

Mixed effects model

Etienne Low-Décarie

2018-01-07

Contents

Mixed effects models practical	1
Example 1: The difference between fitting a fixed and a random factor	1
Example 2: Coping with pseudoreplication	3
Example 3: Longitudinal data	12
Example 4: Multiple nested terms	17
Example 5:	18

Mixed effects models practical

This exercise is an introduction to Mixed Effects Modelling (initialially written by Diana Bowler (now at Bangor)), with adaptations from various sources, mainly datasets within R). There are a number of questions here for you to work through, enough to occupy two sessions if you need it.

Example 1: The difference between fitting a fixed and a random factor

Data: 7 fish were tested five times for the time it took them to locate a new food resource added to their tank. Do fish vary in the time taken to locate food?

Import the datasheet into R

```
data1<-read.csv("./Data/data1.csv")
names(data1)
```

```
## [1] "Time" "fishId"
```

Remember, once you attach() a data frame, you can call on variables without specifying the name of the data frame.

It is useful to look at summary(data.frame) and str(data.frame) to see simple information on your data sheet
fishId is numerical variable so we need to tell R it is a factor or it assumes it to be a continuous variable

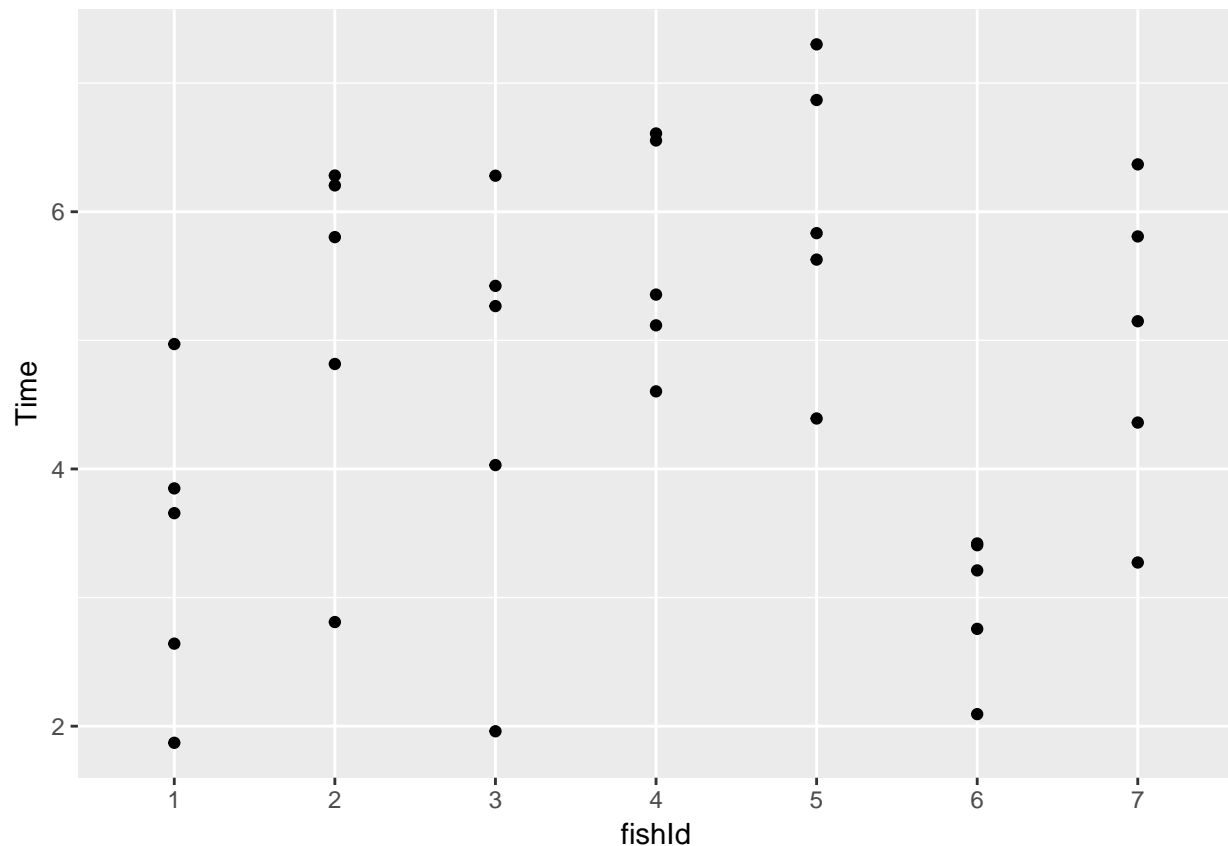
```
data1$fishId<-factor(data1$fishId)
```

View the data as a box plot

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
p <- qplot(data=data1,
           x=fishId,
           y=Time)
print(p)
```



To illustrate the differences between an lm and lme, firstly carry out a straight forward ANOVA with fishId as a fixed term

```
lm1<-lm(Time~fishId,
         data=data1)
summary(lm1)
```

```
##
## Call:
## lm(formula = Time ~ fishId, data = data1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.6317 -0.5965  0.2332  0.8477  1.6881
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.3977     0.5405   6.286 8.5e-07 ***
## fishId2       1.7853     0.7643   2.336 0.02690 *
## fishId3       1.1945     0.7643   1.563 0.12934
## fishId4       2.2500     0.7643   2.944 0.00645 **
## fishId5       2.6071     0.7643   3.411 0.00199 **
## fishId6      -0.4200     0.7643  -0.549 0.58706
```

```
## fishId7      1.5939      0.7643      2.085  0.04627 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.209 on 28 degrees of freedom
## Multiple R-squared:  0.4814, Adjusted R-squared:  0.3702
## F-statistic: 4.331 on 6 and 28 DF,  p-value: 0.003268
```

Now run a lme model with fishId as a random term and the overall mean (denoted by 1) as the only fixed term – you will need to load the nlme library

```
library(nlme)
lme1<-lme(Time~1,random=~1|fishId,
          data=data1)
summary(lme1)
```

```
## Linear mixed-effects model fit by REML
## Data: data1
##      AIC      BIC    logLik
## 127.7182 132.2972 -60.85908
##
## Random effects:
## Formula: ~1 | fishId
##      (Intercept) Residual
## StdDev:    0.9864484 1.208538
##
## Fixed effects: Time ~ 1
##              Value Std.Error DF   t-value p-value
## (Intercept) 4.684939 0.4251375 28 11.01982      0
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -2.1952906 -0.4960046  0.0289981  0.8339982  1.3790577
##
## Number of Observations: 35
## Number of Groups: 7
```

Compare the summary outputs which show you the parameters being estimated by each model:
 * The lm shows parameters for the differences among the fish used in the experiment. The default parameter estimates in R are called “treatment contrasts”. The first parameter (labelled as intercept) is the mean response for the first factor level (fishId 1) and the remaining parameters are the differences between each factor level and the first level.

* In the lme, the effects of individual fish are modelled as random deviations from the population mean; these random effects are assumed to be normally distributed with mean zero and variance as predicted by the model (reported as standard deviation = 0.986 in the model summary).

What are the key differences between the two model outputs? Whether the fixed or random effects’ model is more appropriate would depend on whether you wished to make inferences about differences in the mean response among the factor levels in the experiment or inferences about the between-factor level variability in the population from which the factor levels were drawn.

Example 2: Coping with pseudoreplication

Data: an experiment tested the effect of nest temperature on the weight gain of blue tits in 12 nest boxes. Each nest box was in one of three treatments: control, heated or cooled and chicks were weighed once. There

were 4 boxes in each treatment and 6 individuals (3 of each sex) in each box.

```
data2<-read.csv("./Data/data2.csv")
names(data2)
```

```
## [1] "BOX"      "SEX"      "TREAT"    "WEIGHT"
```

```
str(data2)
```

```
## 'data.frame':    72 obs. of  4 variables:
## $ BOX      : int  1 1 1 1 1 1 2 2 2 2 ...
## $ SEX      : Factor w/ 2 levels "Female","Male": 2 2 2 1 1 1 2 2 2 1 ...
## $ TREAT    : Factor w/ 3 levels "Control","Cooled",...: 2 2 2 2 2 2 2 2 2 2 ...
## $ WEIGHT: num  8.6 9.49 8.66 9.32 10.07 ...
```

```
levels(data2$TREAT)
```

```
## [1] "Control" "Cooled"  "Warmed"
```

Explore the data to see the effects of treatments and sex on growth. Play around with the following code to understand how it works.

```
require(dplyr)
```

```
## Loading required package: dplyr
```

```
##
```

```
## Attaching package: 'dplyr'
```

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

```
##
```

```
## collapse
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
## filter, lag
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
## intersect, setdiff, setequal, union
```

```
data2 %>%
```

```
  group_by(TREAT) %>%
```

```
  summarise(mean_WEIGHT=mean(WEIGHT))
```

```
## # A tibble: 3 x 2
```

```
##   TREAT mean_WEIGHT
```

```
##   <fctr>      <dbl>
```

```
## 1 Control    9.990555
```

```
## 2 Cooled     9.515473
```

```
## 3 Warmed     10.240153
```

```
data2 %>%
```

```
  group_by(TREAT, SEX) %>%
```

```
  summarise(mean_WEIGHT=mean(WEIGHT))
```

```
## # A tibble: 6 x 3
```

```
## # Groups:   TREAT [?]
```

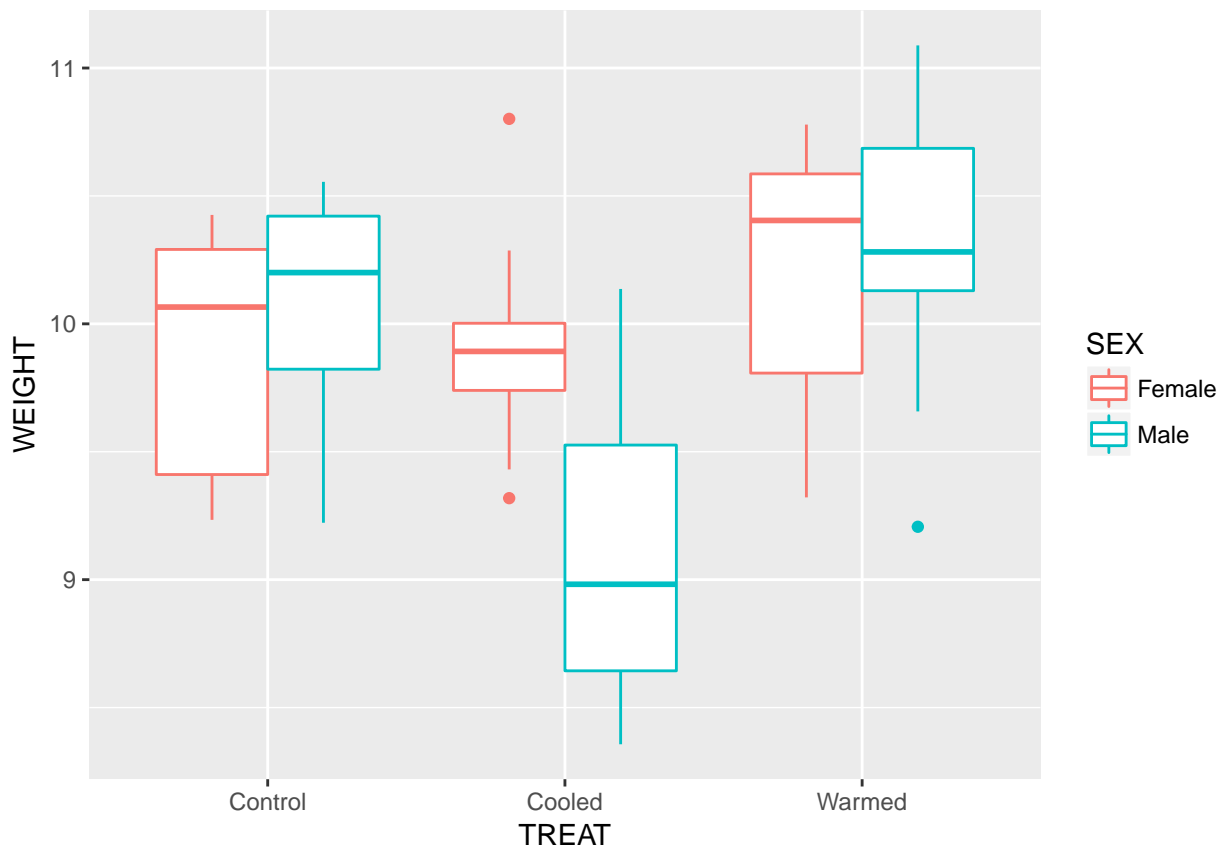
```
##   TREAT SEX mean_WEIGHT
```

```
##   <fctr> <fctr>      <dbl>
```

```
## 1 Control Female    9.899443
```

```
## 2 Control   Male   10.081668
## 3 Cooled   Female  9.907605
## 4 Cooled   Male   9.123340
## 5 Warmed   Female  10.182789
## 6 Warmed   Male   10.297516
```

```
require(ggplot2)
p <- qplot(data=data2,
            x=TREAT,
            y=WEIGHT,
            colour=SEX,
            geom="boxplot")
print(p)
```



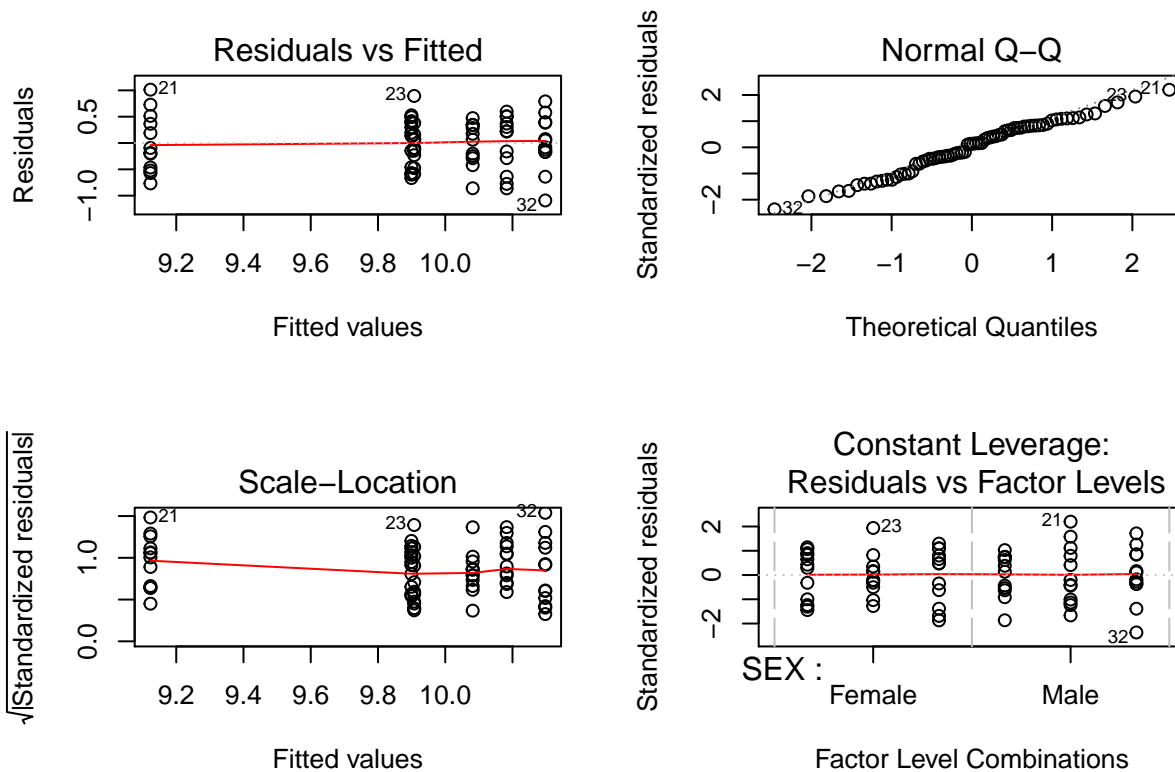
Do you think there is an effect of nest box, treatment or sex?

Briefly ignore the grouping in the data and fit a simple linear model to test for the effects of treatment and sex

```
chicklm1<-lm(WEIGHT~SEX*TREAT,
             data=data2)
```

View model diagnostics to check assumptions and try and interpret the model outputs. From the anova table see if there's a treatment effect?

```
par(mfrow=c(2,2))# to plot four graphs on one sheet
plot(chicklm1)
```



```
summary(chicklm1)
```

```
##
## Call:
## lm(formula = WEIGHT ~ SEX * TREAT, data = data2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.09080 -0.28797  0.06284  0.36663  1.01295
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.899443   0.138885  71.278 < 2e-16 ***
## SEXMale         0.182225   0.196413   0.928 0.356911
## TREATCooled     0.008162   0.196413   0.042 0.966978
## TREATWarmed     0.283346   0.196413   1.443 0.153861
## SEXMale:TREATCooled -0.966490  0.277770  -3.479 0.000895 ***
## SEXMale:TREATWarmed -0.067498  0.277770  -0.243 0.808760
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4811 on 66 degrees of freedom
## Multiple R-squared:  0.4067, Adjusted R-squared:  0.3618
## F-statistic:  9.05 on 5 and 66 DF,  p-value: 1.371e-06
```

```
anova(chicklm1)
```

```
## Analysis of Variance Table
##
## Response: WEIGHT
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## SEX           1  0.4749   0.4749   2.0519  0.156738
## TREAT          2  6.5053   3.2527  14.0522 8.238e-06 ***
## SEX:TREAT      2  3.4937   1.7468   7.5468  0.001118 **
## Residuals    66 15.2770   0.2315
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

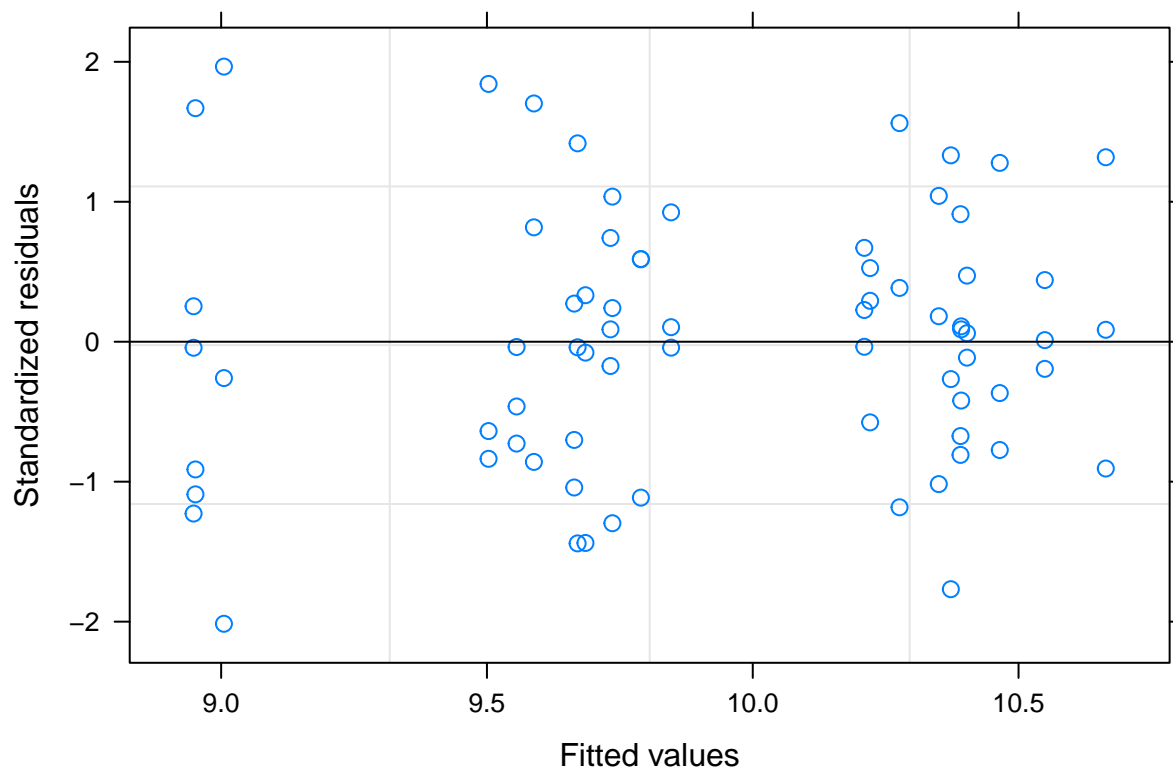
However, this analysis is wrong and does not taken into account the grouping of the data. Chicks within the same nest box are not independent data points.

Specify the model as a mixed model with Box as the random effect. The random term will account for variation due to box.

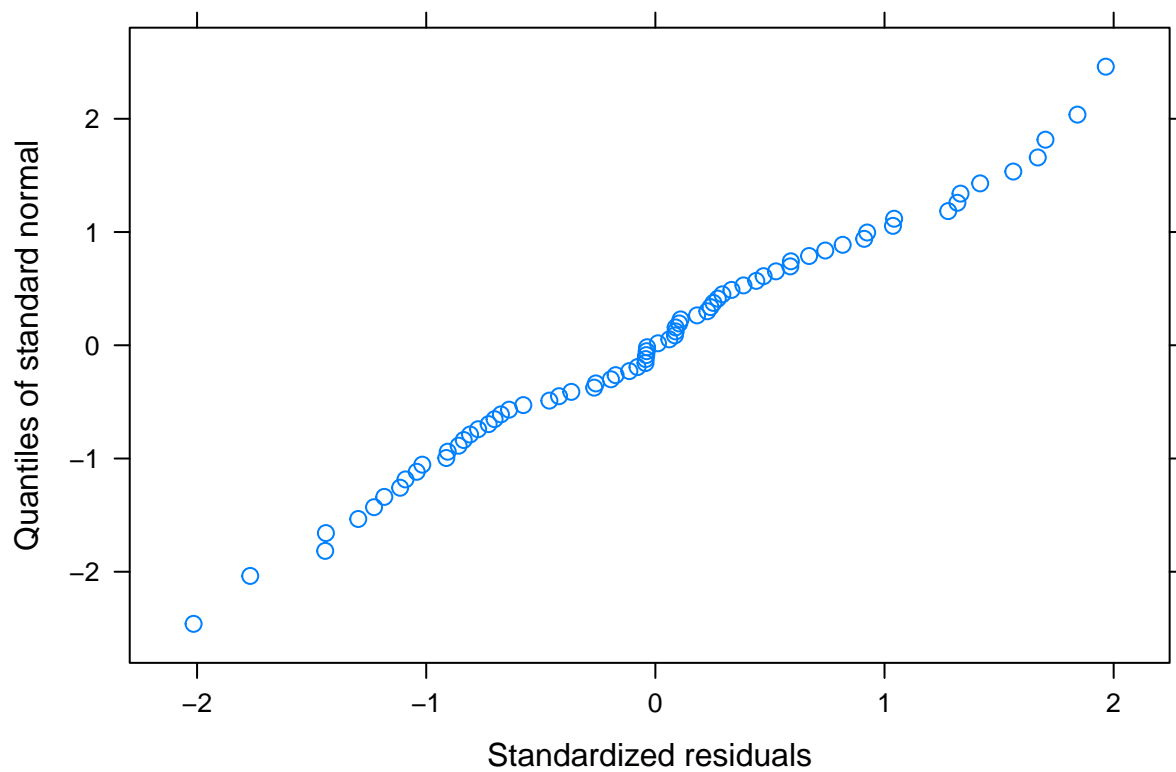
```
chicklme1<-lme(WEIGHT~SEX*TREAT,random=~1|BOX,
               data=data2)
```

Graphs for model diagnostics have to be specified more explicitly for mixed models, but should be checked in a similar way to standard linear models. The following will give you plots of fits vs. residuals, normality of residuals, and normality of random effects respectively.

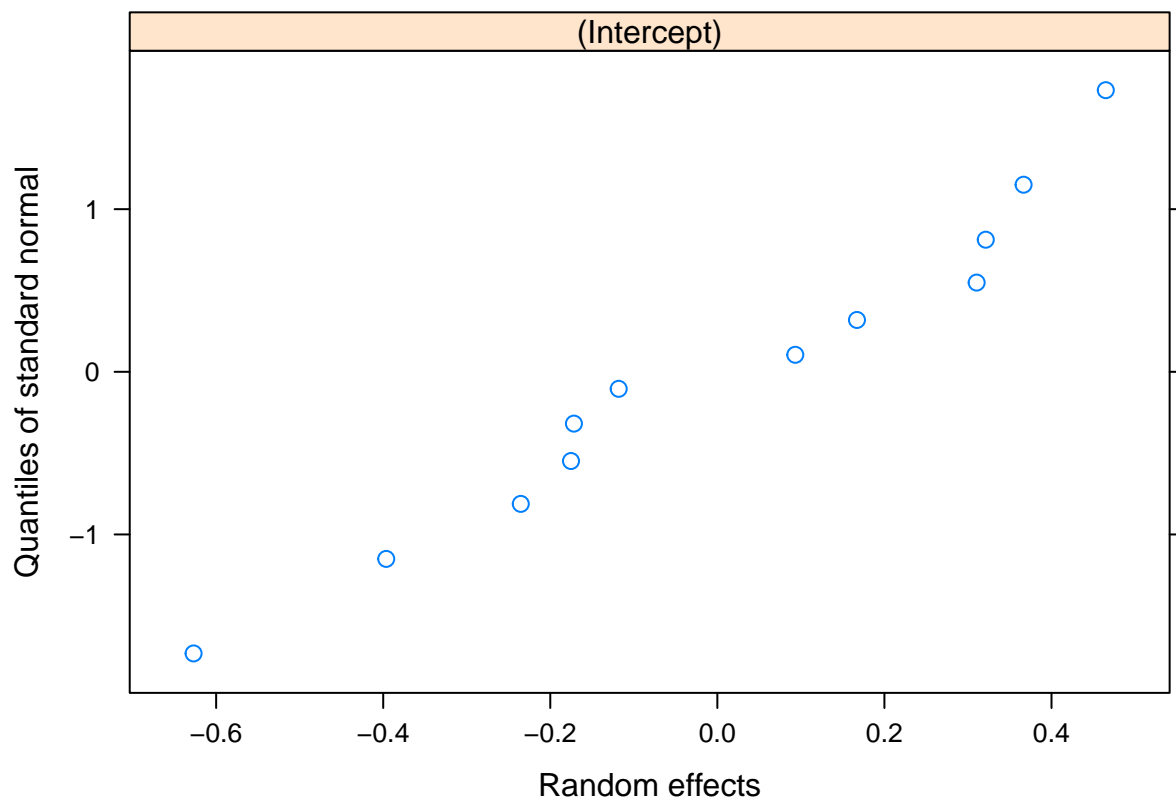
```
plot(chicklme1)
```



```
qqnorm(chicklme1)
```



```
qqnorm(chicklme1,~ranef(.))
```



The significance of fixed terms in mixed models can be tested by looking at the `anova(model)` output for the model, which will give conditional F-tests. Remember the terms are tested

sequentially so the order of terms in the model can matter!

Look at the summary and anova outputs.

```
summary(chicklme1)
```

```
## Linear mixed-effects model fit by REML
## Data: data2
##      AIC      BIC    logLik
##  89.34314 106.8604 -36.67157
##
## Random effects:
## Formula: ~1 | BOX
##      (Intercept)  Residual
## StdDev:    0.3953251 0.3218725
##
## Fixed effects: WEIGHT ~ SEX * TREAT
##              Value Std.Error DF   t-value p-value
## (Intercept)    9.899443 0.2184124 57 45.32455 0.0000
## SEXMale        0.182225 0.1314039 57  1.38676 0.1709
## TREATCooled    0.008162 0.3088818  9  0.02643 0.9795
## TREATWarmed    0.283346 0.3088818  9  0.91733 0.3829
## SEXMale:TREATCooled -0.966490 0.1858332 57 -5.20085 0.0000
## SEXMale:TREATWarmed -0.067498 0.1858332 57 -0.36322 0.7178
## Correlation:
##              (Intr) SEXMal TREATC TREATW SEXM:TREATC
## SEXMale      -0.301
## TREATCooled  -0.707  0.213
## TREATWarmed  -0.707  0.213  0.500
## SEXMale:TREATCooled 0.213 -0.707 -0.301 -0.150
## SEXMale:TREATWarmed 0.213 -0.707 -0.150 -0.301  0.500
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -2.0153036 -0.7082614 -0.0123904  0.5418996  1.9656051
##
## Number of Observations: 72
## Number of Groups: 12
```

```
anova(chicklme1)
```

```
##      numDF denDF  F-value p-value
## (Intercept)    1    57 6797.970 <.0001
## SEX           1    57   4.584 0.0366
## TREAT         2     9   3.124 0.0933
## SEX:TREAT      2    57  16.861 <.0001
```

Is there a treatment effect?

Compare these outputs with the outputs from the linear model – remember you can copy code and outputs into a script file (go to File>New Script).

Note that we have explained more variation in the data by incorporating the box effect as a random term (stddev of box effect = 0.395), and the detected interaction term is larger (compare the F-values in the anova table) in the lme than in the lm.

Other useful information can be seen by:

`random.effects(chicklme1)` to get the predicted random effects (i.e. deviation from the population mean)

for each box.

```
random.effects(chicklme1)
```

```
##      (Intercept)
## 1  -0.17168099
## 2  -0.17521212
## 3  -0.11810409
## 4   0.46499720
## 5   0.16713960
## 6  -0.62701047
## 7   0.09329385
## 8   0.36657701
## 9   0.31039639
## 10 -0.23530383
## 11 -0.39640415
## 12  0.32131159
```

intervals(chicklme1) to get 95% confidence intervals on model parameters.

```
intervals(chicklme1)
```

```
## Approximate 95% confidence intervals
##
## Fixed effects:
##               lower      est.      upper
## (Intercept)   9.46207944  9.899442727 10.3368060
## SEXMale      -0.08090668  0.182225112  0.4453569
## TREATCooled  -0.69057686  0.008162266  0.7069014
## TREATWarmed  -0.41539279  0.283346331  0.9820855
## SEXMale:TREATCooled -1.33861464 -0.966490092 -0.5943655
## SEXMale:TREATWarmed -0.43962227 -0.067497716  0.3046268
## attr("label")
## [1] "Fixed effects:"
##
## Random Effects:
## Level: BOX
##               lower      est.      upper
## sd((Intercept)) 0.236586  0.3953251  0.6605714
##
## Within-group standard error:
##      lower      est.      upper
## 0.2678934  0.3218725  0.3867282
```

VarCorr(chicklme1) to get the variance and standard deviations of the random effects and residuals (note: we already get the standard deviations in the summary output.

```
VarCorr(chicklme1)
```

```
## BOX = pdLogChol(1)
##           Variance StdDev
## (Intercept) 0.1562819 0.3953251
## Residual    0.1036019 0.3218725
```

Using `predict(model, level=0)` you will get the predictions of the way the response varies due to the fixed effects for each treatment (i.e. the average effects for nest box treatment and sex) while `predict(model, level=1)` will give you the predictions of the fixed effects plus the random effect for each level of the random factor (i.e. effects of treatment, sex and box). Run the following

code to understand what this means!

Plot the weight gain predictions of the model.

1. Plot fixed effects

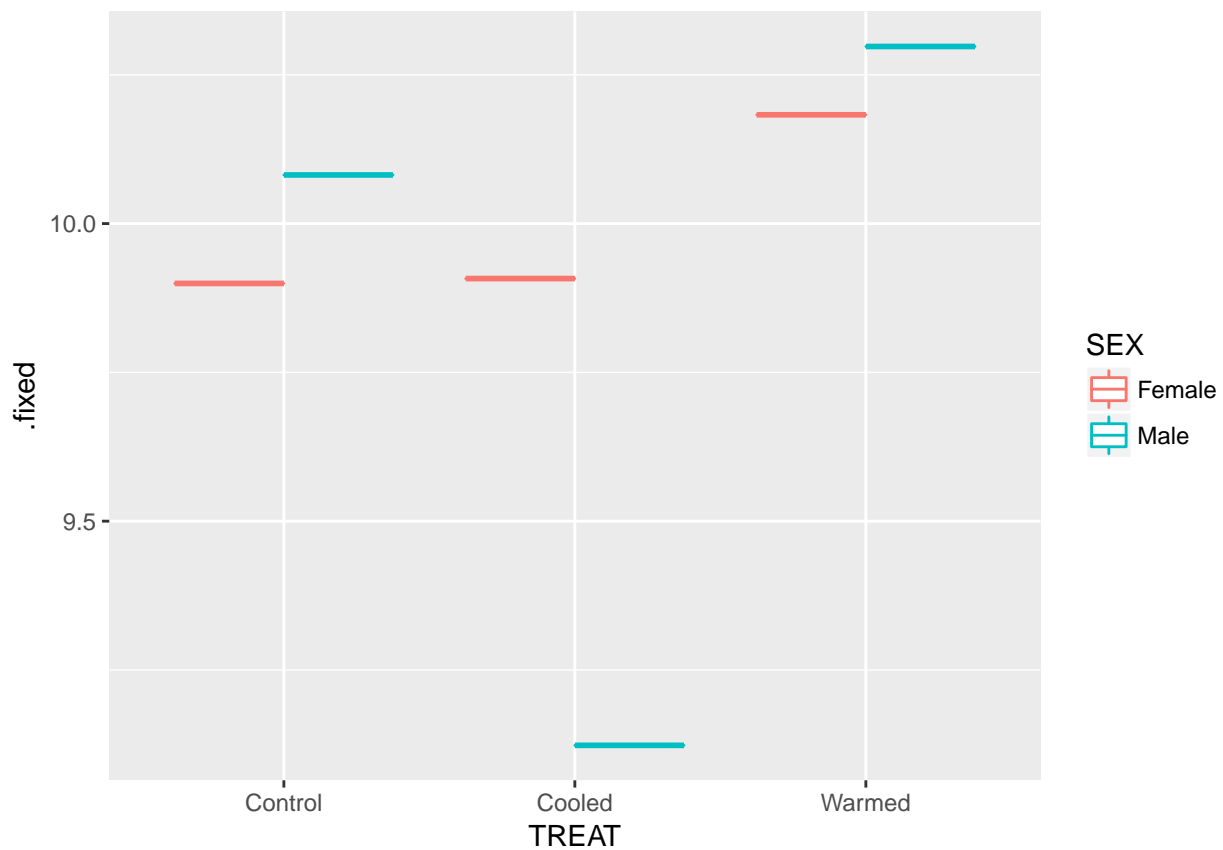
```
require(broom)
```

```
## Loading required package: broom
```

```
augmented_lme1 <- augment(chicklme1)
```

```
p <- qplot(data=augmented_lme1,  
          y=.fixed,  
          x=TREAT,  
          colour=SEX,  
          geom="boxplot")
```

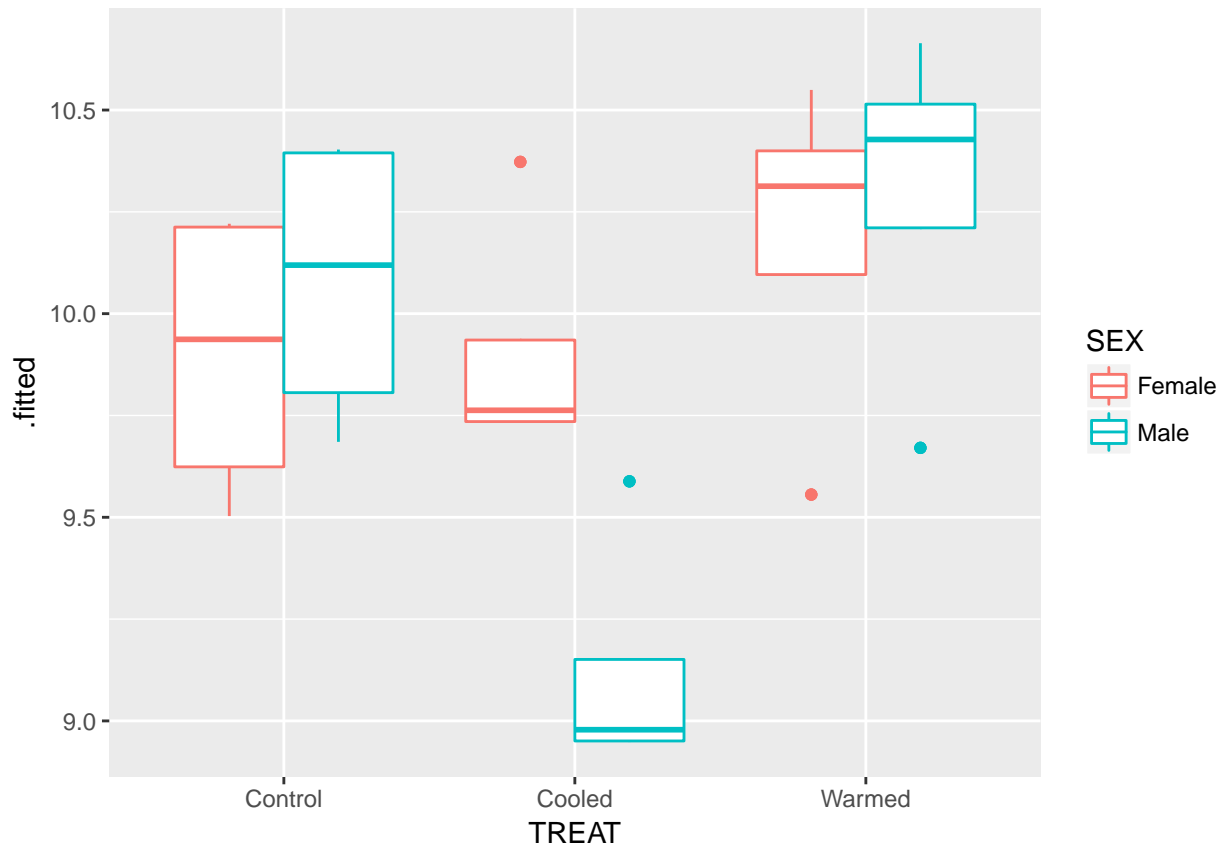
```
print(p)
```



2. Plot fixed and random effects

```
p <- qplot(data=augmented_lme1,  
          y=.fitted,  
          x=TREAT,  
          colour=SEX,  
          geom="boxplot")
```

```
print(p)
```



Note that in the first graph, there is a unique prediction for each of the 4 boxes in a treatment group because of the random term for box.

Example 3: Longitudinal data

Data: Rats (n=4 per group) were fed on one of two diets and subsequently weighed at 11 time points until day 64.

```
data3<-read.csv("./Data/data3.csv")
names(data3)

## [1] "weight" "Time"   "Rat"    "Diet"

str(data3)

## 'data.frame':   88 obs. of  4 variables:
##  $ weight: int  410 415 425 428 438 443 442 446 456 468 ...
##  $ Time : int  1 8 15 22 29 36 43 44 50 57 ...
##  $ Rat : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Diet : int  1 1 1 1 1 1 1 1 1 1 ...

data3$Diet<-factor(data3$Diet)
data3$Rat<-factor(data3$Rat)
```

Estimate mean differences in weight between diets.

```
data3 %>%
  group_by(Diet) %>%
```

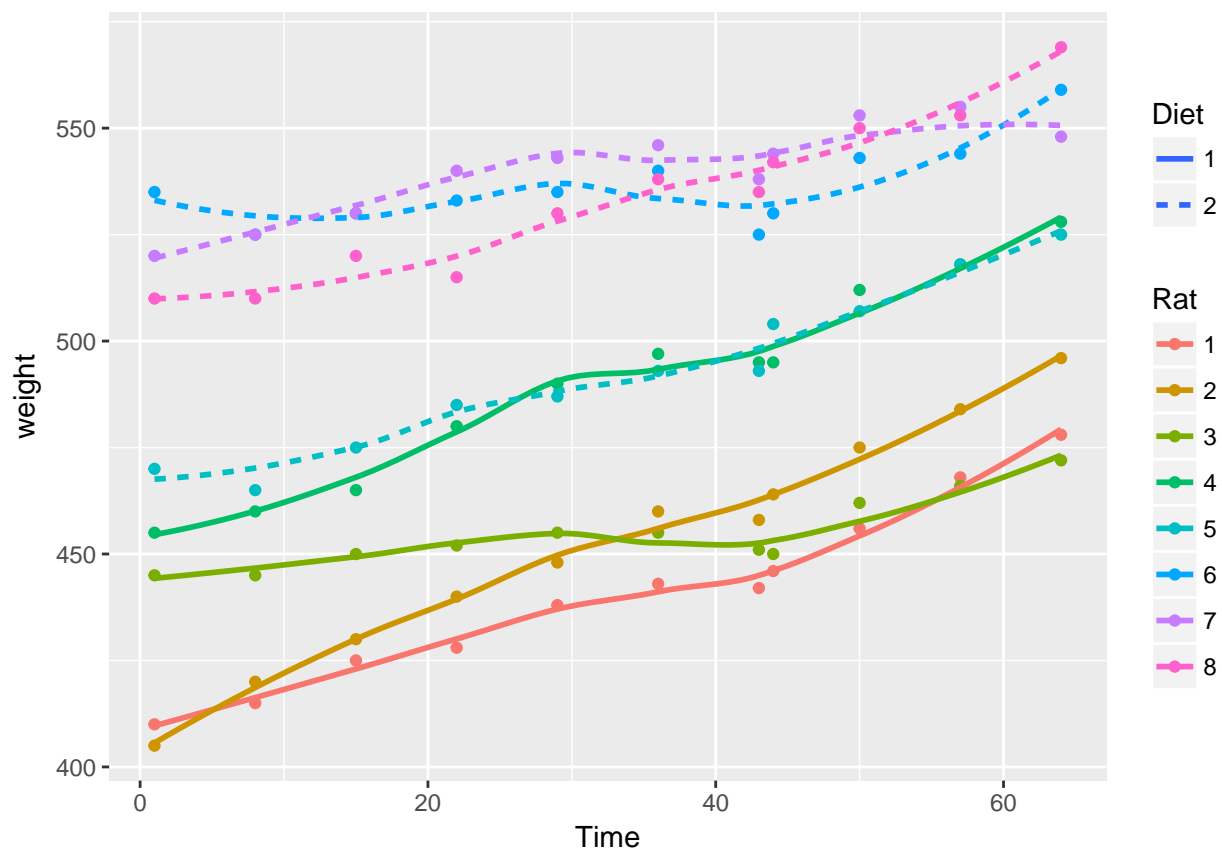
```
summarise(mean_weight=mean(weight))
```

```
## # A tibble: 2 x 2
##   Diet mean_weight
##   <fctr>      <dbl>
## 1     1    459.7045
## 2     2    525.7955
```

Produce some plots e.g.

```
p <- qplot(data=data3,
           x=Time,
           y=weight,
           colour=Rat)+
  geom_smooth(aes(linetype=Diet), se=F)
print(p)
```

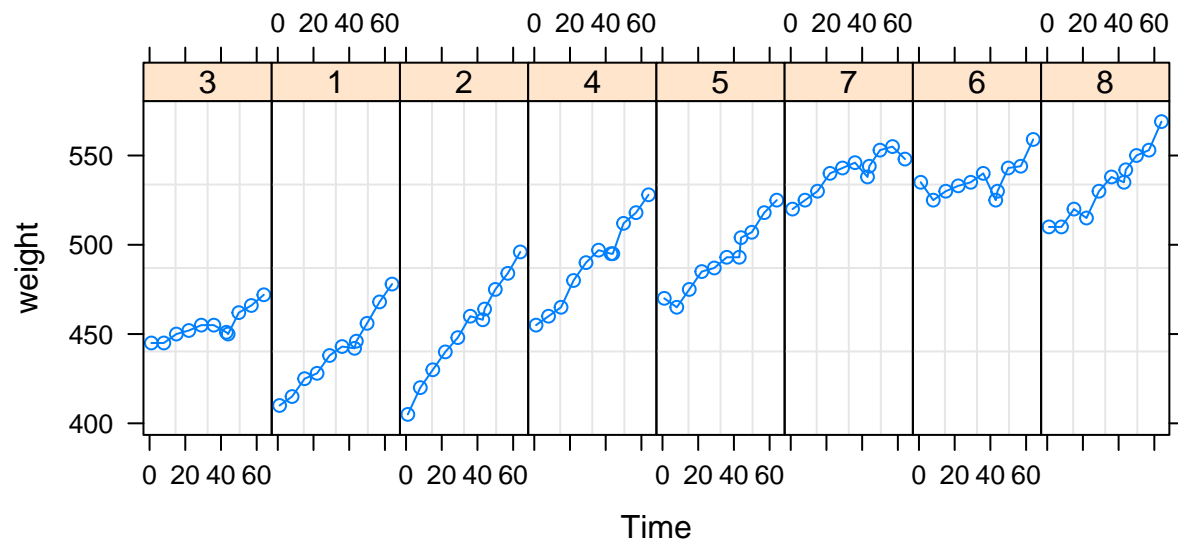
```
## `geom_smooth()` using method = 'loess'
```



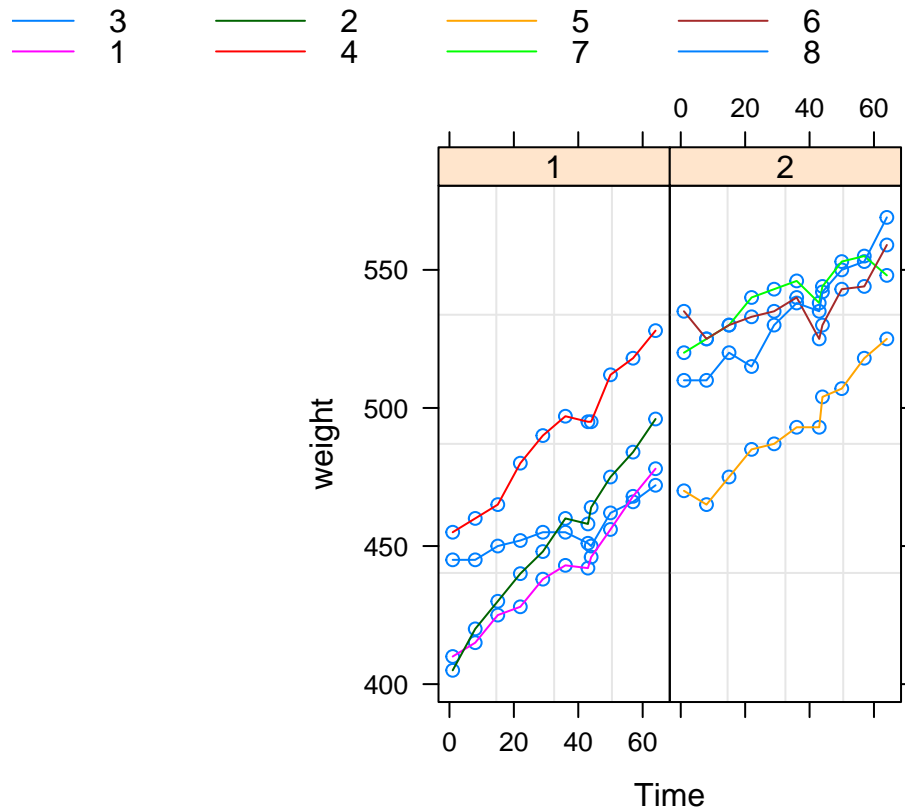
Data on the same rat is not independent so, as in previous models here, we can incorporate a random term for rat which will allow the intercept to vary between rats. In longitudinal data like this, we can also allow the slope to vary between rats by incorporating random slopes. This will become clear after running the following code!!

For this model, we will firstly specify the data as a groupedData object (from library nlme) which just specifies the grouping in the data and will allow us to get complicated graphs simply. “Outer” refers to the outer, or highest level grouping variable.

```
gd1<-groupedData(weight~Time|Rat,outer=~Diet,
                 data=data3)
plot(gd1)
```



```
plot(gd1,outer=T)
```



Now undertake two models, one with random intercepts ($\text{random}=\sim 1|\text{Rat}$) and one with both random intercepts and slopes ($\text{random}=\sim \text{Time}|\text{Rat}$).

random intercepts:

```
ratlme1<-lme(weight~Time*Diet,random=~1|Rat,
             data=gd1)
```

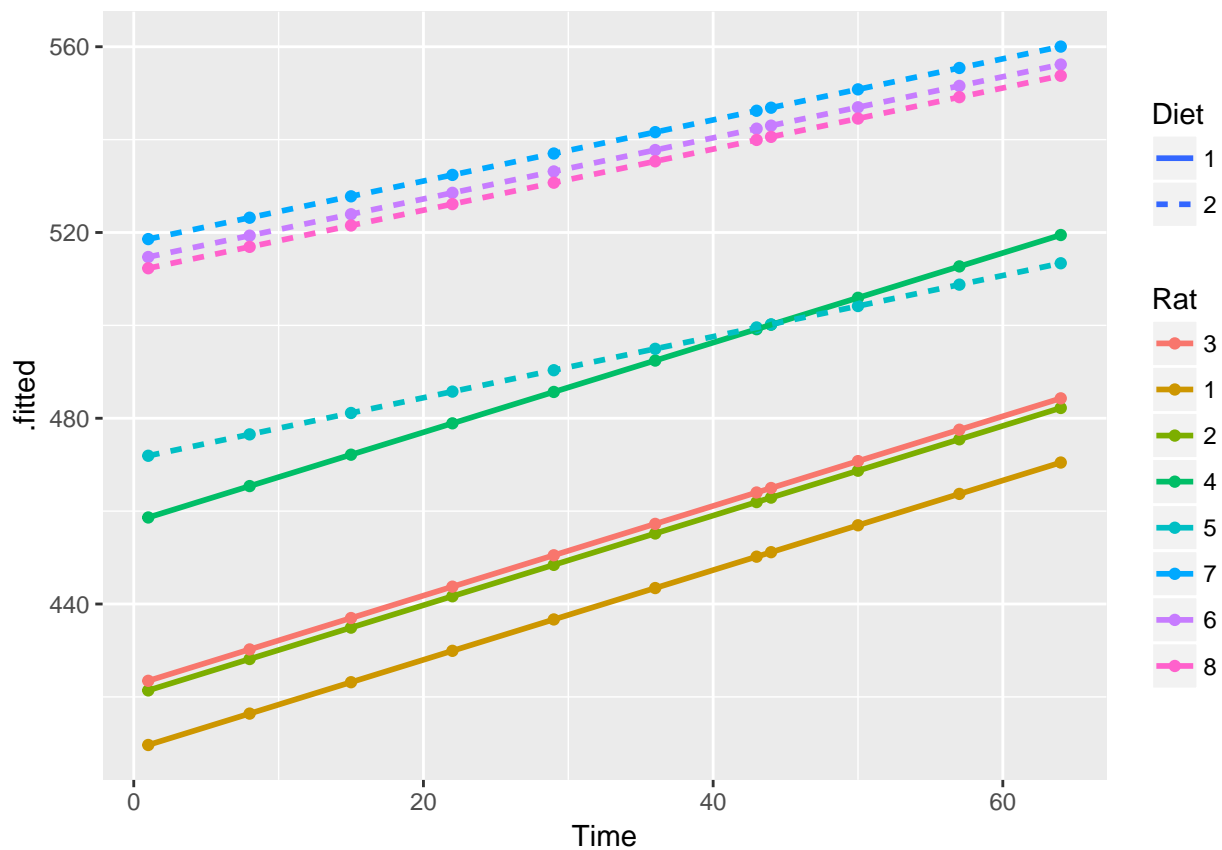
random slopes and intercepts:

```
ratlme2<-lme(weight~Time*Diet,random=~Time|Rat,
             data=gd1)
```

Plot the predictions of these models to see the difference between fitting these random terms.

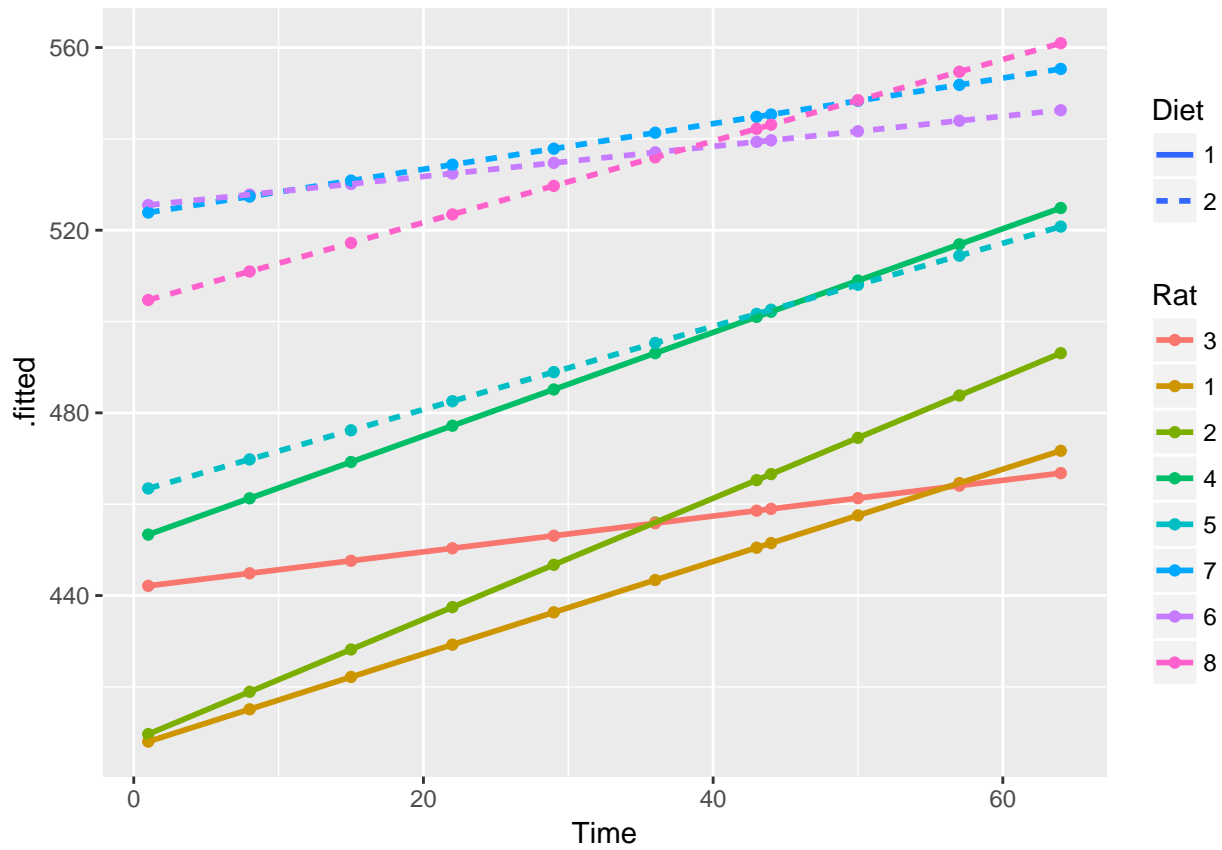
```
augmented_ratlme1 <- augment(ratlme1)
p <- qplot(data=augmented_ratlme1,
           x=Time,
           y=.fitted,
           colour=Rat)+
  geom_smooth(aes(linetype=Diet), se=F)
print(p)
```

```
## `geom_smooth()` using method = 'loess'
```



```
augmented_ratlme2 <- augment(ratlme2)
p <- qplot(data=augmented_ratlme2,
           x=Time,
           y=.fitted,
           colour=Rat)+
  geom_smooth(aes(linetype=Diet), se=F)
print(p)
```

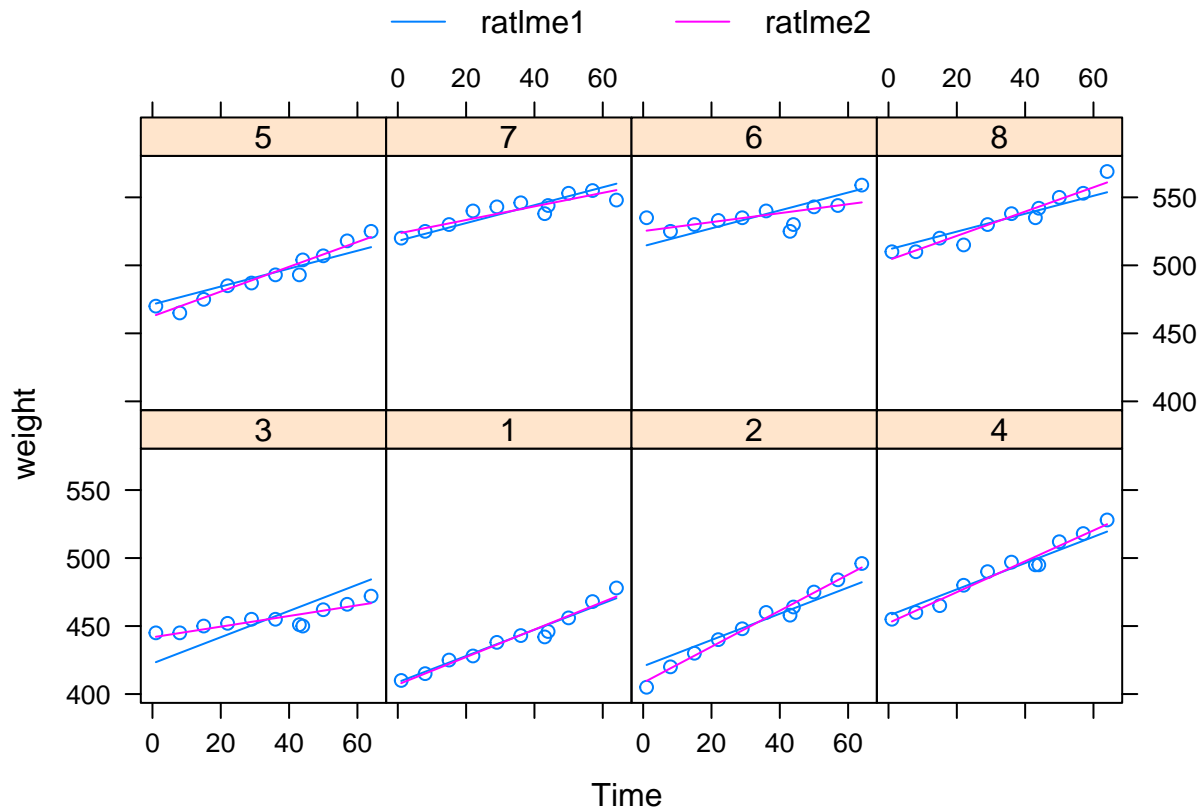
```
## `geom_smooth()` using method = 'loess'
```



Compare the graphs side by side (move the upper graph aside as they'll be on top of each other). Note that in the second graph, the rats within a treatment group vary in both the intercept and the slope of the fitted line.

You can also compare the predictions of each model for each rat with the following:

```
plot(comparePred(ratlme1, ratlme2), length.out=2)
```

The significance of random terms can be tested by comparing the model with the random term with the model without the random term e.g. `anova(model1,model2)`. This will give a likelihood ratio test for the significance of the random term.

Does the model with random slopes and random intercepts provide a significantly better fit to the data than the model with only random intercepts? Test the significance of including random slopes.

```
anova(ratlme1, ratlme2)
```

##	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
##	ratlme1	1	657.3615	671.9464	-322.6808			
##	ratlme2	2	605.5888	625.0354	-294.7944	1 vs 2	55.77272	<.0001

Also, look at the `anova()` outputs of each model? What happens to the `time:diet` interaction? Can you think why this might be?

Example 4: Multiple nested terms

Data: Maths scores of male and female students taken from 187 students in total from 10 schools – note we have more than one score for each student, and more than one student per school. What explains the variation in scores?

Data can be grouped at two nested levels: maths scores from the same student and scores from students in the same school.

```
data4<-read.csv("./Data/data4.csv")
names(data4)
```

```
## [1] "schoolId" "studentId" "score" "sex"
```

```
str(data4)
```

```
## 'data.frame': 774 obs. of 4 variables:
## $ schoolId : int 4410 4410 4410 4410 4410 4410 4410 4410 4410 4410 ...
## $ studentId: int 281518961 281518961 281518961 281518961 281518961 281518961 291417891 291417891 291417891 291417891 ...
## $ score : num -1.218 -0.921 -0.419 0.516 0.798 ...
## $ sex : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 2 2 2 2 2 ...
```

Explore the data using averaging and plots.

```
mathslme1<-lme(score~sex,random=~1|schoolId/studentId,
               data=data4)
```

Make sure you understand the code for this model.

Look at the model diagnostic graphs to check model assumptions as before. Note that now we have two random intercept terms, we need to specify `ranef(model, level=1)` or `ranef(model, level=2)` to get the predicted random effects for school (level=1) or student (level=2) (e.g. `qqnorm(mathslme1,~ranef(.,level=2))`)

Look at `summary()` or `VarCorr()` to compare the estimates for each random term. Which is more important? Test for the significance of the “student” effect?

```
mathslme1<-lme(score~sex,random=~1|schoolId/studentId,
               data=data4)
mathslme2<-update(mathslme1,random=~1|schoolId,
                  data=data4)
anova(mathslme1,mathslme2)
```

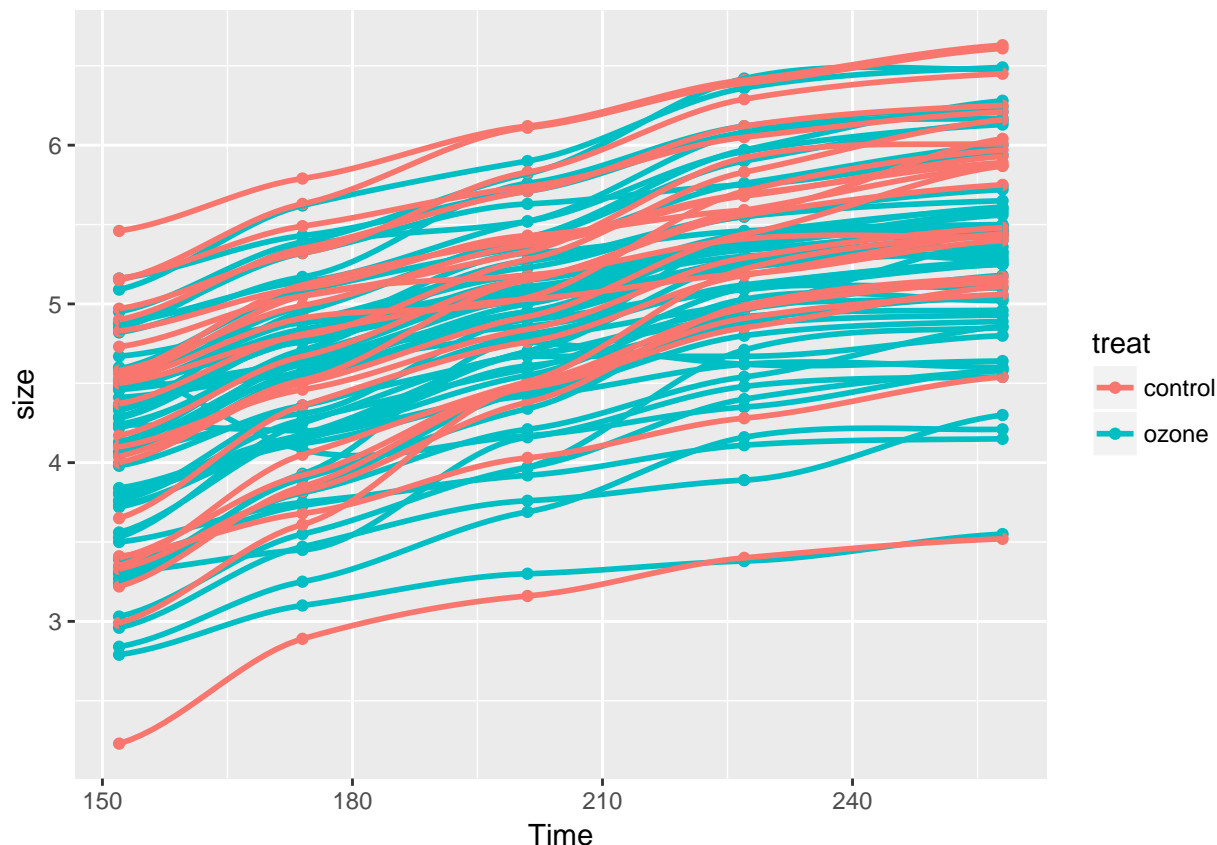
##	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
##	mathslme1	1	5	2740.707	2763.952	-1365.354		
##	mathslme2	2	4	2830.950	2849.546	-1411.475	1 vs 2	92.24317 <.0001

Test the schoolId random term and compare which random term is more important? Is there an effect of sex on scores?

Example 5:

Data: 79 spruce trees were measured 5 times over consecutive months that were in either ozone enriched chambers or control chambers.

```
require(MASS)
?Sitka
p <- qplot(data=Sitka,
           y=size,
           x=Time,
           colour=treat,
           group=tree)+
  geom_smooth(se=F)
print(p)
```



Analyse it! Clearly, the model structure needs to take into account the repeated measures (so tree must be a random factor) (but the growth of each tree could vary in intercept alone, or both slope and intercept)...

The traditional way of analysing a dataset such as this would be either to use a formal repeated measures' ANOVA or to estimate the growth rate of each tree (either by putting a regression through the data for each tree and taking the slope/intercept, or by some simple rate calculation $((\text{end size} - \text{initial size}) / \text{time})$). For the former, the standard approach would be to use the aov command (in MASS library). E.g.

```
rm1<-aov(size ~ treat*Time + Error(factor(tree)),
          data=Sitka)
summary(rm1)
```

```
##
## Error: factor(tree)
##           Df Sum Sq Mean Sq F value Pr(>F)
## treat      1   3.81   3.810    2.019  0.159
## Residuals 77 145.31   1.887
##
## Error: Within
##           Df Sum Sq Mean Sq F value   Pr(>F)
## Time        1  89.56   89.56 2383.47 < 2e-16 ***
## treat:Time   1   0.55    0.55   14.65 0.000156 ***
## Residuals  314  11.80    0.04
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For the latter: a simple way to get the tree specific data, would be to put regressions thru all trees and look at the tree-specific variation in the intercept or slope e.g.

```
options(contrasts=c("contr.treatment", "contr.poly"))
mod<-lm(size~Time*factor(tree)-1,
        data=Sitka)
```

And these coefs could then be the dependent variable in a lm. So first extract the intercepts and slopes

```
treeintercepts<-as.numeric(mod$coef[2:80])
treeslopes<- as.numeric(mod$coef[1]-c(0,mod$coef[81:158]))
```

Set up a new treatment column:

```
t2<-rep(0,79)
for(i in 1:79){t2[i]<-Sitka$treat[((i-1)*5)+1]} #what IS this doing?
s2<-data.frame(cbind(treeslopes,treeintercepts,t2))
rm2a<-lm(treeintercepts~t2,s2)
rm2b<-lm(treeslopes~t2,s2)
summary(rm2a)
```

```
##
## Call:
## lm(formula = treeintercepts ~ t2, data = s2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.0093 -0.6814  0.1632  0.6296  1.7657
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.9000     0.3791   5.012 3.35e-06 ***
## t2             0.2217     0.2170   1.021   0.31
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8972 on 77 degrees of freedom
## Multiple R-squared:  0.01337,    Adjusted R-squared:  0.0005523
## F-statistic: 1.043 on 1 and 77 DF,  p-value: 0.3103
```

```
summary(rm2b)
```

```
##
## Call:
## lm(formula = treeslopes ~ t2, data = s2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.007735 -0.002221  0.000059  0.001968  0.010779
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0151598  0.0013493  11.235 < 2e-16 ***
## t2           0.0021385  0.0007725   2.768  0.00706 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.003194 on 77 degrees of freedom
## Multiple R-squared:  0.09051,    Adjusted R-squared:  0.0787
```

F-statistic: 7.663 on 1 and 77 DF, p-value: 0.007058

Run the lm and compare it to a properly specified (intercepts varying only) lme.

```
rm3<-lme(size~Time*treat,random=~1|factor(tree),Sitka)
summary(rm3)
```

```
## Linear mixed-effects model fit by REML
## Data: Sitka
##      AIC      BIC    logLik
## 175.517 199.3293 -81.75852
##
## Random effects:
## Formula: ~1 | factor(tree)
##      (Intercept) Residual
## StdDev:   0.6082011 0.1938483
##
## Fixed effects: size ~ Time * treat
##              Value Std.Error DF   t-value p-value
## (Intercept)  2.1217179 0.15439225 314 13.742386  0.0000
## Time         0.0141472 0.00046190 314 30.628557  0.0000
## treatozone   0.2216775 0.18674206  77  1.187078  0.2388
## Time:treatozone -0.0021385 0.00055868 314 -3.827802  0.0002
## Correlation:
##              (Intr) Time   tretzn
## Time         -0.606
## treatozone   -0.827  0.501
## Time:treatozone 0.501 -0.827 -0.606
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -2.6949604 -0.4923652  0.1176021  0.5733781  4.3279068
##
## Number of Observations: 395
## Number of Groups: 79
```

```
anova(rm3)
```

```
##              numDF denDF F-value p-value
## (Intercept)      1    314 4904.888 <.0001
## Time            1    314 2383.472 <.0001
## treat           1     77   2.019 0.1594
## Time:treat      1    314  14.652 0.0002
```

Do the three methods give similar conclusions?