

Mixed-Effect Models in R

Andrew McAdam

2019-03-20

Contents

Note	1
Mixed-Effect Models	1
Difference Between Fixed and Random Effects	1
Number of Levels	2
Meta-Data	3
A Starting Example	3
lme4	7
P-values for fixed effects in mixed-effect models	8
Approach #1	8
Approach #2	9
Approach #3	11
ANCOVA Example	13
Assessing Diagnostics with Mixed-Effect Models	13
Test for Heteroscedasticity	14
Normality of Residuals	18
Assumptions about the random effects	19
Fitting Random Slopes	20

Note

This document is a work in progress. In many places it is not as well annotated as I would like, but it's a start!

Mixed-Effect Models

So far, we have talked about various types of statistical models, but all of the effects that we wanted to include as predictors of some response variable, were called ‘fixed effects’. That is, the influence of this variable is defined to be fixed, or constant. In other cases, we might be interested in controlling for the effect of some predictor, but the effect of this predictor is not considered constant. Instead, it is assumed to come from a distribution of possible effects. We call these ‘random effects’. A mixed effect model is simply a model that contains both fixed and random effects as predictors.

Difference Between Fixed and Random Effects

It can sometimes be difficult to understand the difference between fixed and random effects. As a start, it is worth noting that while fixed effects can be categorical (e.g. effect of treatment A, B or C) or continuous

(e.g. effect of temperature), random effects must always be categorical. Recall that there is a philosophical difference between a fixed effect and a random effect. When we include a categorical variable as a fixed effect we are saying that we care exactly what the effects are of each level in that factor. For example, we care what the effect of the pesticide treatment is on crop yield relative to the control, or the application of nematodes to the fields. That is, we care what those specific parameters are for each level of the factor. For random effect, we don't care what the specific effects of each level of the factor are, we are just interested in controlling for variation in that factor. For example, we might have applied each of these treatments in each of 25 farms accross southern Ontario. We really don't care what the effect of farm #2 is relative to farm #22, we just want to control for the fact that farms are likely to differ in their yields and we want to account for this lack of independence in observations. When we fit 'farm' as a random effect, we are not really estimating the effect of each farm, per se. Instead what we are estimating is the *variance in yield among farms*.

What we are saying by fitting farm as a random effect is that if we were to repeat the experiment, it would make perfect sense to use a totally different group of farms. There was nothing special about the 25 that we used originally, we just needed replication in farms and wanted to control for the fact that we had repeated observations for each farm. Sometimes people say that 'random effects' are used because the levels of the factor are selected 'at random' from all possible levels. This is a decent description and is often true, but just because you didn't select your levels of the factor at random doesn't mean that you cannot fit the factor as a random effect. For example, you might only have access to 10 greenhouses within which you can perform your experiments. So there weren't selected at random. Nevertheless you might not care about the specific greenhouses and want to control for differences in your response variable among the greenhouses, so you might fit *greenhouse* as a random effect.

Number of Levels

Beyond the philosophical difference between fixed and random effects, there is also a practical distinction that you need to pay attention to. In order to get reasonable estimates for a random effect this effect needs to have at least 6 levels (e.g. 6 greenhouses, 6 fields, 6 populations, 6 individuals). The reason for this is that, unlike fixed effects, mixed models estimate a variance for a random effect (i.e. variance among the levels of your random factor). It starts to get very hard to measure a variance among fewer than 6 levels and this can cause big problems (i.e. unstable estimates) for random effects with less than 6 levels. So if you have a factor that ought to be considered a random effect, but which only has 5 levels, then fit it as a fixed effect instead.

There are some additional packages that we will load for Mixed Models.

```
library(lme4)
```

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

We are going to return to the red squirrel cone handling example where we were interested in the effects of cone handling behavior and cone morphology on the time taken to harvest the seeds from a cone. Remember that in this example we had multiple cone handling observations per individual ("squirrel"). We will account for this sub-sampling here.

I am going to load this csv file here.

```
ConeHandling<-read.table("data/ConeHandling.csv", sep="," , header=T)
```

```
summary(ConeHandling)
```

##	Squirrel	Trial	Length	Width	Handed
##	Mireille : 6	Min. :1.000	Min. :35.00	Min. :10.00	L:20
##	Mark : 5	1st Qu.:2.000	1st Qu.:41.95	1st Qu.:12.85	R:19
##	Adi : 4	Median :3.000	Median :45.30	Median :14.30	
##	Christina: 4	Mean :2.769	Mean :45.32	Mean :13.86	
##	Mike : 4	3rd Qu.:4.000	3rd Qu.:49.45	3rd Qu.:14.80	
##	Moe : 4	Max. :6.000	Max. :57.00	Max. :16.30	

```
## (Other) :12
## Chirality      Time      Sex
## D:19      Min.    : 61.0   F:22
## S:20      1st Qu.: 82.5   M:17
##           Median :105.0
##           Mean   :115.3
##           3rd Qu.:146.0
##           Max.   :209.0
##
```

Meta-Data

These data were from an experiment that I ran with field assistants as part of the Kluane Red Squirrel Project. I was interested in whether the handedness of squirrels affected the speed with which they could extract seeds from spruce cones with cone scales arranged in a dextral or sinistral pattern. Instead of measuring cone harvesting by squirrels, I instead asked field assistants to remove seeds one at a time, while spiraling the cone like a left-handed or right-handed squirrel. Each field assistant ('squirrel') was timed handling more than one cone.

- *Squirrel* - name of the field assistant who handled the cone.
- *Trial* - trial number for that particular squirrel (1-6)
- *Length* - length of the cone (mm)
- *Handed* - whether the field assistant handled the cone in a right-handed (R) or left-handed (L) way.
- *Chirality* - whether the cone scales were arranged on the cone in a dextral (D) or sinistral (S) pattern.
- *Time* - time it took the assistant to remove all the seeds from the cone (seconds) **response variable**
- *Sex* - sex of the field assistant (M = male; F = female). This should actually be called gender since I was unaware of the sex of each of the field assistants and instead inferred gender based on the way they presented themselves.

A Starting Example

Let's start with the simplest possible example. Let's just say that we were interested in the average time that it takes to harvest seed from a cone.

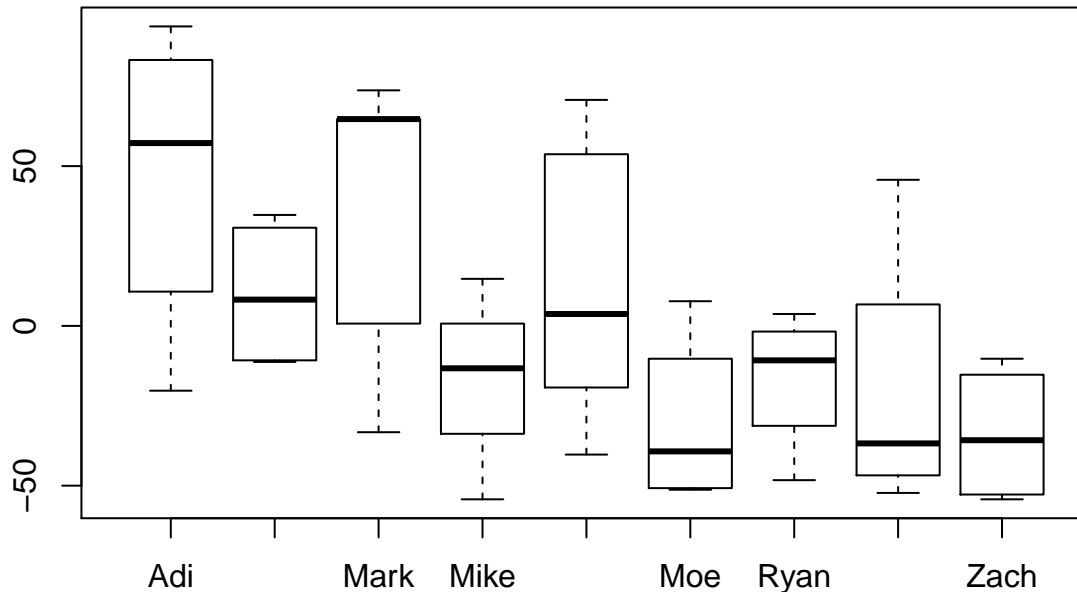
```
cone<-lm(Time~1, data=ConeHandling)
summary (cone)
```

```
##
## Call:
## lm(formula = Time ~ 1, data = ConeHandling)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -54.28  -32.78  -10.28   30.72   93.72
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   115.282      6.823   16.89  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 42.61 on 38 degrees of freedom
```

So this model that contains only the intercept provides us with an estimate of the mean time taken to harvest a cone (along with a standard error and a test of whether this differs from zero). But this inference is based on the fact that these 39 observations are independent, and we know that they are not. We measured each squirrel more than once. We can look at this graphically as.

```
plot(ConeHandling$Squirrel, resid(cone))
```



This means that our inference will be overly liberal since we do not have independent observations (or more importantly errors). Also because there are differences among squirrels our overall mean time is not a mean of the squirrels but a mean of all observations and will be affected by the number of observations that we have for each squirrel.

We can account for these differences among squirrels by including squirrel as a fixed effect in the model.

```
cone2<-lm(Time~Squirrel, data=ConeHandling)
summary (cone2)
```

```
##
## Call:
## lm(formula = Time ~ Squirrel, data = ConeHandling)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -67.40  -21.00    2.25   25.25   65.75
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      162.25      18.47   8.785 8.54e-10 ***
## SquirrelChristina  -37.00      26.12  -1.417  0.16692
## SquirrelMark       -12.85      24.78  -0.519  0.60787
## SquirrelMike       -63.50      26.12  -2.431  0.02124 *
## SquirrelMireille   -34.92      23.84  -1.464  0.15350
## SquirrelMoe       -77.50      26.12  -2.967  0.00585 **
## SquirrelRyan      -63.50      26.12  -2.431  0.02124 *
```

```
## SquirrelSarah      -67.00      26.12  -2.565  0.01556 *
## SquirrelZach       -81.00      26.12  -3.101  0.00417 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36.94 on 30 degrees of freedom
## Multiple R-squared:  0.4067, Adjusted R-squared:  0.2485
## F-statistic: 2.571 on 8 and 30 DF,  p-value: 0.02886
```

But this doesn't really do what we want. We are not so much interested in estimating the mean length of time for each squirrel to eat their cones. Instead we simply want a population mean value, but we want to account for the fact that we have multiple observations per squirrel and that squirrels vary in the length of time it takes them to eat a cone. We can do this by instead including Squirrel identity as a random effect.

To do this we need to load a new package:

```
library(nlme)
```

```
##
## Attaching package: 'nlme'
## The following object is masked from 'package:lme4':
##
##      lmList
```

We can then fit a random effect using lme

```
cone.lme<-lme(Time~1, random=~1|Squirrel, data=ConeHandling)
```

Note the notation for specifying a random effect compared to a fixed effect.

```
summary(cone.lme)
```

```
## Linear mixed-effects model fit by REML
## Data: ConeHandling
##      AIC      BIC    logLik
## 399.4202 404.333 -196.7101
##
## Random effects:
## Formula: ~1 | Squirrel
##      (Intercept) Residual
## StdDev:      22.37429 36.96217
##
## Fixed effects: Time ~ 1
##              Value Std.Error DF t-value p-value
## (Intercept) 114.2372  9.544501 30 11.9689      0
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -1.4875823 -0.8059453 -0.1090589  0.6934115  1.7916246
##
## Number of Observations: 39
## Number of Groups: 9
```

Note that we have a slightly different output here. Note that instead of getting estimates of the effect of each squirrel (fixed effects) we now have an estimate of the amount of variance among squirrels in the amount of time it takes for them to eat a cone (sd=22.4). Note also that we had 39 observations for 9 squirrels. Note also that our estimate of the mean has now been adjusted. This new estimate now takes into account the fact that we have multiple observations per squirrel.

Note also that we can convert our sd's into variances and calculate the proportion of variance explained by the random effect.

```
(22.37429^2)/(22.37429^2+36.96217^2)
```

```
## [1] 0.2681626
```

So about 27% of the variation in cone handling time is accounted for by variation among squirrels. In this case this is the proportion of the total variance explained by the random effect because we have included only the intercept as a fixed effect in the model. Otherwise this would be the proportion of variance not explained by the fixed effects that was explained by the random effect.

You can think of this proportion as the **repeatability** of cone handling time for squirrels.

This is just a simple example that I have used for the purpose of demonstration. We haven't actually tested anything yet. But let's say we wanted to test for an effect of Cone Handling behavior on handling time. We could simply add the effect if Handed to our model. This is analagous to a "Block" design where the effect of Handedness is blocked by Squirrel. Here "Squirrel" is acting like "bench" or "field" or "plot" or "stand".

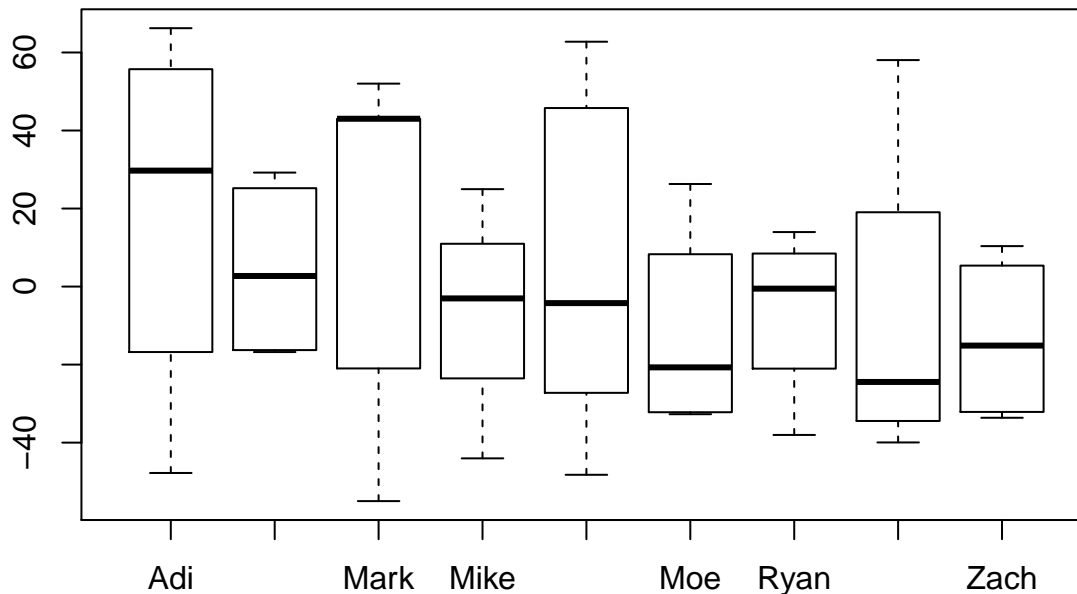
```
cone.lme2<-lme(Time~Handed, random=~1|Squirrel, data=ConeHandling)
summary (cone.lme2)
```

```
## Linear mixed-effects model fit by REML
## Data: ConeHandling
##      AIC      BIC    logLik
## 390.8531 397.2968 -191.4266
##
## Random effects:
## Formula: ~1 | Squirrel
##      (Intercept) Residual
## StdDev:    23.57168 35.09346
##
## Fixed effects: Time ~ Handed
##              Value Std.Error DF  t-value p-value
## (Intercept) 103.08871  11.13432 29  9.258645  0.0000
## HandedR      22.67845  11.25996 29  2.014079  0.0534
## Correlation:
##      (Intr)
## HandedR -0.494
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -1.4369393 -0.6659539 -0.2208025  0.6377257  1.7128386
##
## Number of Observations: 39
## Number of Groups: 9
```

Note that we have an estimate of the effect of Handedness on cone handling time that has taken into account the fact that we have multiple observations per squirrel

Note that now the residuals are independent of squirrels.

```
plot(ConeHandling$Squirrel, resid(cone.lme))
```



lme4

We can now repeat this analysis using a different package called lme4. This package was written by the same person that wrote the nlme package, but it is more recent and can do many things that nlme cannot.

```
library(lme4)
```

This might take a while. If R tells you that a package is missing then download/load that and try and load lme4 again

or

```
require(lme4)
```

The notation for lme4 is slightly different from nlme.

```
cone.lme3<-lmer(Time~Handed+(1|Squirrel), data=ConeHandling)
summary(cone.lme3)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Time ~ Handed + (1 | Squirrel)
## Data: ConeHandling
##
## REML criterion at convergence: 382.9
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.4369 -0.6660 -0.2208  0.6377  1.7128
##
## Random effects:
## Groups   Name                Variance Std.Dev.
## Squirrel (Intercept)  555.6      23.57
## Residual                1231.6     35.09
## Number of obs: 39, groups: Squirrel, 9
```

```
##
## Fixed effects:
##           Estimate Std. Error t value
## (Intercept)  103.09      11.13   9.259
## HandedR      22.68      11.26   2.014
##
## Correlation of Fixed Effects:
##      (Intr)
## HandedR -0.494
```

So there are some important things to point out about this output. First of all, note that we still have almost all of the same information. Also note that the Std.Dev. is not a measure of the uncertainty of the random effect variance estimate but is instead simply the sqrt of the variance estimate. I guess so people prefer one while others prefer the other so they have output both.

```
sqrt(555.6)
```

```
## [1] 23.57117
```

P-values for fixed effects in mixed-effect models

OK so many of you have probably noticed that for the fixed effects we don't have a df for the T test statistic and we don't have a p-value. Yikes, no P-value! What are we to do with no P-value? **Now is when I warn you that I am going to tell you something that will disturb many of you for a couple of reasons.** These important bits of information are not included because there is currently no consensus on a general way for these df to be calculated. Without a specified df we cannot calculate a p-value. In some special cases, the appropriate calculations are known, but there is not a general solution here. So rather than provide estimates that are just that - estimates, the authors of the package decided not to report these values. Note also that you will get df and p-values if you run the model with the older nlme and also with proc MIXED in SAS. However, just because these functions provide this information doesn't mean that it is right! I will not go into the details of why, but i will simply say that this is something that the hard core statisticians are currently working on right now. Expect a solution to this at some point. For more info on this see the "[R] lmer, p-values and all that.pdf" file or visit <https://stat.ethz.ch/pipermail/r-help/2006-May/094765.html>. You could also see <http://www.maths.bath.ac.uk/~jjf23/ELM/mixchange.pdf> for updates to the Faraway "Extending the Linear Model" book.

There are several approaches for dealing with this situation:

Approach #1

If we are comfortable making our own decision about what the appropriate degrees of freedom are then we could calculate an appropriate p-value...

```
2*pt(2.014, df=29, lower.tail=FALSE)
```

```
## [1] 0.05336944
```

Note that the 2 up front is because a t-test of a parameter in a model is always two-tailed.

Note that this result is the same as what was provided by nlme. The 29 df are based on n=39 minus one df for the intercept and one for the parameter for handedness and one for each of the 8 additional levels of the random effect. This is based on the assumption that the random effect consumes 8 df.

IMPORTANT If I were to do a poll in class of how many people would be comfortable reporting the p-value from the nlme cone.lme2 model and how many people would be comfortable reporting the p-value above from the lmer cone.lme3 model and `2*pt(2.014, df=29, lower.tail=FALSE)`, I would bet that almost all would

report the former and almost none would report the latter. This is because you inherently trust a canned package and inherently distrust your own ability to do statistics. After checking to see that the p-values are exactly the same, take some time to reconsider both your trust in canned packages (SAS, SPSS, R, etc) and in yourself!

The above example with 29 df is based on a simplifying assumption that the predicted random effects are the true values, which is why each additional level (beyond the intercept) consumes a single df. The important issue here is that the random effects are not known but are predicted from the model. It is not entirely straight-forward to sort out how many df should be used for a random effect when designs are not balanced. It could range anywhere from 1 to K-1, where K is the number of levels in the random effect. If you wanted to assess the range of possible p-values you could use an error df based on 1 df for the random effects and a separate analysis with K-1 df for the random effects and use these results to get a sense of the range of possible p-values (a brute force approach).

For example, in this case what if we used 39 observations - 1df for the intercept - 1 df for *handedness* - 1 df for the random effect = 36 df. This assumes that the random effect uses only one df.

```
2*pt(2.014, df=36, lower.tail=FALSE)
```

```
## [1] 0.05152923
```

Notice that the consequences for the p-value are not huge, so maybe we shouldn't worry so much about this.

This approach still assumes that the F (or t) distribution is the appropriate distribution to test the fixed effect parameters against and this is not always clear. Basically the statisticians need to get busy and sort all this mess out for us ecologists!

Approach #2

We can assess the significance of the fixed effects by using our good friend the likelihood ratio test! We can test the significance of Handedness by comparing the model with Handedness to the simpler model with just the intercept.

```
cone.lme3<-lmer(Time~Handed+(1|Squirrel), data=ConeHandling)
cone.lme4<-lmer(Time~1+(1|Squirrel), data=ConeHandling)
anova (cone.lme4, cone.lme3, test="Chi")
```

```
## refitting model(s) with ML (instead of REML)
## Data: ConeHandling
## Models:
## cone.lme4: Time ~ 1 + (1 | Squirrel)
## cone.lme3: Time ~ Handed + (1 | Squirrel)
##           Df    AIC    BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## cone.lme4  3 405.71 410.70 -199.85   399.71
## cone.lme3  4 403.83 410.49 -197.92   395.83 3.8747     1 0.04902 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note the comments above the output regarding the fact that the models were refitted using ML rather than REML. When models differ in fixed effects we need to use ML rather than REML as our measure of the fit of the model to the data.

Note that the simpler model is listed before the more complex one and that the null hypothesis is that the simpler model is better. In this case the result is significant which indicates that the more complicated model is significantly better than the simpler one.

We normally use restricted maximum likelihood here (REML) which is “restricted” to consider the likelihood after taking into account the fixed effects in the model. When you change the fixed effects in the model the

likelihood based on REML are no longer directly comparable. But we can use good old ML instead. Bottom line... when you are using likelihoods to assess the fit of models that differ in their fixed effect structure then you need to use maximum likelihood (REML=F) rather than Restricted Maximum Likelihood (REML=T; the default in R). In other cases REML does a better job of estimating variances and likelihoods so it is preferred when you are NOT comparing models or when the models differ only in their random effects.

```
cone.lme3<-lmer(Time~Handed+(1|Squirrel), data=ConeHandling, REML=F)
cone.lme4<-lmer(Time~1+(1|Squirrel), data=ConeHandling, REML=F)
anova (cone.lme4, cone.lme3, test="Chi")

## Data: ConeHandling
## Models:
## cone.lme4: Time ~ 1 + (1 | Squirrel)
## cone.lme3: Time ~ Handed + (1 | Squirrel)
##           Df      AIC      BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## cone.lme4  3 405.71 410.70 -199.85   399.71
## cone.lme3  4 403.83 410.49 -197.92   395.83 3.8747      1   0.04902 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that this is the same as above.

So the likelihood ratio test suggests that the effect of handedness is in fact significant.

We can similarly assess the significance of the random effect also using a likelihood ratio test.

```
cone.lme3<-lmer(Time~Handed+(1|Squirrel), data=ConeHandling)
cone3<-lm(Time~Handed, data=ConeHandling)
```

Since there was no random effect in the second model we needed to use lm rather than lmer. Unfortunately these models are of a different “class” so we can’t use the anova function for our LRT.

```
#anova (cone3, cone.lme3, test="Chi")

# Gives an error
```

So we will need to perform the LRT ourselves

```
as.numeric(2*(logLik(cone.lme3)-logLik(cone3, REML=T)))

## [1] 4.257735
```

Note that we needed to specify REML=T for the lm because the default for this model is not REML it is ML and our lmer model was fit with REML. We need these to be the same to have a valid comparison of the likelihoods.

```
as.numeric(2*(logLik(cone.lme3)-logLik(cone3, REML=F)))

## [1] 16.86828
```

Notice the difference!!!!

We can test this ourselves against a Chi-sq distribution

```
pchisq(4.257733, 1, lower=F)

## [1] 0.03907203
```

So the random effect is significant. That is the model with the random effect is significantly better fit to the data than the model without the random effect. Note that this LRT was tested against 1 df because we are only estimating the one parameter with the random effect (the among squirrel variance in handling time) - we are not fitting a separate parameter for each squirrel!

The final twist that I will likely not go into in class is that using a Chisq distribution for a LRT is only an approximation. We can get a more accurate estimate of the p-value for the test by using a parametric bootstrap approach. The Faraway “Extending the Linear Model” book also provides a good description of this. I will simply provide the code below. Please refer to the Faraway book for more info.

McAdam check this code... Why all the singular fit errors?

```
l1rstat<-numeric(1000)

for (i in 1:1000){
  rTime<-unlist(simulate(cone3))
  cone3r<-lm(rTime~Handed, data=ConeHandling)
  cone.lme3r<-lmer(rTime~Handed+(1|Squirrel), data=ConeHandling)
  l1rstat[i]<-2*(logLik(cone.lme3r)-logLik(cone3r, REML=T))
}
```

This vector “l1rstat” is a vector of 1000 bootstrapped LRT test statistics. We can therefore determine what percentage of these are as extreme or more extreme than the 4.26 value that we calculated above.

```
mean(l1rstat>4.257733)
```

```
## [1] 0.016
```

Note that this is a bootstrap and is based on only 1000 iterations so the number will not be the same each time.

This is similar to what we calculated before based on Chisq.

Now that we have gone to great lengths to use this likelihood ratio approach, I will warn you that likelihood ratio tests are not recommended for testing the fixed effects in mixed effect models because they are unreliable when sample sizes are moderate to low (see Bolker et al. 2009). Bolker et al. 2009 recommend only using LRTs when the ratio of total sample size to # fixed effects are very large and the number of levels of the random effect are very large.

Approach #3

I wanted to present you with some alternatives, but there is a package that has been written that will allow you to get df and p-values from fixed effects in mixed effect models. The package is called *lmerTest*.

```
library(lmerTest)
```

```
##
```

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

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

```
##
```

```
## lmer
```

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

```
##
```

```
## step
```

```
citation("lmerTest")
```

```
##
```

```
## To cite lmerTest in publications use:
```

```
##
```

```
## Kuznetsova A, Brockhoff PB, Christensen RHB (2017). "lmerTest
```

```
## Package: Tests in Linear Mixed Effects Models." _Journal of
```

```
## Statistical Software_, *82*(13), 1-26. doi: 10.18637/jss.v082.i13
## (URL: http://doi.org/10.18637/jss.v082.i13).
##
## A BibTeX entry for LaTeX users is
##
## @Article{,
##   title = {{lmerTest} Package: Tests in Linear Mixed Effects Models},
##   author = {Alexandra Kuznetsova and Per B. Brockhoff and Rune H. B. Christensen},
##   journal = {Journal of Statistical Software},
##   year = {2017},
##   volume = {82},
##   number = {13},
##   pages = {1--26},
##   doi = {10.18637/jss.v082.i13},
## }
```

Once we have loaded the package into our R session, the package will provide p-values for the fixed effects when we use the summary command. Nothing else needed!

```
cone.lme3<-lmer(Time~Handed+(1|Squirrel), data=ConeHandling)
summary (cone.lme3)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: Time ~ Handed + (1 | Squirrel)
## Data: ConeHandling
##
## REML criterion at convergence: 382.9
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.4369 -0.6660 -0.2208  0.6377  1.7128
##
## Random effects:
## Groups   Name                Variance Std.Dev.
## Squirrel (Intercept)  555.6      23.57
## Residual                1231.6     35.09
## Number of obs: 39, groups: Squirrel, 9
##
## Fixed effects:
##              Estimate Std. Error    df t value Pr(>|t|)
## (Intercept)   103.09      11.13  13.45   9.259 3.33e-07 ***
## HandedR        22.68      11.26  29.04   2.014  0.0533 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr)
## HandedR -0.494
```

Note that I didn't need to specify anything for lmerTest. Just once it is loaded then a summary of an lmer model will automatically estimate p-values using a df based on what is called a Satterthwaite approximation. Note that the error df for the fixed effect is estimated to be 29.04 here. This is similar to what we calculated before. I don't know exactly how the Satterthwaite approximation estimates these df, but this is the technique that is used by SAS in proc Mixed. REMEMBER though that this is just ONE solution and there is no consensus among statisticians that this is an appropriate answer!!! Just because SAS does it doesn't mean

that it is necessarily correct!!!

For now, however, this appears to be the widely accepted approach for estimating df and p-values for fixed effects in a mixed-effect model.

ANCOVA Example

I now want to leave this simple example and move to the ANCOVA example that we talked about in class.

```
cone.lme5<-lme(Time~Width, random=~1|Squirrel, data=ConeHandling)
cone.lme6<-lmer(Time~Width+(1|Squirrel), data=ConeHandling)
summary (cone.lme6)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: Time ~ Width + (1 | Squirrel)
## Data: ConeHandling
##
## REML criterion at convergence: 372.3
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.88633 -0.34753 -0.04923  0.39848  2.03466
##
## Random effects:
## Groups Name Variance Std.Dev.
## Squirrel (Intercept) 442.3 21.03
## Residual 866.7 29.44
## Number of obs: 39, groups: Squirrel, 9
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept) -133.513    55.395   35.230  -2.410  0.0213 *
## Width       17.860     3.948   35.819   4.524 6.44e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr)
## Width -0.988
```

We could assess the significance of the random effect using the procedures described above.

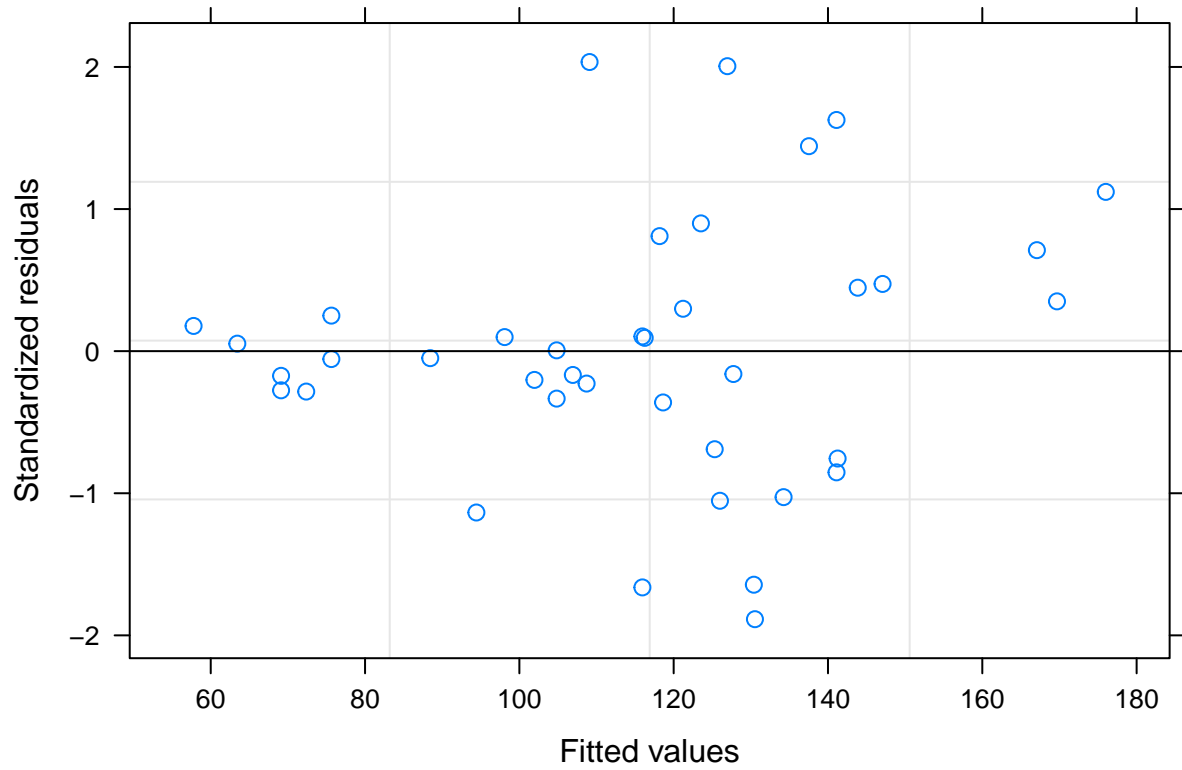
Assessing Diagnostics with Mixed-Effect Models

We will work with our most recent model and assess the diagnostics for this model using both lmer and nlme

```
cone.lme6<-lmer(Time~Width+(1|Squirrel), data=ConeHandling)
cone.lme5<-lme(Time~Width, random=~1|Squirrel, data=ConeHandling)
```

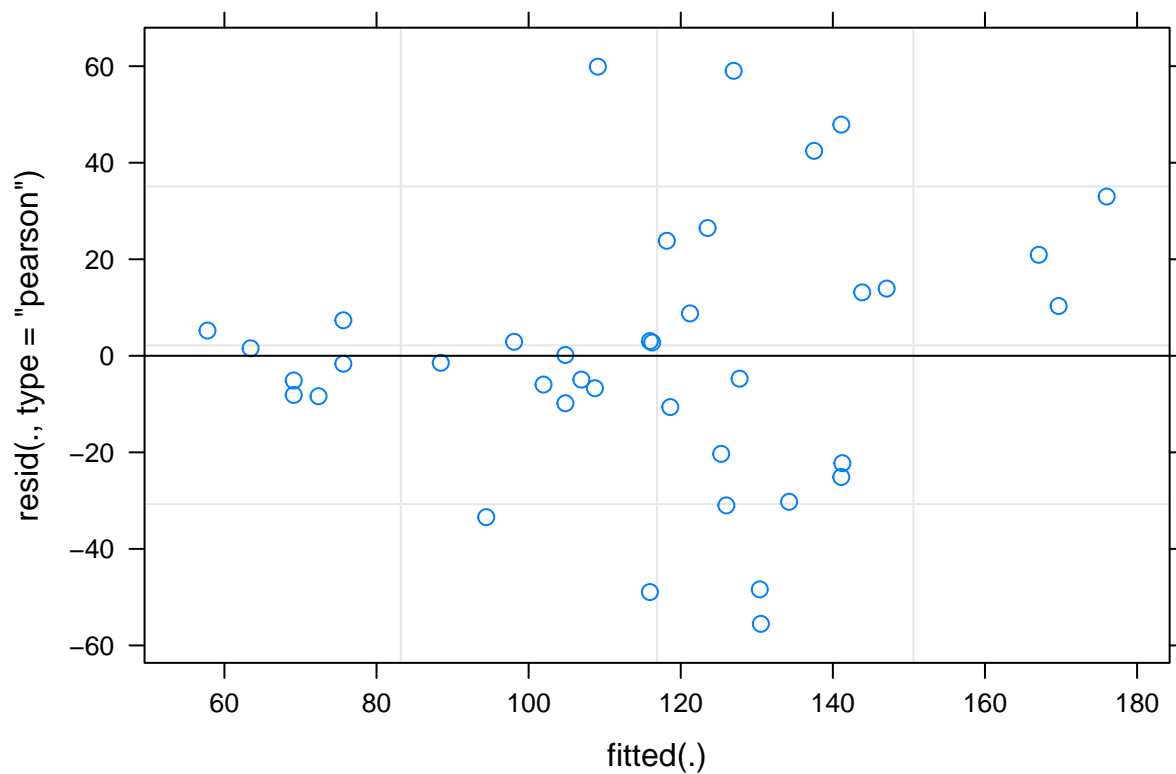
Test for Heteroscedasticity

```
plot(cone.lme5)
```



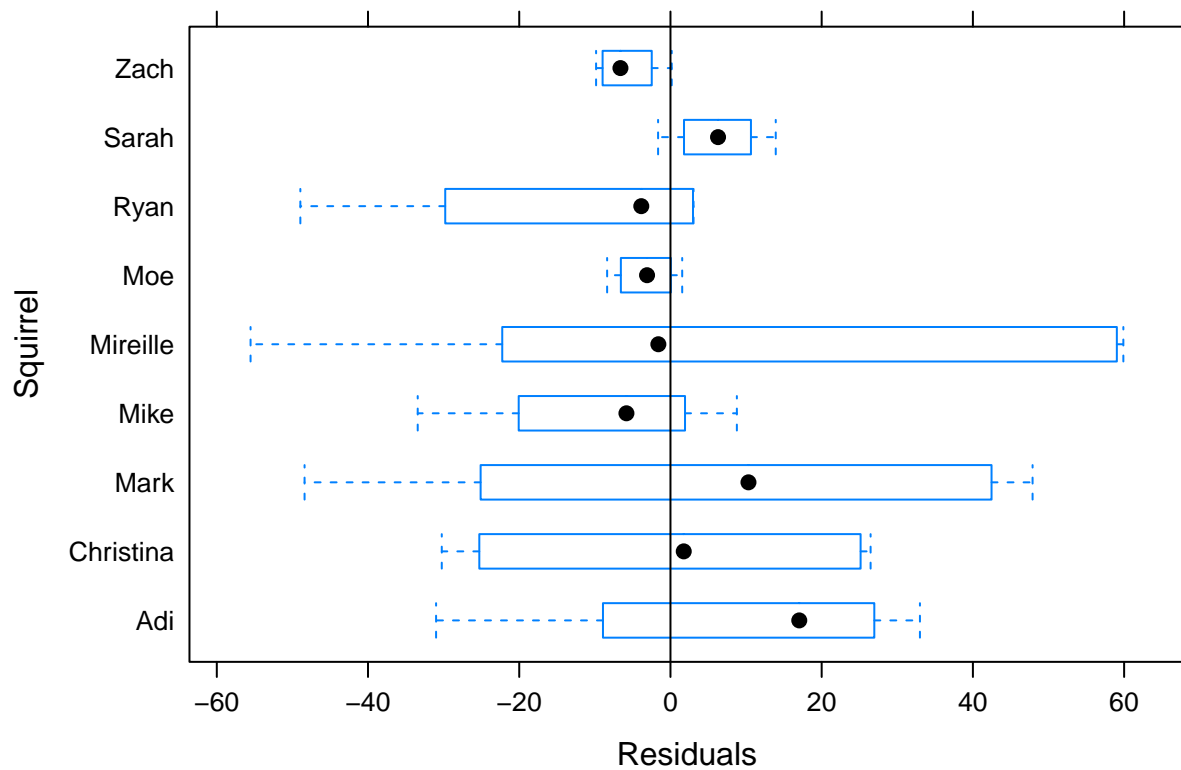
or

```
plot(cone.lme6)
```



Are the within squirrel residuals centered on zero and have a constant variance?

```
plot(cone.lme5, Squirrel~resid(.), abline=0)
```

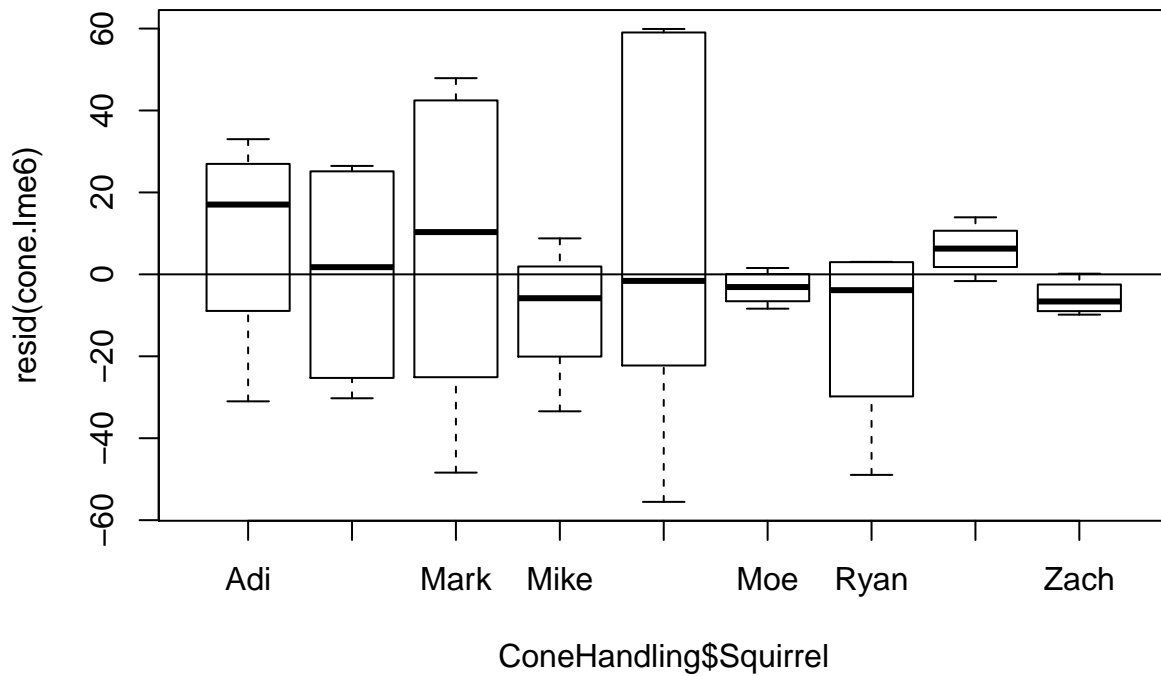


The boxes are somewhat centered around zero. The variances look like they differ among groups but this is hard to tell with so few observations in each group. You should look to see if there is some consistent pattern

in either the average residual or the size of the box between some known groups in the data. For example, do males consistently have higher variance in residuals than females?

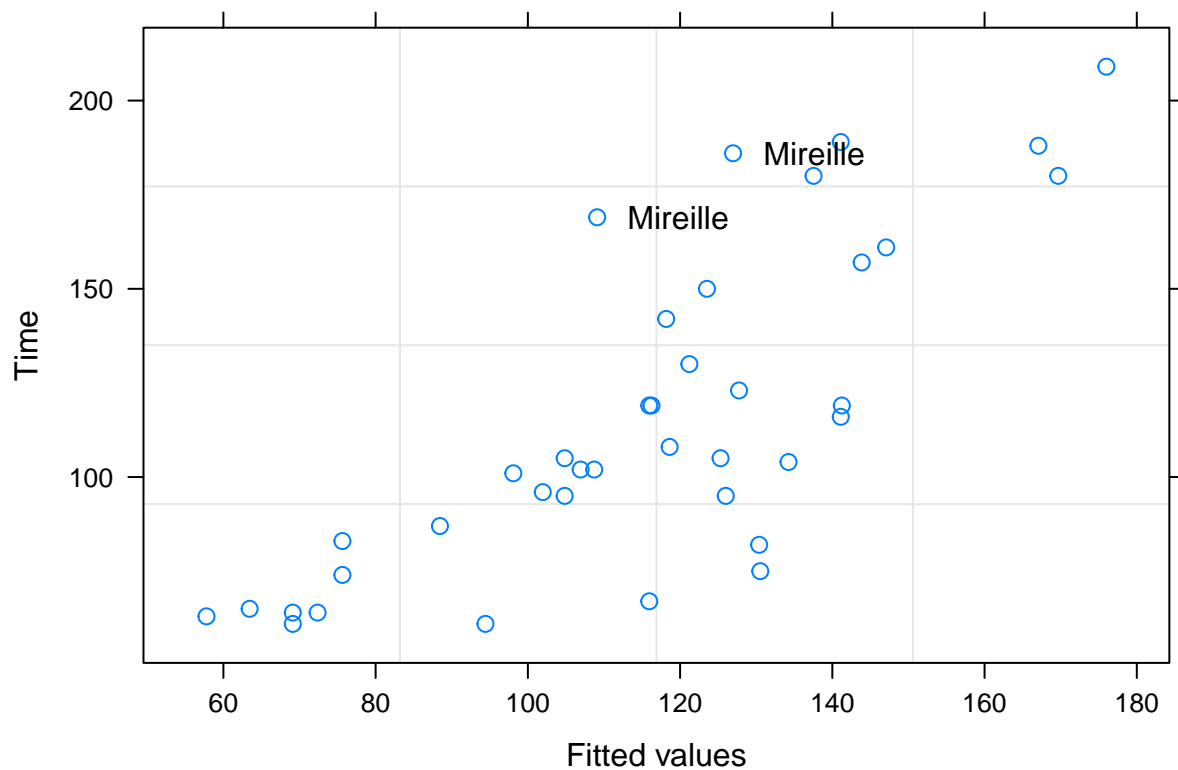
OR

```
plot(resid(cone.lme6)~ConeHandling$Squirrel)
abline(h=0)
```



The overall fit of the model can be assessed using

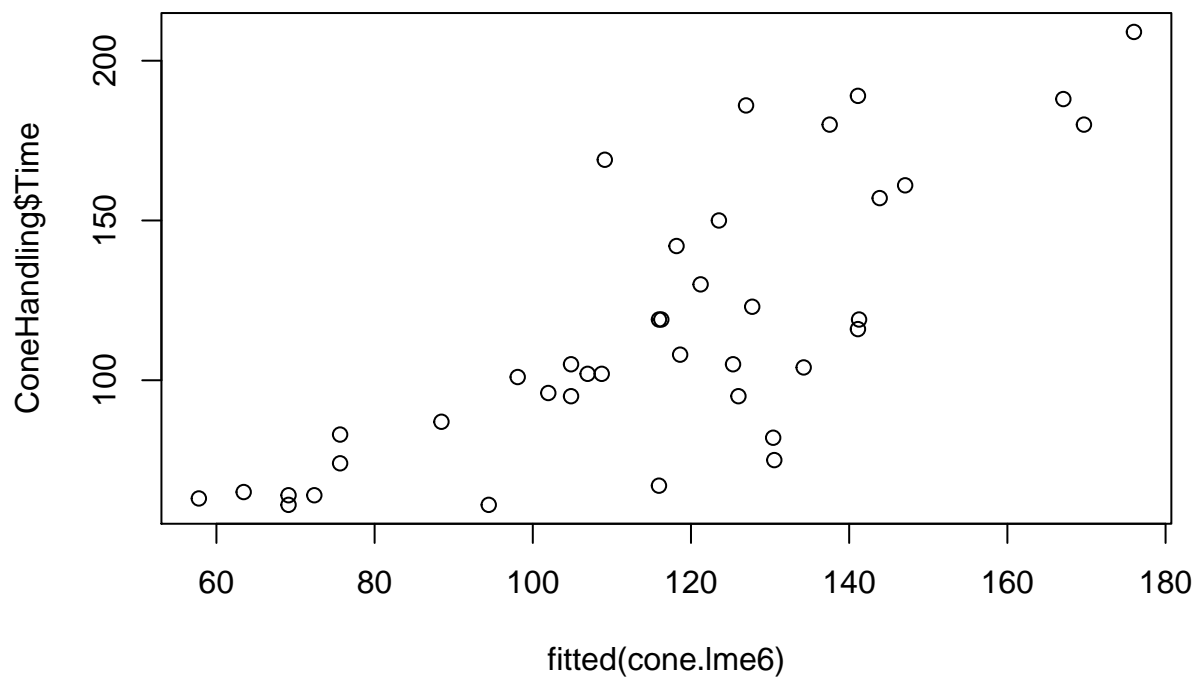
```
plot(cone.lme5, Time~fitted(.), id=0.05, adj=-0.3)
```

This plots the model fitted values against observed values. Any point with an unusual value ($id=0.05$) is labeled and the label is offset from the data point ($adj=-0.3$). id must be a number between zero and 1. Points with standardized residuals greater than $(1-id)/2$ in absolute value will be identified.

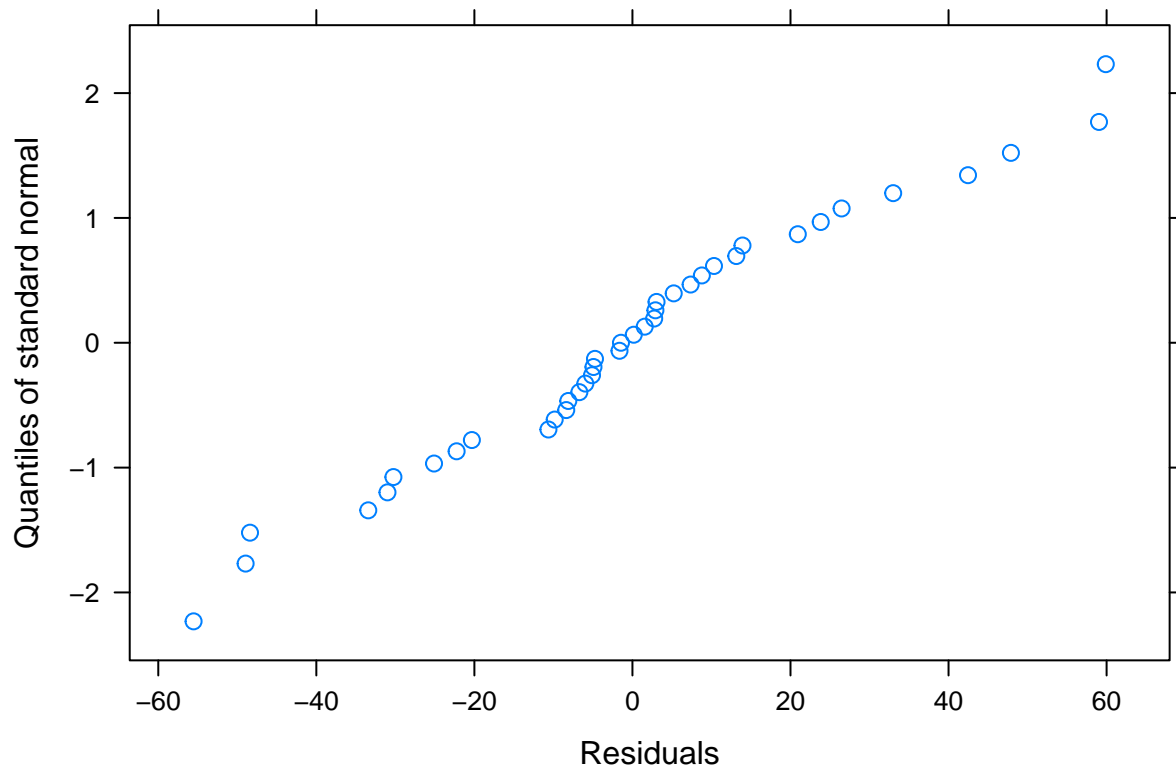
This can also be done (somewhat) using the more general

```
plot(ConeHandling$Time~fitted(cone.lme6))
```



Normality of Residuals

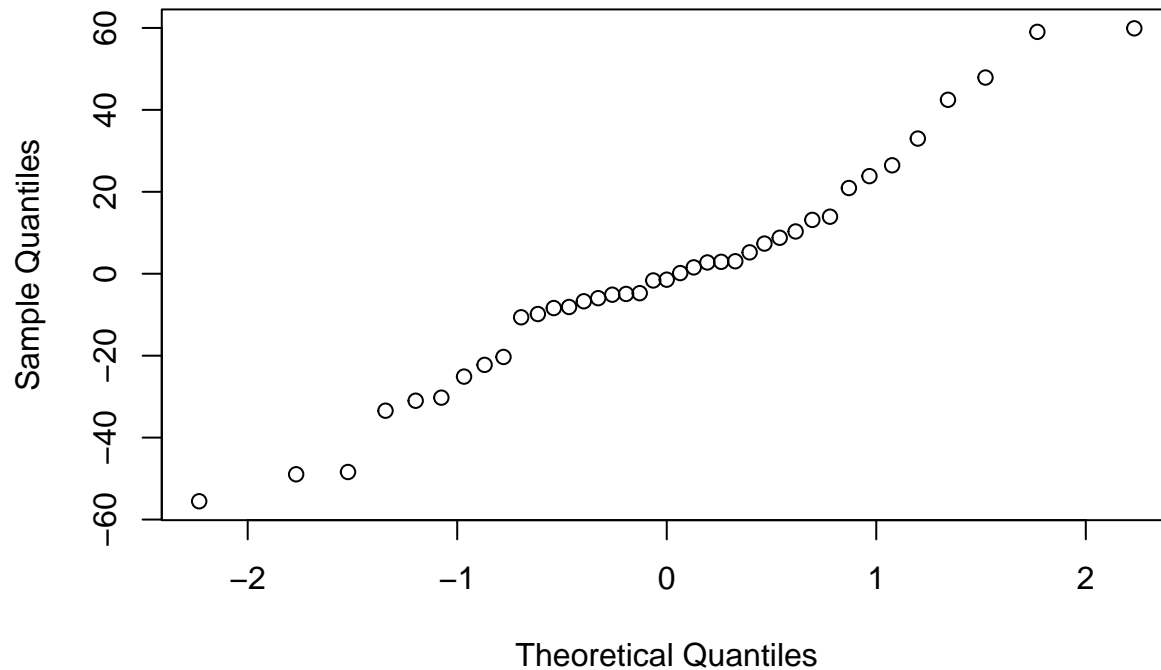
```
qqnorm(cone.lme5, ~resid(.), id=0.05, adj=-0.75)
```



Note that the residuals and the expected values are reversed on the axes in that plot. We can also use...

```
qqnorm(resid(cone.lme6))
```

Normal Q-Q Plot



Note that the distribution of errors tends to have long tails.

Assumptions about the random effects

We can predict the random effects from the model using

```
ranef(cone.lme6)$Squirrel
```

```
##          (Intercept)
## Adi          18.4064126
## Christina    -0.1343578
## Mark          13.8703596
## Mike         -18.5202329
## Mireille     19.3697748
## Moe          -6.6397481
## Ryan         -27.3607333
## Sarah        12.6945378
## Zach        -11.6860128
```

Note that there is also a “ranef” function in the nlme package but when the lme4 package loaded it over-wrote it and so the ranef function for the lme4 package only runs with an lmer model.

```
nlme::ranef(cone.lme5)
```

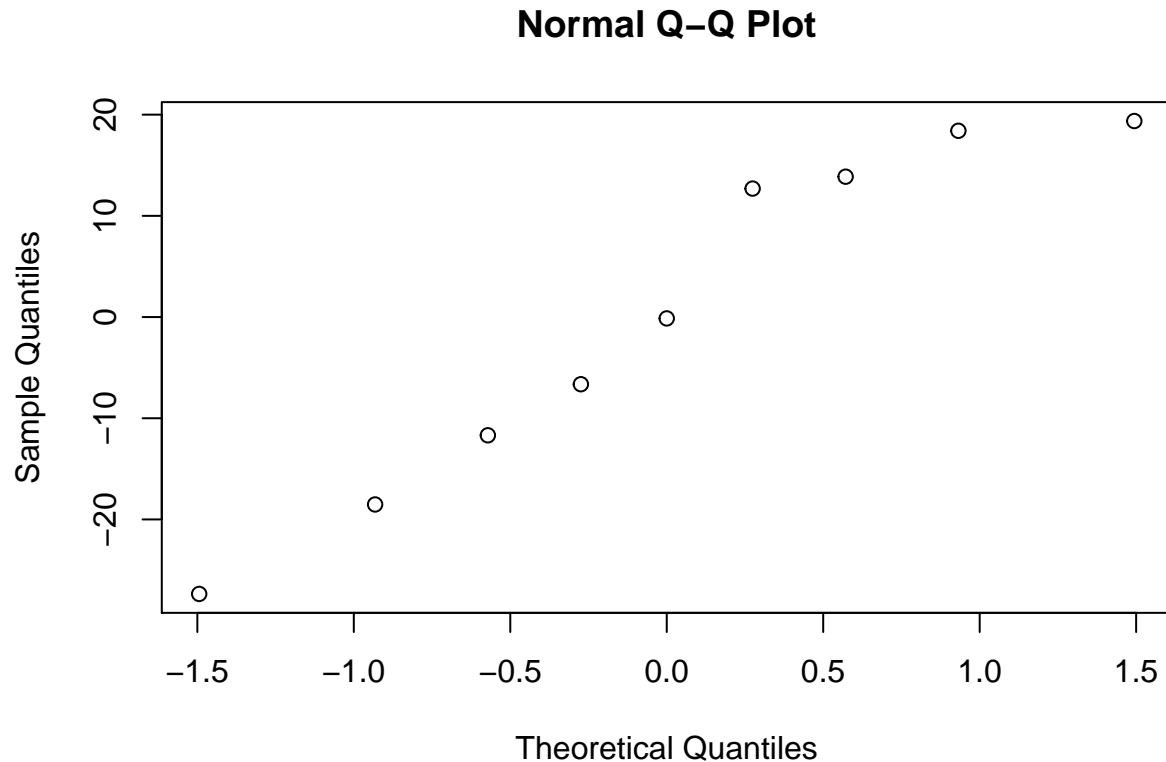
```
##          (Intercept)
## Adi          18.4066678
## Christina    -0.1343679
## Mark          13.8705227
## Mike         -18.5205236
## Mireille     19.3700316
## Moe          -6.6398041
```

```
## Ryan      -27.3611807
## Sarah     12.6948065
## Zach      -11.6861523
```

So we needed to use the “::” to specify that the ‘ranef’ function should be found in the nlme package

Remember that these random effects are not estimated by the model. The model is used to PREDICT these values based on the ESTIMATED among squirrel variance.

```
qqnorm (ranef(cone.lme6)$Squirrel[[1]])
```



So the distribution of these random effects looks good.

So the diagnostics of this model look pretty good.

Fitting Random Slopes

In this model we have fit a random effect for variation among squirrels in the average time it took them to harvest a cone. However, we can also assess whether or not there is variation in their response to cones of various widths. That is, does the slope of the relationship vary among squirrels.

```
cone.lme7<-lme(Time~Width, random=~Width|Squirrel, data=ConeHandling)
summary (cone.lme7)
```

```
## Linear mixed-effects model fit by REML
## Data: ConeHandling
##      AIC      BIC    logLik
## 383.7423 393.4078 -185.8711
##
## Random effects:
## Formula: ~Width | Squirrel
```

```
## Structure: General positive-definite, Log-Cholesky parametrization
##           StdDev   Corr
## (Intercept) 30.466584 (Intr)
## Width       3.411942 -0.966
## Residual    29.368650
##
## Fixed effects: Time ~ Width
##           Value Std.Error DF   t-value p-value
## (Intercept) -117.34129  51.29581 29 -2.287542  0.0296
## Width       16.73724   3.80781 29  4.395504  0.0001
## Correlation:
##      (Intr)
## Width -0.988
##
## Standardized Within-Group Residuals:
##      Min          Q1          Med          Q3          Max
## -1.81416325 -0.45127356 -0.05566821  0.37756442  2.16911378
##
## Number of Observations: 39
## Number of Groups: 9
```

Also

```
cone.lme8<-lmer(Time~Width+(Width|Squirrel), data=ConeHandling)
summary (cone.lme8)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: Time ~ Width + (Width | Squirrel)
##      Data: ConeHandling
##
## REML criterion at convergence: 371.6
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.6915 -0.4081 -0.1100  0.3336  2.2432
##
## Random effects:
##      Groups      Name      Variance Std.Dev.  Corr
## Squirrel (Intercept) 20213.2  142.17
##           Width       108.2   10.40  -0.99
## Residual           820.3   28.64
## Number of obs: 39, groups: Squirrel, 9
##
## Fixed effects:
##           Estimate Std. Error      df t value Pr(>|t|)
## (Intercept) -113.198    74.848    1.780  -1.512  0.2841
## Width       16.368     5.368    1.998   3.049  0.0929 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr)
## Width -0.995
```

Note that there is a very strong negative correlation between the random intercept and the random slope.

This can be caused by data that are not centered. We can try and standardize the data first to see if this reduces this correlation but a correlation between the slope and intercept can also arise because there are few data points per individual.

We can ask whether this model that fits a random slope for each individual improves the fit of the model to the data by comparing the fit of the two models.

```
anova (cone.lme6, cone.lme8)
```

```
## refitting model(s) with ML (instead of REML)
## Data: ConeHandling
## Models:
## cone.lme6: Time ~ Width + (1 | Squirrel)
## cone.lme8: Time ~ Width + (Width | Squirrel)
##           Df    AIC    BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## cone.lme6  4 390.84 397.50 -191.42   382.84
## cone.lme8  6 394.07 404.05 -191.03   382.07 0.7746    2    0.6789
```

So the model with the random slope for each squirrel does not fit the data better than the model that assumed a constant slope. So we prefer the simpler model where a common slope is assumed (cone.lme6).