simple model we can already see how random effects allow us to quantify different sources of variation in the data. In this case the model estimates say that the variation in *Cryptomonas* density *among lakes* is about half as much as the variation in density *within lakes* over time. So even though lakes vary in the abundance of this species, there is somewhat more variation over time within lakes, which is not too suprising considering the dramatic seasonal cycle in most temperate lake ecosystems.

The model also contains one fixed effect, labeled (Intercept). This parameter is the *mean density across all lakes*. In the syntax I used earlier, this parameter is μ_α, the mean of the distribution from which the WaterbodyID random effects are drawn. So the mean density is 5.04, and the standard deviation is 0.59. This hyperparameter (μ_α) is listed under fixed effects because it is not modeled as coming from some random distribution; it is just a number we want to estimate.

Random effects 'estimates'

So far this model has told us what the mean log(density) of *Cryptomonas* is, how much that density varies across waterbodies, and how much that density varies within waterbodies. Although we are modeling the variation across waterbodies as a random process, we would often like to know what the best estimates are for each waterbody. In a standard fixed effects model with WaterbodyID as a factor, we would get a parameter for each waterbody. When WaterbodyID is modeled using random effects, the only parameters that the model truly estimates are the three we just described: the mean density across waterbodies; the variance in density across waterbodies; and the variance in density within waterbodies.

One of the magical things about mixed models is that even though we are truly estimating the variance across groups (waterbodies), we can use the fitted model to calculate the 'best' estimates for each individual group. We can extract these from the model with ranef():

```
ranef(rand.mod)
```

```
## $WaterbodyID
##     (Intercept)
## 401   -0.315713
## 402    0.089169
```

```
## 403   -0.004237
## 404    0.667615
## 405   -0.388181
## 406   -0.331948
## 410   -0.477245
## 411    0.079690
## 501    0.060166
## 502    0.403596
## 503   -0.138564
## 504   -0.045408
…
```

I have abbreviated this output, which gives a random effect estimate for each waterbody. ranef() returns a list for all the random effects terms; in this case we have only WaterbodyID, and we have modeled the intercept as varying by this factor, hence the name of the column. The row names are the levels of WaterbodyID.

What are these numbers? Let's return to our random effects syntax. The model is assuming the the random effects have a distribution like this:

$$\alpha_j \sim \text{Normal}(\mu_\alpha, \sigma_\alpha)$$

where α_j is the mean density (aka the intercept) for waterbody *j*. We can rewrite this in two steps:

$$\eta_j \sim \text{Normal}(0, \sigma_\alpha)$$
$$\alpha_j = \mu_\alpha + \eta_j$$

I've just separated α_j into the mean across waterbodies (μ_α) and the random variate η_j (greek 'eta'), which is the difference between waterbody *j* and the overall mean. The η_j are normally distributed with a mean of 0 and a standard deviation of σ_α. This is an equivalent way to write the same model.

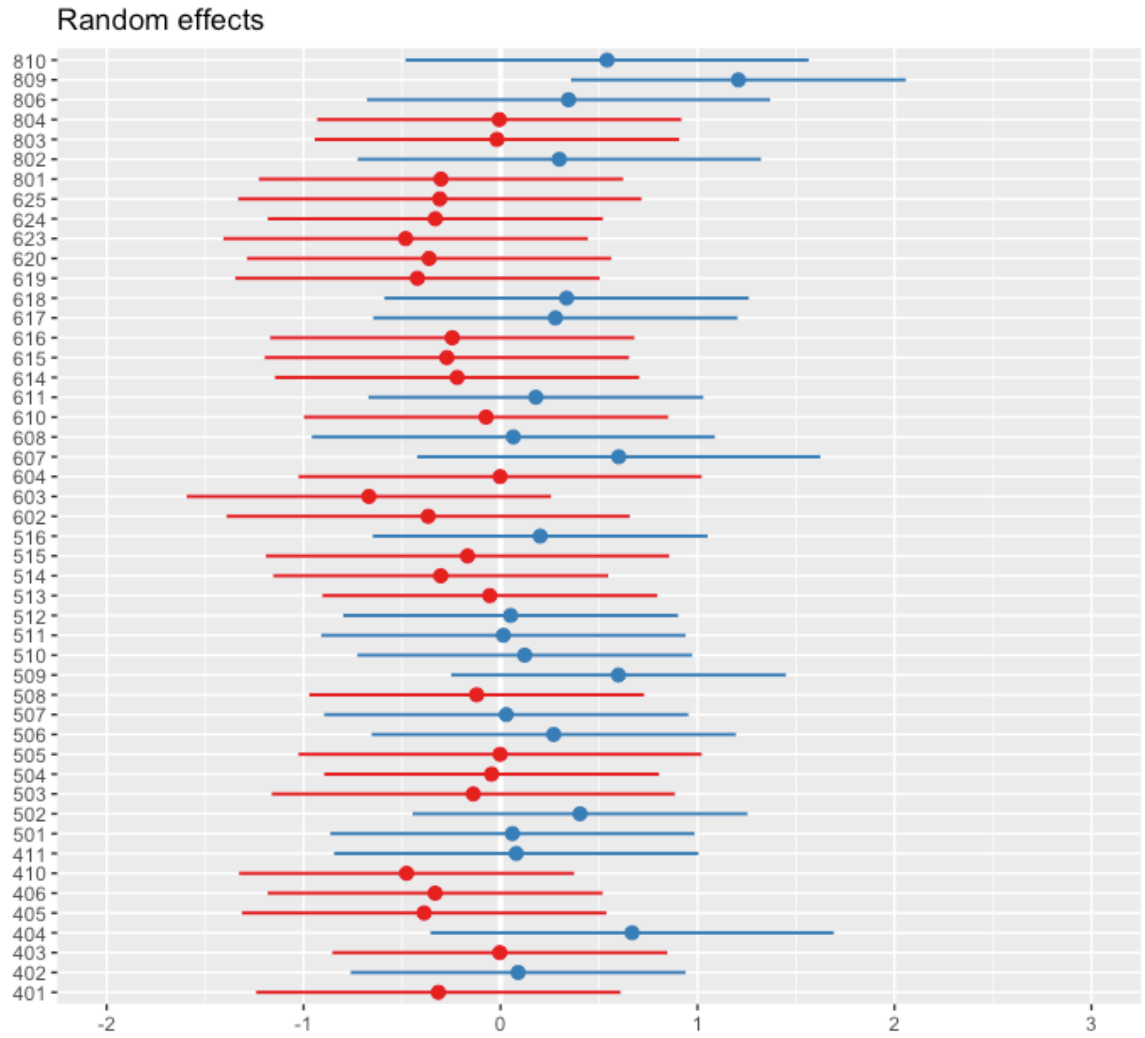
The function ranef() is returning the η_j for WaterbodyID. If we look at the help file for ranef(), it says that these numbers are the "conditional modes of the random effects…for a linear mixed model the conditional modes are also the conditional means". What are conditional modes? Essentially what we are asking here is "given the maximum likelihood estimates for the model parameters (μ_α, σ_α, and σ_Y), and given the observed data, what is the most likely value for η_j?" The most likely value is the peak of the probability density function, which is called the mode, and it is a conditional mode because it is conditional on the observed data.

I'm going through these seemingly arcane details just to emphasize that the random effects estimates are not really estimated parameters in the same sense as the parameters of a linear model. Nonetheless I will refer to them as 'random effects estimates' or 'random effects' for simplicity.

So now we know that ranef() is returning the random effect estimate for each waterbody, except that these estimates are centered around zero, so to get the full

estimate we would add in the fixed effect intercept (μ_α). The random effects can be visualized using the plot_model() function in the package 'sjPlot'.

```
plot_model(rand.mod, type = "re")
```



This function orders the random effects based on the order of the factor levels, and labels them on the y-axis. It's includes 95% CI by default. Here we aren't interested in any particular waterbody, so it's not that interesting to look at, but it gives us a sense for the variation in log(density) across waterbodies. Remember that these estimates are centered around 0. Clearly there's a lot of uncertainty in any particular random effect estimate; this makes sense, because we only have 1-3 observations per waterbody (more on that below).

Mixed effects diagnostics

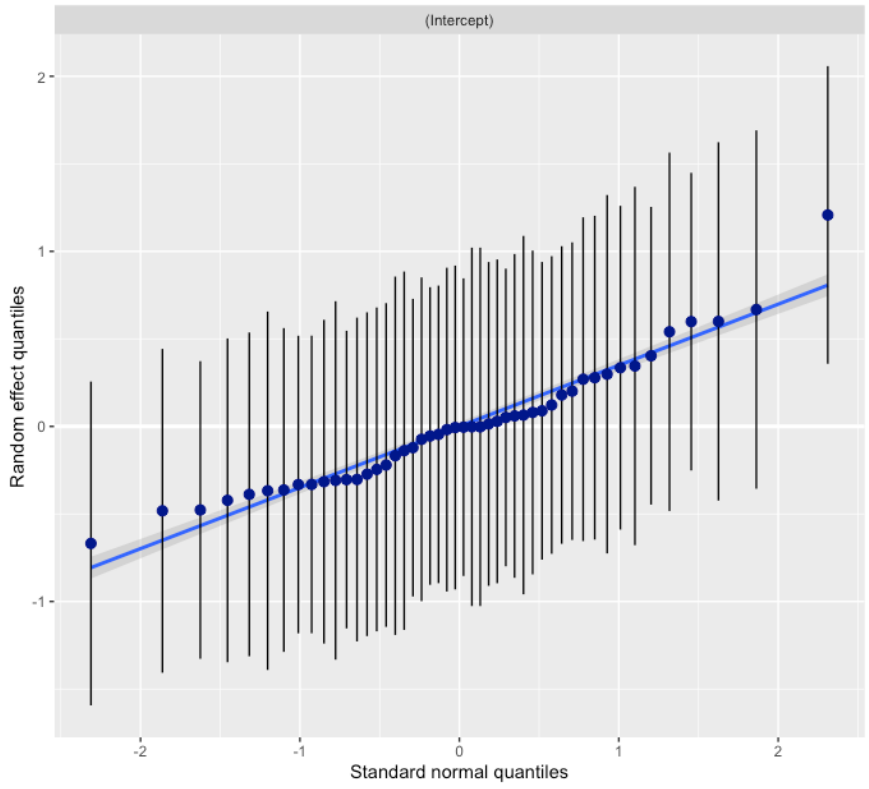
When we start putting random effects into models, diagnostics get more complicated. We'd like to plot residuals vs. fitted values, for example, but what exactly are the fitted values when the random effects are assumed to be random

variables? And what exactly are the residuals, now that we have multiple kinds of random variation in the model?

Now that we have multiple levels of random variation, one perspective is that we now have multiple levels of residuals. The η_j's returned by ranef() can be thought of as residual variation at the level of waterbodies. The other source of random variation is the residual variation within waterbodies. Alternatively, we may not want to think of this variation as 'residual error', but rather biologically meaningful variation that has various causes and which we are trying to estimate. In either case, we need to check whether the assumptions of the model are met.

The model assumptions for random effect terms are straightforward: we are assuming that the random effects are independently distributed, from a normal distribution. We can assess whether the normality assumption is met by making a quantile-quantile plot of the random effect estimates using plot_model() with the option 'diag' for diagnostics:

```
plot_model(rand.mod, type = 'diag')[[2]]
```

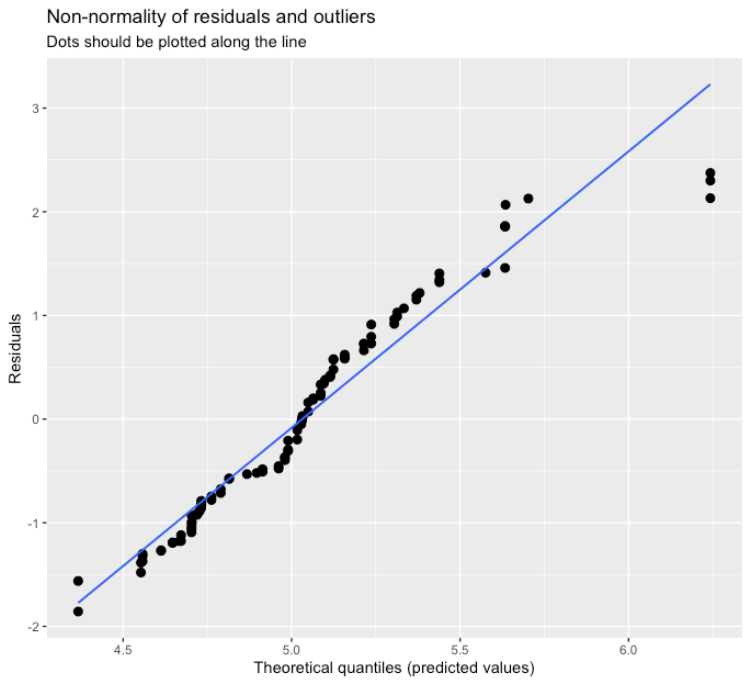


The distribution of the random effects looks pretty good. It should be noted that plotting the random effect estimates is a bit of a compromise. As I noted above, these are the conditional modes from the model. So to some extent they are model estimates, while also representing 'residual' variation. Nonetheless, we can inspect

them for an outliers or other egregious problems. We can also, in principle, make residuals vs. fitted plots for random effects, but in this model we do not have any predictors that would be explaining variation at the level of waterbodies.

In addition to the random effects, we also need to inspect the within-waterbody residuals. These can be extracted with the resid() function, and they are calculated as (observed data) – fitted(model). So in that sense they are the same as the residuals from a standard linear model. However, what does fitted() calculate for a mixed model? In order to make model predictions for each observation, we need to know both 1) the value of any fixed effects predictors for that observation, and 2) the value of any random effects that apply to that observation. Therefore, the random effects estimates (conditional modes) are again used to get the 'fitted' values for the model. Because the random effects estimates are not technically estimated by the model, this means that the residuals are not ideally behaved (similar to how GLM residuals are not ideally behaved), but nonetheless we can inspect them for egregious outliers and other strong patterns. They can also be plotted with plot_model():

```
plot_model(rand.mod, type = 'diag')[[1]]
```



Looks pretty good.

Likelihood for random effects models

Now we've gone through the basics of fitting, interpreting, visualizing and diagnosing a model with random effects. We can start applying the method to more interesting cases, but first there are some other features of random effects that you should be aware of.

As for the other models we've covered so far (GLMs, GAMs), mixed models are fit using maximum likelihood. The likelihood function for mixed models is more complex than the likelihood function for GLMs, and these models require specialized algorithms. Although we won't get into the gory details, it is useful to understand how the likelihood function for mixed models is different, because this will help us understand what these models are actually doing.

I've said several times that when we model a factor using random effects, we are not estimating a separate parameter for each group (e.g. each waterbody), but rather we are fitting the mean and the variance (hyperparameters) for the distribution from which the group means are drawn. So for our waterbodies example, our model will say that the mean *Cryptomonas* density across waterbodies is μ_α and the standard deviation across waterbodies is σ_α. The next question is, how do we find the maximum likelihood values for these two parameters? Recall that the likelihood function is found by calculating the joint probability density for the data, as a function of the values of the parameters; we often abbreviate the parameters as the vector θ. Let's say that observation Y_1 is taken from lake 3. The probability density for observation Y_1 would be:

$$P(Y_1|\theta) = \text{dnorm}(Y_1, \text{mean} = \alpha_3, \text{sd} = \sigma_Y)$$

Here I've used R syntax to abbreviate the probability density function for the normal distribution. Because observation Y_1 is taken in lake 3, that means the probability of observing this value depends on the mean density for that lake (α_3), as well as the residual standard deviation σ_Y, which determines how much within-lake variation there is. The trick here is that we don't know want to estimate α_3 as a separate parameter, rather we want to estimate the hyperparameters of the distribution from which α_3 was drawn: μ_α and σ_α. But at the same time, *we need a value of α_3 to calculate the likelihood because the probability of Y_1 depends on α_3*.

The clever solution to this quandary is to treat α_3 as an unknown quantity, and evaluate the likelihood for all possible values of that quantity. In other words, we say "I don't know what α_3 is, and I don't really want to know, so I will integrate (average) the likelihood calculation across all possible values of α_3". And we will do the same thing for all the random effects, i.e. all the α_j's. So we can think about the process like this:

- The likelihood of Y_i depends on $\alpha_{j[i]}$
- But $\alpha_{j[i]}$ is a random number whose probability depends on μ_α and σ_α

- So we get a likelihood for the hyperparameters by averaging the probability of observing Y_i for all possible values of $\alpha_{j[i]}$

In mathematical terms the likelihood function looks like this:

$$L(\mu_\alpha, \sigma_\alpha, \sigma_Y | Y) = \int \prod_i^N P(Y_i | \mu_\alpha, \sigma_\alpha, \sigma_Y, \vec{\alpha}) \, d\,\vec{\alpha}$$

We want the likelihood of some values of the parameters $\mu_\alpha, \sigma_\alpha, \sigma_Y$, conditional on the data Y. This is equal to the joint probability density of the data, conditional on the parameter values. But the probability density of the data also depends on the random effect estimates (I've summarized the α_j's in one vector $\vec{\alpha}$), so we integrate the probability density across all possible values for the random effects $\vec{\alpha}$. Because of this integration, this is called a *marginal likelihood*.

I realize this is probably a somewhat mystifying idea, but hopefully it gives you some sense for the fact that the random effects are necessary for fitting the model, while at the same time not being estimated in the same way that a typical 'fixed effect' parameter is, because in reality they are treated as unknown values that are averaged over in the likelihood calculation.

Shrinkage / partial pooling

There is an important difference in the way fixed effects and random effects are estimated. Let's compare the model I fit earlier:

```
rand.mod = lmer(log(Cry_ero) ~ 1 + (1|WaterbodyID), data = crysub1)
```

with an equivalent model that fits WaterbodyID as a fixed effect term:

```
fixed.mod = lm(log(Cry_ero) ~ WaterbodyID, data = crysub1)
```

fixed.mod fits a parameter for each waterbody, as expected:

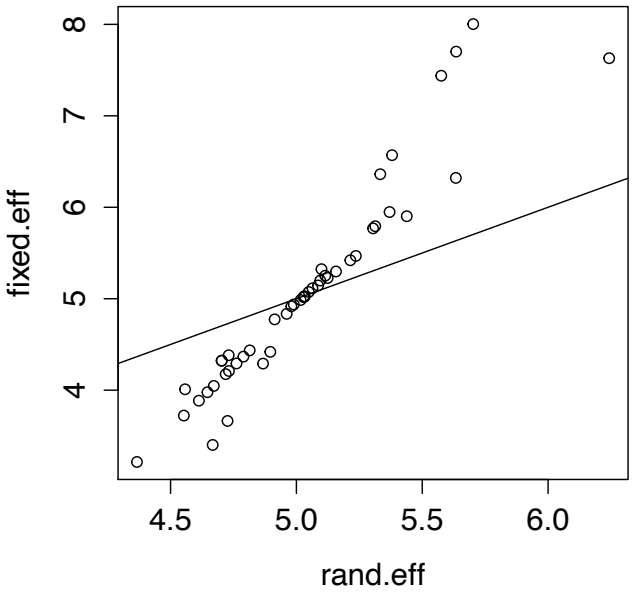
```
Call:
lm(formula = log(Cry_ero) ~ WaterbodyID, data = crysub1)

Residuals:
    Min      1Q  Median      3Q     Max
-2.0725 -0.4576  0.0000  0.3828  2.1684

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    4.17521    0.75564   5.525 1.07e-06 ***
WaterbodyID402 1.05126    0.97552   1.078 0.286170
WaterbodyID403 0.85056    0.97552   0.872 0.387267
WaterbodyID404 3.82781    1.30880   2.925 0.005099 **
WaterbodyID405 -0.19733   1.06863  -0.185 0.854219
WaterbodyID406 0.14644    0.97552   0.150 0.881259
WaterbodyID410 -0.16575   0.97552  -0.170 0.865737
WaterbodyID411 1.07666    1.06863   1.008 0.318353
WaterbodyID501 1.02350    1.06863   0.958 0.342614
WaterbodyID502 1.72685    0.97552   1.770 0.082560 .
WaterbodyID503 0.24363    1.30880   0.186 0.853056
```

This is abbreviated output from summary(). So now we have a model that fits a separate parameter for the mean of each waterbody, and a random effects model from which we can also extract estimates for the mean of each waterbody. Are they the same? Let's compare:

```
fixed.effects = effect("WaterbodyID", fixed.mod)
fixed.eff = fixed.effects$fit[,1]
rand.eff = ranef(rand.mod)$WaterbodyID[,1] + fixef(rand.mod)[1]
```

To get the fixed effects, I use the effect() function in the effects package (otherwise I would have to add all the model coefficients to the baseline intercept, etc). And I extract the corresponding random effects using ranef(), and add them to the overall mean, which is extracted using fixef(). Now we can plot them to compare:

```
plot(fixed.eff ~ rand.eff)
abline(a = 0, b = 1)
```



The line is the 1:1 line. Clearly the estimates are strongly correlated, but they differ systematically in magnitude. In particular, the fixed effects estimates have a greater overall spread, and are either higher than the random effects values (for values greater than the mean) or lower than the random effects values (for values lesser than the mean).

What's going on here? In a fixed effects model, the only information used to estimate the mean density in a lake are the values from that lake. So if we observe values of 4.5, 4.7, and 5.1 in lake #401, then the model-estimated mean for that lake will be (4.5 + 4.7 + 5.1)/3 = 4.77. In contrast, *a random effect estimate depends on data from that lake as well as the data from the other lakes*. Why is that? The key difference is that random effects are assumed to come from a common distribution. That means that all the data from all the lakes informs the model estimates for the mean and variance (μ_α and σ_α), and when we go to calculate the conditional modes with ranef(), those random effect estimates depend both on the data in that waterbody, as well as the model-fitted mean variance.

In our fitted random effects model, the estimated mean across lakes was 5.04 and the estimated standard deviation was 0.59. Now let's say there's a lake where we only have two observations, 2.4 and 4.6, for a mean of 3.5. This is a highly unlikely value if the mean of the lakes is 5.04 and the standard deviation is 0.59. In fact, the probability that a lake would have a mean this extreme is 0.005. When we use the fitted model to calculate the random effect for this lake, the math essentially says "The mean for this lake is low, but probably not as low as the data says, because there were only two observations, and those could have just been two weird

observations. So we will upwardly revise the estimated mean for this lake, to balance the evidence from that lake with the model-estimated mean and variance across lakes." Of course this is not actually how the process works; rather the random effect estimate is the 'best' based on the information in the model, and the model assumptions.

Random effects estimates are often called 'shrinkage' estimates, because they will be shrunk towards the overall mean, as I just explained. Andrew Gelman calls this 'partial pooling', because the information in the data is partially pooled across groups. The extent of the shrinkage depends on the sample size within each group. For the simple varying-intercepts model I've fit, the estimated mean for a waterbody follows this approximation:

$$\hat{\alpha}_j \approx \frac{\frac{n_j}{\sigma_Y^2}\bar{Y}_j + \frac{1}{\sigma_\alpha^2}\bar{Y}_{all}}{\frac{n_j}{\sigma_Y^2} + \frac{1}{\sigma_\alpha^2}}$$

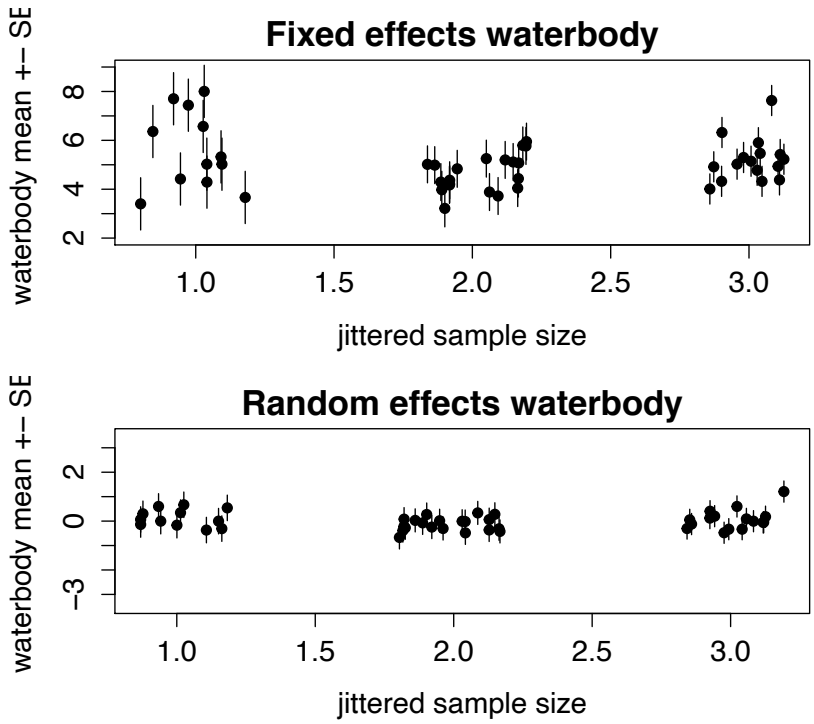
where $\hat{\alpha}_j$ is the random effect estimate (mean density) for waterbody j, n_j is the number of observations in waterbody j, $\bar{Y}_j$ is the mean density of the observations in waterbody j, and $\bar{Y}_{all}$ is the mean density across all observations. What this means is that when sample size within a group (n_j) is small, the random effect estimate for that group will be shrunk towards the overall mean, but as sample size within a group gets large, the shrinkage is weak and the estimate converges on the same estimate you would get from a fixed effects model.

In the previous plot of fixed effect estimates vs. random effects estimates, the fixed effects estimates have a much larger spread because the random effects estimates are based on those same numbers, but shrunk towards the overall mean, to an extent that depends on the sample size within each waterbody.

This whole business of the estimate for a waterbody depending on data from other waterbodies may seem suspicious. Won't this lead to biased results? To estimate what's happening in a lake, shouldn't we only use information from that lake? As it turns out, if the model assumptions are met (i.e. if the group means are truly normally distributed), then the random effects model will indeed be mildly biased in the sense of shrinking estimates towards the mean, but it will also be more accurate. To convince you (and myself) of this, let's look at some real data and then a simulation.

The example we are looking at with phytoplankton in lakes is as extreme as it gets for shrinkage, because each lake only has 1-3 observations, and so the model has relatively little confidence in the data from any particular waterbody. We can see the influence of sample size in this example:

```
xvals = as.vector(jitter(sample.size))
plot(xvals, fixed.effects$fit[,1], pch = 19, xlab = 'jittered sample size', yl
ab = 'waterbody mean +- SE', ylim = c(2,9), main = 'Fixed effects waterbody')
segments(xvals, fixed.effects$fit[,1] + fixed.effects$se, xvals, fixed.effects
$fit[,1] - fixed.effects$se)

library(arm)

xvals = as.vector(jitter(sample.size))
plot(xvals, ranef(rand.mod)$WaterbodyID[,1], pch = 19, xlab = 'jittered sample
 size', ylab = 'waterbody mean +- SE', ylim = c(-3.5,3.5), main = 'Random effe
cts waterbody')
segments(xvals, ranef(rand.mod)$WaterbodyID[,1] + se.ranef(rand.mod)$Waterbody
ID[,1], xvals, ranef(rand.mod)$WaterbodyID[,1] - se.ranef(rand.mod)$WaterbodyI
D[,1])
```



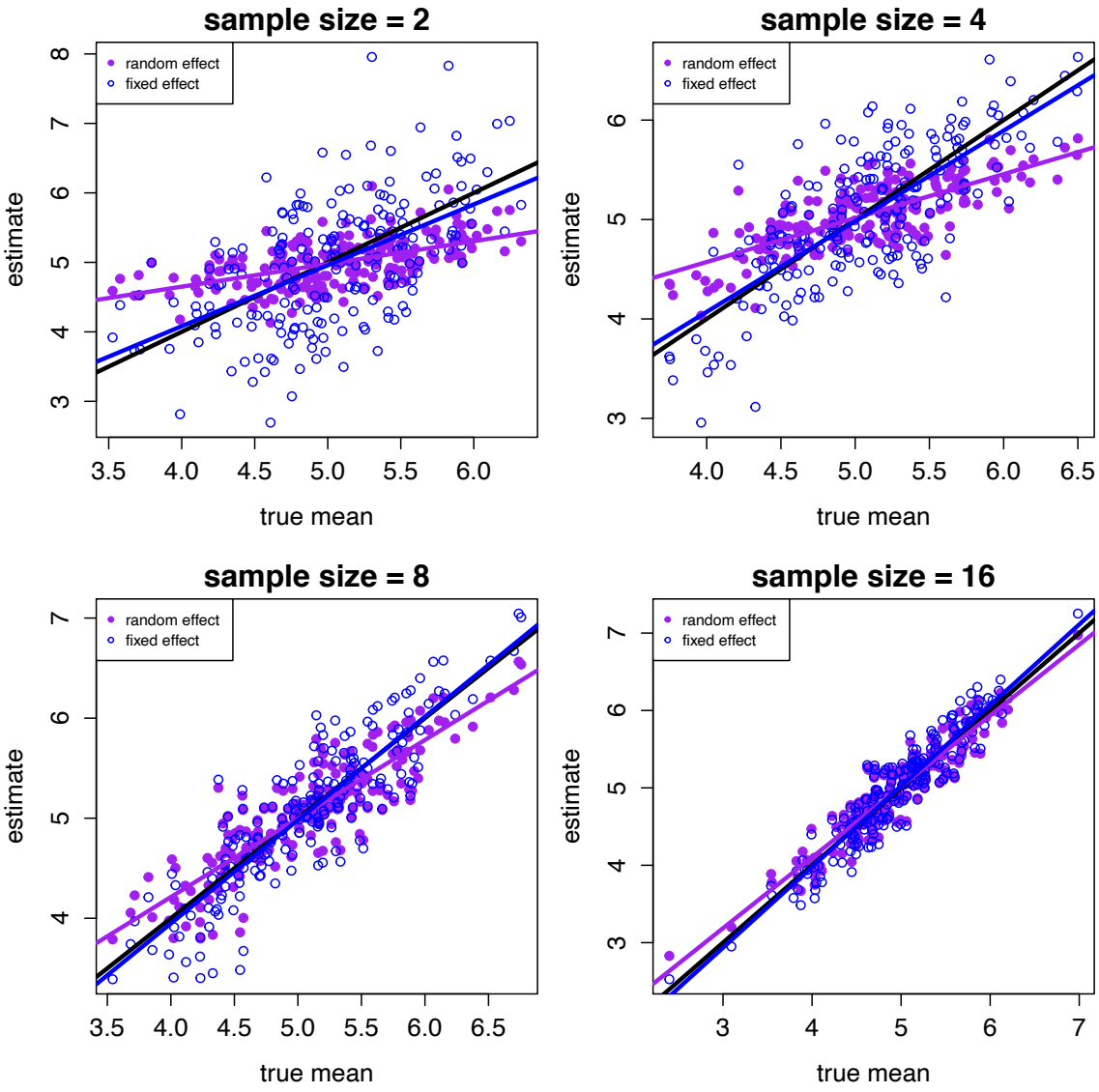
Here I've plotted the model estimate for each waterbody, +/- standard error, for both the fixed effects and random effects model. And I've arranged these estimates by the sample size within that waterbody. Sample size is jittered on the x-axis to make the plot legible.

The most notable pattern for the fixed-effects model is that the spread of estimates is larger for the fixed effects model when sample size = 1. This isn't surprising; with a single observation in a waterbody, we expect to get pretty variable estimates of mean density in that waterbody. This variability is reduced for sample size = 2, and again for 3. In contrast, the variability of estimates in the random effects model does not depend on sample size. That's because waterbodies with less data get their estimates shrunk towards the mean more strongly. In this respect, the results from the random effects model are more realistic, because we know that the real

variation in waterbodies has no relation to how many samples we collected. In addition, there is less variability overall for random vs. fixed effects estimates, because the model has only modest confidence in an estimate based on n = 3. This also seems likely to be more accurate.

A shrinkage simulation

While writing this lecture I was interested to see just how much the shrinkage of random effects estimates causes bias vs. increased accuracy. So I coded up a simulation where 1) 200 imaginary lakes were sampled, and each lake gets a mean *Cryptomonas* density, drawn from a normal distribution with mean = 5 and sd = 0.6. 2) Each lake is sampled N times (I varied N from 2 to 16). Then I fit a fixed effects model and a random effects model to the simulated data, to see which one does a better job of estimating the true lake means. Here's a plot comparing both the fixed effect estimates and the random effect estimates for each lake to the true lake means. This comparison was made for N = 2, 4, 8, and 16.



The random effect estimats are in purple, and the fixed effects are in blue. The black line is 1:1 (estimates = the true values), and the blue and purple lines show how the estimates compare to the true values on average.

For sample size = 2, it is clear that the fixed effects estimates are noisier than the random effects estimates. On average the random effects estimates are definitely closer to the true values (the black line). At the same time, it is clear that there is some bias in the random effects estimates. The purple regression line shows that large (true) deviations tend to be underestimated. If the true value is 6, it will be shrunk downwards to about 5.2 on average; if the true value is 4, it will be shrunk upwards to about 4.6 on average. However, even when these extreme values are biased towards the mean, the mean squared error is still not larger, compared to the fixed effects model. So one way to think about this is as a *tradeoff between bias and accuracy*. The random effects model achieves greater accuracy at the cost of

some bias. This is another version of the overfitting-underfitting tradeoff we have discussed. The fixed effects model overfits the data (treats big deviations with small sample sizes as real effects), while the random effects model achieves better predictive power by underfitting the most extreme values.

As the sample size is increased from 2 to 4 to 8, the same phenomenon is present, but the difference between fixed and random effects estimates diminishes. At sample size 16 it is virtually gone, and there is little shrinkage. At this point the main difference between the two approaches is that random effects models can better incorporate different kinds of complexity, as we will continue to discuss. So it would seem that the shrinkage of random effects is really only relevant at very small sample sizes (per group), but these are relatively common in biological data, and in addition sample size per group can be very heterogeneous in observational data.