# ENTMLGY 6707 Entomological Techniques and Data Analysis

## R Activity 8: Linear Mixed-Effects Models

For the linear models we have worked with so far, a key assumption for each was that the residuals/observations were independent. When analyzing ecological data, we often have to contend with correlated data. For example, if we have samples from a few different sites, observations from a given site tend to be more similar in value to one another than observations at different sites.

Mixed-effects (ME) models provide a very useful tool for analyzing correlated data. We will use the `lmer()` command from the `lme4` package to fit ME models. The syntax for `lmer()` is quite similar to the `lm()` command, except the `lm()` models only included predictors fit as "fixed effects". We will now be fitting random effects - random intercepts and random slopes. Hence, MIXED effects models include both fixed and random effects. For a standard regression model:

$Y_{ij} = \beta_0 + \beta_1 X_{ij} + \epsilon_i$

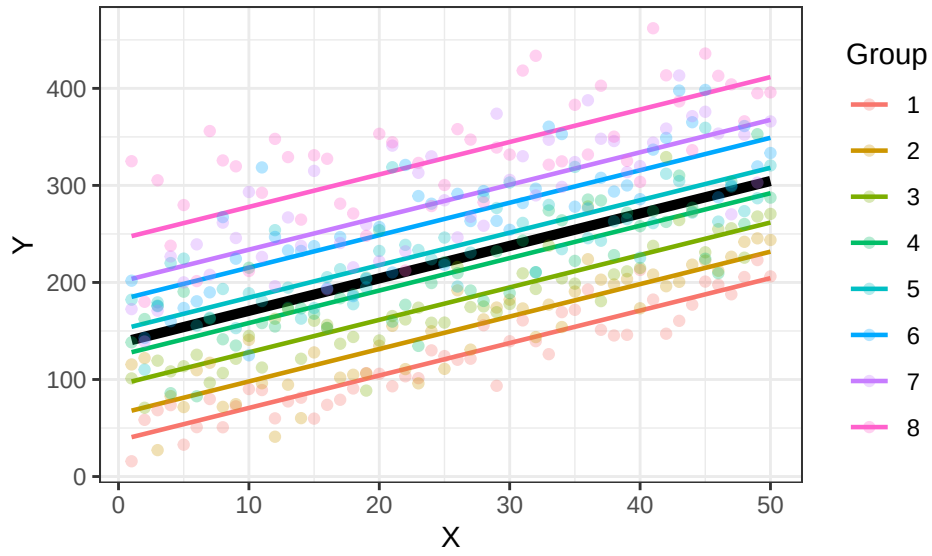
We can add a random intercept, $b_{0j}$, as follows: $Y_{ij} = \beta_0 + \beta_1 X_{ij} + \epsilon_i + b_{0j}$

Which could also be written: $Y_{ij} = (\beta_0 + b_{0j}) + \beta_1 X_{ij} + \epsilon_i$

where $b_i$ is an additive shift for each level of a grouping variable (e.g., sites, plots, branches, etc.) compared to the "global intercept". That is, we get a single, best fit line for all the data, and then each grouping variable gets a unique intercept estimated for it. However, instead of estimating the mean $\pm$ standard error of each $b_i$ (like we would in simple linear regression), we are just estimating the variance *among* those intercept values.

This creates an additional assumption that the random intercepts are normally distributed with a mean of 0 and variance $\tau^2$: $b_{0j} \sim N(0, \tau^2)$. The distributional assumption makes sense: each $b_{0j}$ is either added or substracted to the global intercept, $\beta_0$. Indeed, you can think of each $b_i$ as a "residual" for $\beta_0$, and so there will be $b_{0j}$ values greater and smaller than 0, but they should "average out" to 0; $\beta_0 + 0 = \beta_0$ .

The black line below indicates the line associated with $\beta_0$ and $\beta_1$ whereas each colored line is associated with a unique $b_{0j}$. Each colored line intercepts the y-axis at $\beta_0 \pm b_{0j}$, but all lines have a slope = $\beta_1$.




# 1 Fitting linear mixed-effects models

Load the `lme4` package into the current R session.

```
library(lme4)
```

Here is a general layout of a ME model. It should look familiar. The additional `(1|Level1/Level2/Level3)` term is the random intercept. In this case, we are saying we have a "nested" data structure, such that `Level1` is nested within `Level2` which is nested within `Level3`.

```
fit_example1 <- lmer(Response ~ Predictor + (1|Level3/Level2/Level1), data=made_up_data)
summary(fit_example1)
```

In more biological terms, here we have a model of `Tree.height` as a function of `DBH` (diameter at breast height) and we happened to measure trees on several plots across multiple sites in multiple regions.

```
fit_example2 <- lmer(Tree.height ~ DBH + (1|Region/Site/Plot), data=made_up_tree_data)
summary(fit_example2)
```

The `Orange` data (which gets loaded in with `lme4`) has multiple measurements across 5 different trees at different values of `age`. We will account for these repeated meaures by fitting random intercept terms. You might notice there are no $p$-values in the below output (we'll get to that!).

```
lmer_gaussian <- lmer(circumference~age  + (1|Tree), data=Orange)
summary(lmer_gaussian)
```

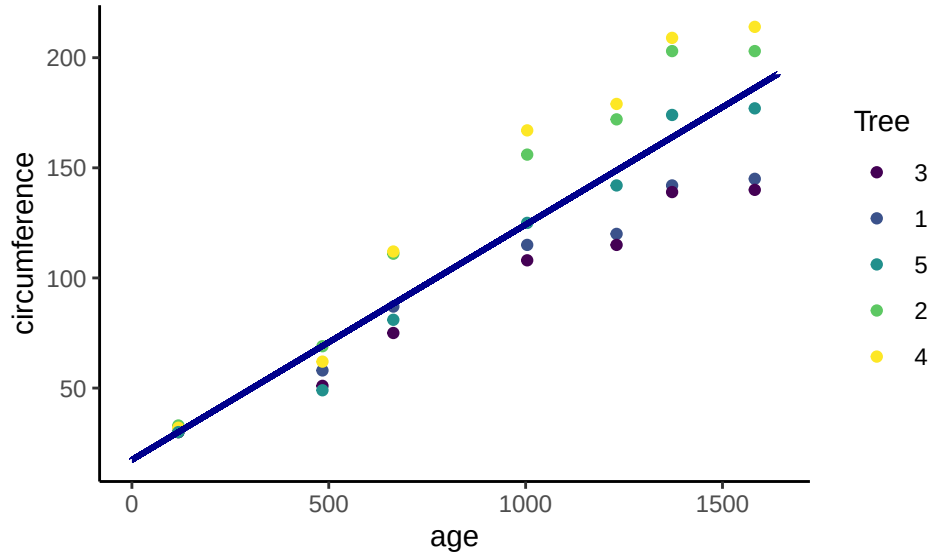
```
## Linear mixed model fit by REML ['lmerMod']
## Formula: circumference ~ age + (1 | Tree)
##    Data: Orange
##
## REML criterion at convergence: 303.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8781 -0.6743  0.2320  0.5053  1.5416
```

```
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Tree     (Intercept) 389.6    19.74
##  Residual             232.9    15.26
## Number of obs: 35, groups:  Tree, 5
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept) 17.399650  10.423696   1.669
## age          0.106770   0.005321  20.066
##
## Correlation of Fixed Effects:
##     (Intr)
## age -0.471
```

## 2 Adding an `lmer()` line to a graph

We plot the "global model" when presenting fits from ME models. The `re.form=NA` argument in the code below essentially tells R to avoid trying to plot the line for each grouping variable. Thus, `Tree` IDs are ignored in the plot but not the model fitting process.

```
new_data <- data.frame(age = seq(0, 1640, 0.01))
new_data$Predicted_lmer  <- predict(lmer_gaussian,
                                    newdata=new_data, re.form=NA)
ggplot(data=Orange, mapping=aes(x=age, y=circumference, col=Tree))+
        geom_point()+theme_classic()+
        geom_line(data=new_data, aes(x=age, y=Predicted_lmer),
                  color="dark blue", linewidth=1)
```




## 3 Interpretation

As for the `Estimates` of the intercept and slope, you can interpret them equivalently to regression models that contain only fixed-effects.

# 4 *p*-values and degrees of freedom

A complete explanation for the absence of *p*-values in the `lmer()` output is beyond the scope of this course. Briefly, since we are assuming there is some structure of correlation in the residuals of ME models, it is unclear how to calculate degrees of freedom (e.g., how much should two observations on the same plot or site contribute to degrees of freedom?...since such observations are not truly independent). For that reason, *p*-values are not provided out of the box in `lme4` (i.e., it was an intentional ommission by the authors of the package). You can load the package `lmerTest` to get degrees of freedom and *p*-values, which uses Satterthwaite's method to estimate the degrees of freedom as the default.

```
library(lmerTest)
```

# 5 ANOVA

This tutorial has so far discussed a model evaluating the effects of a continuous predictor (`age`) on a continuous response (`circumference`). Linear mixed-effects models can also be used to quantify the effects of a categorical predictor (they can also be used to analyze response variables that follow a Poisson or binomial distribution, but that will be covered later). For now, pretend we want to fit `age` as a fixed-effect after converting it to a factor (in practice, I don't recommend converting a continuous predictor into a cateogrial one without a compelling reason). I won't cover this in detail here, but the interpretation of these models (i.e., `y ~ categorical x`) is generally the same as when fitting a fixed-effects only ANOVA (which you did earlier in the course).

Important: we are working with one factor here, but when working with multiple factors, the `anova()` command will give you very different output before and after loading the `lmerTest` package. I recommend always loading in the `lmerTest` before running `anova()` on mixed-effects models, as R will then use the `lmerTest` version of `anova()` (there is a `base R` version too) when the command is fed a `lmer()` model. Notice below that this `anova()` command let's you input the "type" of sums of squares (`type=3`). When working with `lm()` models and `anova()`, we have to load in the `car` package and use `Anova(fit, type="III")`.

```
lmer_gaussian_fact <- lmer(circumference~factor(age)+ (1|Tree), data=Orange)
anova(lmer_gaussian_fact, type=3)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##             Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## factor(age)  96051   16008     6    24   85.86 5.064e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(lmer_gaussian_fact)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: circumference ~ factor(age) + (1 | Tree)
##    Data: Orange
##
## REML criterion at convergence: 248.2
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.53450 -0.65554  0.05986  0.72895  1.42666
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Tree     (Intercept) 396.3    19.91
```

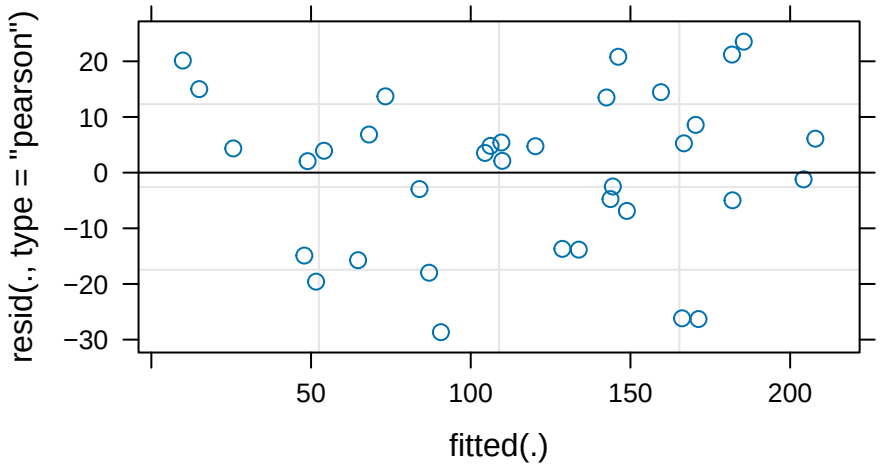
```
##  Residual             186.4    13.65
## Number of obs: 35, groups:  Tree, 5
##
## Fixed effects:
##                 Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)       31.000     10.795   7.418   2.872  0.02249 *
## factor(age)484    26.800      8.636  24.000   3.103  0.00485 **
## factor(age)664    62.200      8.636  24.000   7.202 1.92e-07 ***
## factor(age)1004  103.200      8.636  24.000  11.950 1.36e-11 ***
## factor(age)1231  114.600      8.636  24.000  13.270 1.52e-12 ***
## factor(age)1372  142.400      8.636  24.000  16.489 1.36e-14 ***
## factor(age)1582  144.800      8.636  24.000  16.767 9.38e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) f()484 f()664 f()100 f()123 f()137
## factr(g)484 -0.400
## factr(g)664 -0.400  0.500
## fctr(g)1004 -0.400  0.500  0.500
## fctr(g)1231 -0.400  0.500  0.500  0.500
## fctr(g)1372 -0.400  0.500  0.500  0.500  0.500
## fctr(g)1582 -0.400  0.500  0.500  0.500  0.500  0.500
```

# 6 Assumptions

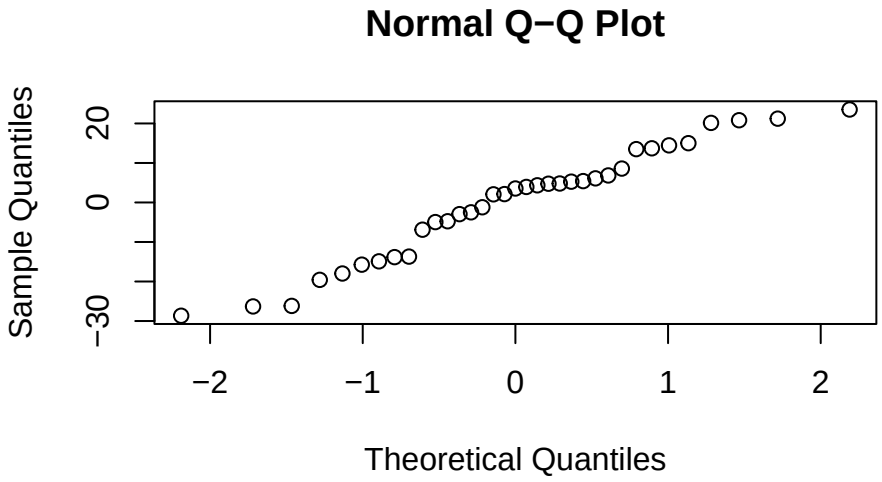
The assumption of independence is "relaxed" a bit, given we are accounting for the lack of independence by fitting random effects. Thus we are assuming that we have done a sufficient job identifying and fitting the variables that induce correlation among the residuals, which requires knowledge about the experimental design and study system.

Otherwise, linear mixed-effects models are linear models. So, the assumptions are mostly the same as when you fit a linear regression or an ANOVA. For example, if you are fitting a mixed-effect ANOVA with `lmer()`, you should assess the assumptions of normality and homoscedasticy. You can check the assumptions using very similar commands:

```
plot(lmer_gaussian)
```

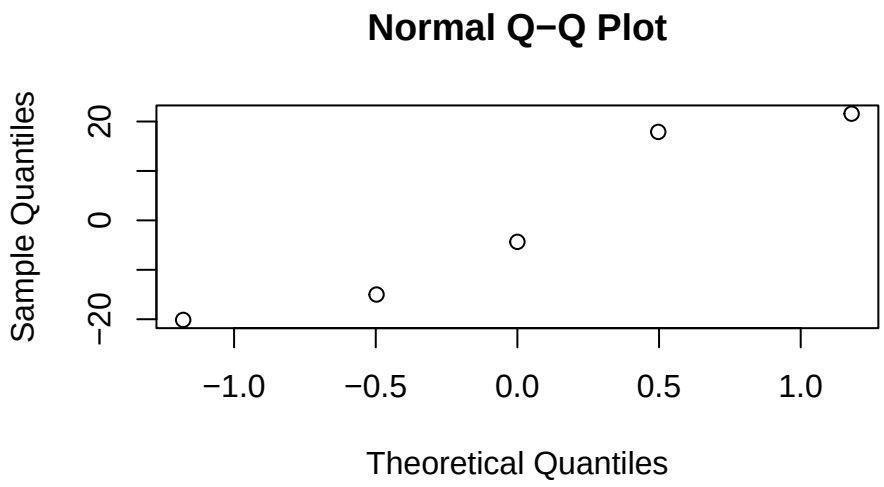



```
qqnorm(residuals(lmer_gaussian))
```

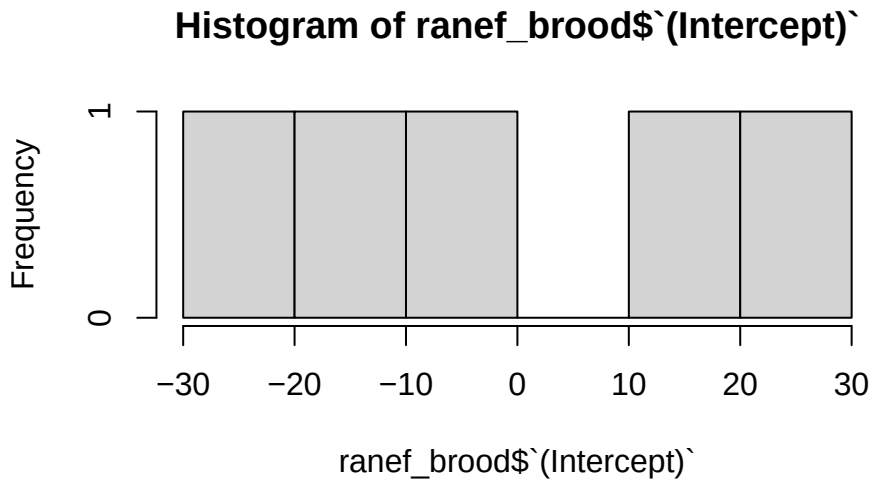



One additional assumption is that the random effects are normally distributed. You can check this as follows:

```
# checking normality of tree random effect
ranef_brood <- ranef(lmer_gaussian)$Tree
qqnorm(ranef_brood$`(Intercept)`)
```




```
hist(ranef_brood$`(Intercept)`)
```




These extra assumptions are often violated in practice. Luckily, ME models are robust to such violations (Schielzeth et al. 2020). There has been some misapplication of ME models, and Silk et al (2020) provide a nice discussion of common pitfalls and how to avoid them.

# 7 Random slope models

Lastly, we can also fit models with random slopes. Thus, ME models could include (i) multiple random intercepts (e.g., nested vs. non-nested), (ii) multiple random slopes, and (iii) multiple random intercepts and slopes. I stick with random intercepts in most cases, but repeated measures on an individual (e.g., longitudinal data) often require random slopes. These instances are beyond the scope of this class, but there are several tutorials online that you can follow if you have a somewhat complicated design and need to fit a more complicated model. The random slope term, $b_{1j}$, is an additive shift to our global slope, $\beta_1$, in an analogous way to adding $b_{0j}$ to $\beta_0$.

$Y_{ij} = (\beta_0 + b_{0j}) + (\beta_1 + b_{1j})X_{ij} + \epsilon_i$

where again we assume the random effects are normally distributed with variances $\tau_1^2$ and $\tau_2^2$:
$b_{0j} \sim N(0, \tau_1^2)$
$b_{1j} \sim N(0, \tau_2^2)$

This model has a random intercept for `Tree` and a random slope for `age` in each `Tree`. This specification indicates that we think the `circumference~age` relationship is slightly different for each tree, but we are viewing this as more of a nuisance rather than an interesting ecological property. Indeed, if we were interested in the effect of `Tree`, we should fit it as a fixed effect (and potentially an interaction).

```
lmer_gaussian_slope <- lmer(circumference ~ age + (age|Tree), data=Orange)
summary(lmer_gaussian_slope)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: circumference ~ age + (age | Tree)
##    Data: Orange
##
## REML criterion at convergence: 281.1
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.09099 -0.50176 -0.07625  0.71181  1.63662
##
## Random effects:
##  Groups   Name        Variance  Std.Dev. Corr
##  Tree     (Intercept) 8.312e+00  2.88310
##           age         5.083e-04  0.02255 0.99
##  Residual             1.016e+02 10.07726
## Number of obs: 35, groups:  Tree, 5
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept) 17.39965    3.88098  7.11618   4.483  0.00274 **
## age          0.10677    0.01068  3.38479   9.999  0.00125 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##     (Intr)
## age 0.037
## optimizer (nloptwrap) convergence code: 0 (OK)
## unable to evaluate scaled gradient
## Model failed to converge: degenerate  Hessian with 1 negative eigenvalues
```

The above model has some convergence issues (see the last line of the output)! You may run into these problems, like perfect correlations between your random slopes and intercepts.

When we take repeated measures on organisms, we often expect the slopes to differ as well. Not properly accounting for the variation in slopes can lead to inflated type I error rates (Shielzeth et al. 2009), but when analyzing ecological data, models can also suffer from near perfect correlation between random intercepts and random slopes. What to do in such instances is case-specific. If I ran into this issue here, I would probably report the random slope model (the one with convergence issues) and just mention there were some conversion issues BUT that the model output was equivalent to the model with just a random intercept. I like the random slope model in this case, despite the convergence issues, because it properly accounts for the repeated measures on trees.

# 8 References


- Arnqvist, G. 2020. Mixed Models Offer No Freedom from Degrees of Freedom. Trends Ecol. Evol. 35: 329–335.
- Harrison, X. A., L. Donaldson, M. E. Correa-Cano, J. Evans, D. N. Fisher, C. E. D. Goodwin, B. S. Robinson, D. J. Hodgson, and R. Inger. 2018. A brief introduction to mixed effects modelling and multi-model inference in ecology. PeerJ. 6: e4794.
- Schielzeth, H., N. J. Dingemanse, S. Nakagawa, D. F. Westneat, H. Allegue, C. Teplitsky, D. Réale, N. A. Dochtermann, L. Z. Garamszegi, and Y. G. Araya-Ajoy. 2020. Robustness of linear mixed-effects models to violations of distributional assumptions. Methods Ecol. Evol. 11: 1141–1152.
- Schielzeth, H., and W. Forstmeier. 2009. Conclusions beyond support: Overconfident estimates in mixed models. Behav. Ecol. 20: 416–420.
- Silk, M. J., X. A. Harrison, and D. J. Hodgson. 2020. Perils and pitfalls of mixed-effects regression models in biology. PeerJ. 8: 1–20.

# 9 R Activity

[We are working with the same data as last week] In this study, two animal species (goats or sheep) were fed one of three diets (control, alfalfa hay, and cottonseed meal) and received a drug injection (slaframine in saline or just saline). The 12 treatments were assigned in a randomized complete block design with twelve blocks (replications). So, each combination of animal × diet × drug combination appears twelve times.

For this activity we are ONLY going to look at the effects of drug (and reps) on glucose blood levels.

1. Load in the `glucose_df.txt` dataset.

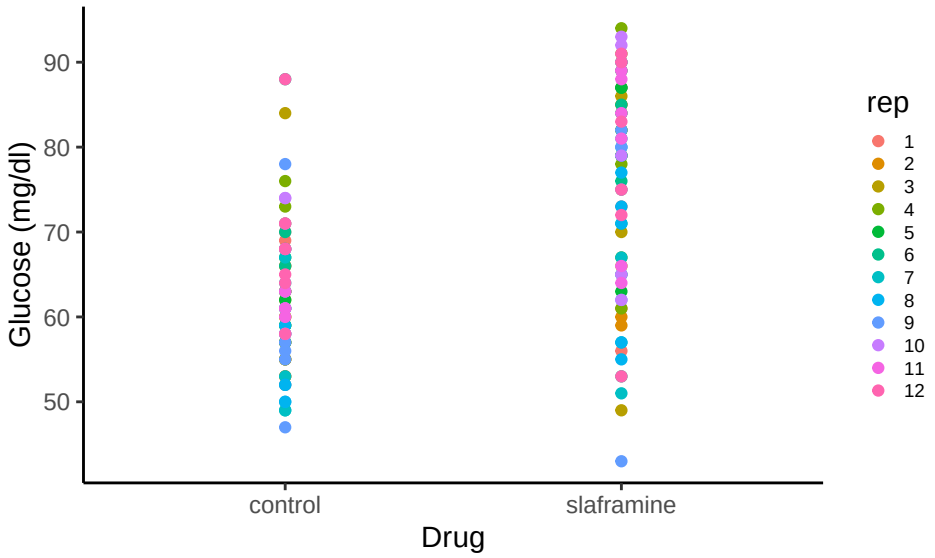
```
gluc_df <- read.table("glucose_df.txt", sep="\t", header=T)
summary(gluc_df)
```

```
##       rep           animal              diet              drug
##  Min.   : 1.00   Length:144         Length:144         Length:144
##  1st Qu.: 3.75   Class :character   Class :character   Class :character
##  Median : 6.50   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 6.50
##  3rd Qu.: 9.25
##  Max.   :12.00
##     glucose
##  Min.   :43.00
##  1st Qu.:59.00
##  Median :66.00
##  Mean   :68.65
##  3rd Qu.:79.00
##  Max.   :94.00
```

```
gluc_df$rep <- as.factor(gluc_df$rep)
```

2. Graph `glucose` as a function of `drug`. Color each point by the variable `rep` and change the axis labels to "`Glucose (mg/dl)`" and "`Drug`".

```
ggplot(gluc_df, mapping=aes( y=glucose, x=drug, color=rep))+
geom_point()+
theme_classic()+
ylab("Glucose (mg/dl)")+
xlab("Drug")+
theme(legend.text=element_text(size=7))+
theme(legend.key.size = unit(0.3, 'cm'))
```




3. Fit a fixed-effects only model of `glucose` as a function of `rep` and `drug`. Provide a `summary()` of the model.

```
fit_lm <- lm(glucose~ rep + drug, data=gluc_df)
summary(fit_lm)
```

```
##
## Call:
## lm(formula = glucose ~ rep + drug, data = gluc_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26.6111  -5.5694  -0.3333   6.9861  21.8611
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     61.9722     3.1358  19.763  < 2e-16 ***
## rep2            -2.3333     4.2608  -0.548   0.5849
## rep3             0.1667     4.2608   0.039   0.9689
## rep4             4.8333     4.2608   1.134   0.2587
## rep5             1.7500     4.2608   0.411   0.6819
## rep6             4.8333     4.2608   1.134   0.2587
## rep7            -4.5833     4.2608  -1.076   0.2840
## rep8            -7.0833     4.2608  -1.662   0.0988 .
## rep9            -4.7500     4.2608  -1.115   0.2670
## rep10            5.0833     4.2608   1.193   0.2350
## rep11            2.9167     4.2608   0.685   0.4948
## rep12            5.0000     4.2608   1.174   0.2427
## drugslaframine  12.3889     1.7394   7.122 6.32e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.44 on 131 degrees of freedom
## Multiple R-squared:  0.359,  Adjusted R-squared:  0.3003
## F-statistic: 6.114 on 12 and 131 DF,  p-value: 1.923e-08
```

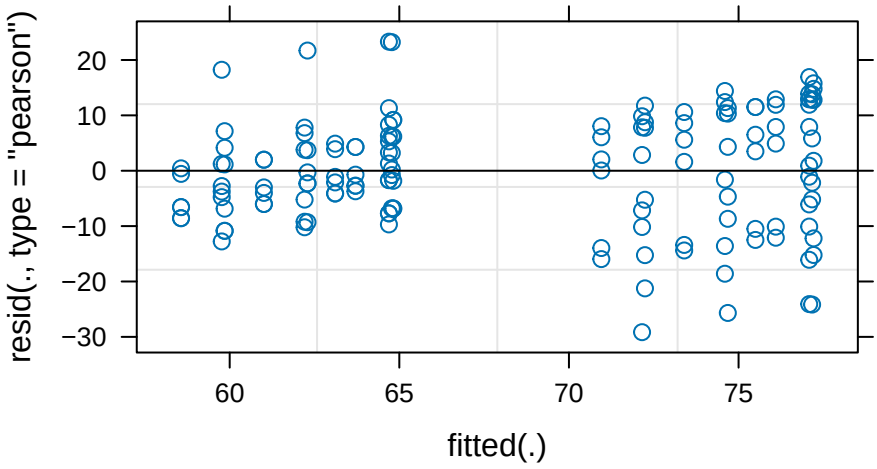
4. Fit a linear mixed-effects model of `glucose` as a function of `drug`. Include a term for `rep` as a random intercept. Provide a `summary()` of the model and check the assumptions (please provide proof you conducted diagnostics and **ensure the summary output has *p*-values**). Are you satisfied the assumptions are met? Why or why not?

```
library(lme4)
library(lmerTest)
```

```
fit_lme_1 <- lmer(glucose~ drug + (1|rep), data=gluc_df)
summary(fit_lme_1)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: glucose ~ drug + (1 | rep)
##    Data: gluc_df
##
## REML criterion at convergence: 1085.5
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.79352 -0.65366  0.01272  0.74784  2.23314
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  rep      (Intercept)   9.606   3.099
##  Residual             108.924  10.437
## Number of obs: 144, groups:  rep, 12
##
## Fixed effects:
##                Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)      62.458      1.521  23.813  41.065  < 2e-16 ***
## drugslaframine   12.389      1.739 131.000   7.122 6.32e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## drugslafrmn -0.572
```

```
plot(fit_lme_1)
```




**ANSWER:** Seems okay - the variance is slightly higher in the slaframine group, but not problematic enough to warrant transformations.

5. Write one sentence comparing the conclusions one would draw from each model and one sentence interpreting the mixed-effects model.

   **ANSWER:** The model estimates are equivalent, and thus the conclusions are generally the same. That is, injection with slaframine was associated with an increase in glucose of 12 mg/dl compared with animals injected with a saline-only solution ($t_{131} > 7.12, p < 0.0001$).