Linear mixed models, part 2: random interactions

Timothée Bonnet

April 16, 2020

On Friday 17/04/2020 I will be presenting this tutorial live on Zoom.

If you have any trouble going through this tutorial then or at a different time you can chat about it on Slack (rsb-r-stats-biology.slack.com , if you are not a member but would like to be, drop me an email) or email me at timotheebonnetc@gmail.com.

If you do not attend the Zoom meeting but would like to receive credit through the COS Career Development Framework program I need you to complete three exercises of your choice. Send me your answers via Slack or email. It does not have to be correct on the first try and you are welcome to get in touch if you are completely stuck. I will provide feedback to help you complete exercises you want to do.

In this tutorial you will learn:

- How to fit random intercept and random interaction models in R
- How to interpret random effects and extract relevant numbers from mixed models output
- Visualize random interactions
- Test for the statistical significance of random effects

You will need some data, available at https://github.com/timotheenivalis/RSB-R-Stats-Biology/raw/master/06.MixedModels2/DataMM2.zip. Unzip in your R project directory.

Contents

1 Reminder: random intercepts **2**

Exercise 1 *Random intercept model practice* 7

2 Random interactions **7**

2.1 Random slopes 9

Exercise 2 *Random slopes and unbalanced data* 10

Exercise 3 *Visualize random slopes* 10

Exercise 4 *Does natural selection vary?* 10

2.2 Random factor interaction 11

Exercise 5 *Random interaction with a factor* 11

Exercise 6 *Beetles: build a model* 11

Exercise 7 *Beetles: look at the model* 11

Exercise 8 *Beetles: interpret* 11

2.3 Is there still time? Then let's talk about correlation between random intercept and random slope! 12

3 More resources **12**

1 Reminder: random intercepts

The simplest and most common mixed models use random effects to model differences in the intercepts of grouping factors.

At RSB, random intercepts are most often used to account for structure in the response variable, often due to the experimental design or the way data were collected.

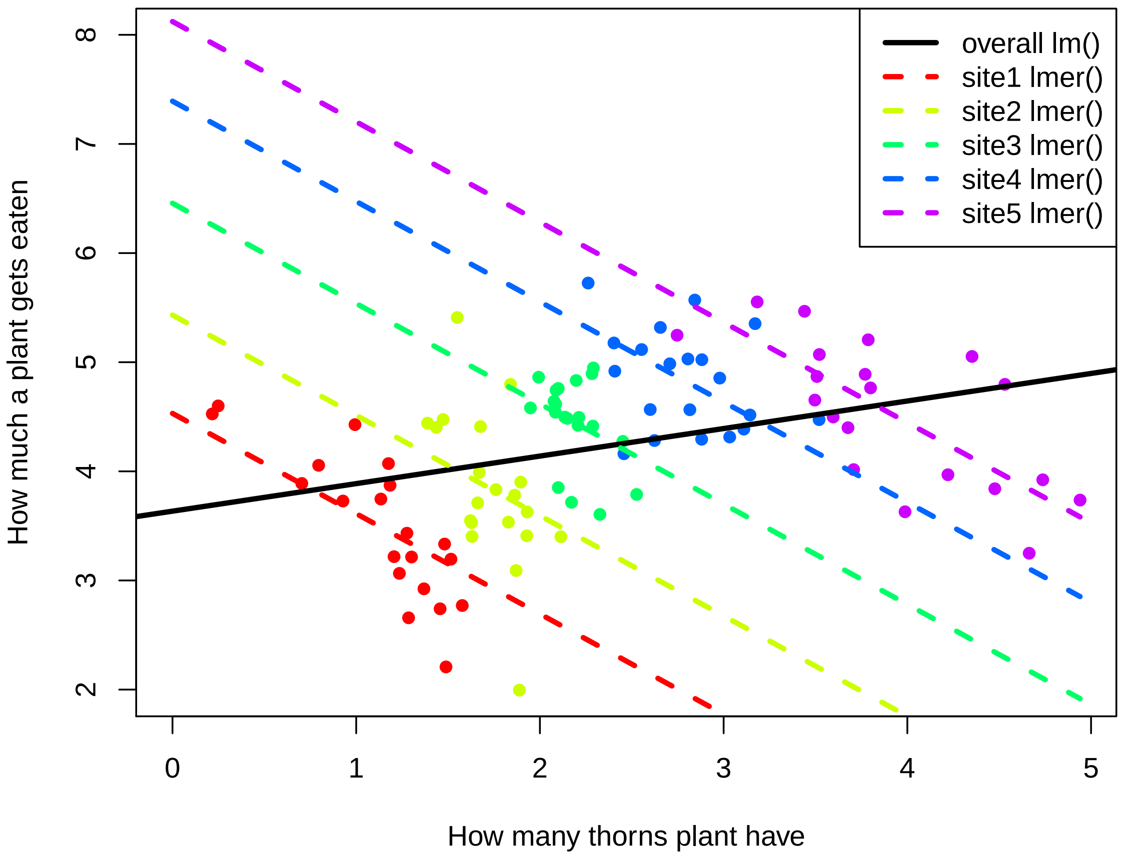
Imagine you have measured how much a plant species is attacked by big herbivores as a function of how many thorns the plant grows on stems. You have collected data in 5 locations and visualise the relationship between herbivory and quantity of thorns (in arbitrary units). The overall slope of herbivory on thorns is positive because of strong differences among sites, but actually the causal effect of thorns on herbivory is negative and we can recover this negative effect by correcting for site, for instance using a random effect.

Try the models below, one not correcting for site, the other fitting a random effect for site (to be precise, a random intercept). Look at their summaries.

```
library(lme4)
thorns <- read.csv(file = "thorndata.csv", header=TRUE)
model_0 <- lm(herbivory ~ 1 +thorndensity, data = thorns)
lmm1 <- lmer(herbivory ~ 1 +thorndensity + (1|site), data=thorns)
```

The data and regression lines of both models look like this:

```
plot(thorns$thorndensity, thorns$herbivory, col=rainbow(5)[thorns$site],
     pch=16, xlab="How many thorns plant have",
     ylab="How much a plant gets eaten", xlim = c(0,max(thorns$thorndensity)),
     ylim = c(min(thorns$herbivory),8))
abline(model_0, lwd=3)


xr <- seq(0, max(thorns$thorndensity), length.out = 3)
yr <- fixef(lmm1)[1]+fixef(lmm1)[2]*xr
for(i in 1:nrow(ranef(lmm1)$site))
{
  lines(xr, yr+ranef(lmm1)$site[i,], col=rainbow(5)[i], lwd=3, lty=2)
}


legend(x = "topright",legend = c("overall lm()",
                                 "site1 lmer()", "site2 lmer()",
                                 "site3 lmer()", "site4 lmer()",
                                 "site5 lmer()"),
       col = c("black", rainbow(5)), lwd=3, lty = c(1,rep(2,5)))
```



We have 5 parallel regression lines, one for each site. Differences in the mean values within each sites are accounted for but we estimate the common slope across all sites. All the information about this model is apparent in the summary:

```
summary(lmm1)

## Linear mixed model fit by REML ['lmerMod']
## Formula: herbivory ~ 1 + thorndensity + (1 | site)
##    Data: thorns
##
## REML criterion at convergence: 165.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.488 -0.606  0.109  0.523  2.873
##
## Random effects:
```

```
##  Groups   Name        Variance Std.Dev.
##  site     (Intercept) 2.129    1.459
##  Residual             0.238    0.488
## Number of obs: 100, groups:  site, 5
##
## Fixed effects:
##                Estimate Std. Error t value
## (Intercept)       6.387      0.731    8.74
## thorndensity     -0.919      0.139   -6.63
##
## Correlation of Fixed Effects:
##             (Intr)
## thorndensty -0.445
```

but let's practice extracting elements of the summary to better understand and visualize the model.

The global intercept of the mixed model can be extracted with:

```
fixef(lmm1)[1]

## (Intercept)
##       6.387

#or
fixef(lmm1)["(Intercept)"]

## (Intercept)
##       6.387
```

and the deviations for each site can be extracted with:

```
ranef(lmm1)$site

##   (Intercept)
## a    -1.85567
## b    -0.95470
## c     0.06988
## d     1.00527
## e     1.73522
```

so, the predicted intercepts in the 5 locations are:

```
fixef(lmm1)[1]+ranef(lmm1)$site

##   (Intercept)
## a       4.531
## b       5.432
## c       6.457
## d       7.392
## e       8.122
```

Note how the spread in these 5 intercepts is consistent with the random effect estimated standard deviation:

```
#see the standard deviation:
VarCorr(lmm1)

##  Groups   Name        Std.Dev.
##  site     (Intercept) 1.459
##  Residual             0.488

#extract the variance:
as.numeric(VarCorr(lmm1)$site)

## [1] 2.129

#extract the standard deviation:
sqrt(as.numeric(VarCorr(lmm1)$site))

## [1] 1.459
```

you expect about 95% of site deviations to fall within two standard deviations of the random effect and 2/3 to fall within one standard deviation. The standard deviation is 1.459, and the random deviations of site are:

```
ranef(lmm1)$site

##   (Intercept)
## a    -1.85567
## b    -0.95470
## c     0.06988
## d     1.00527
## e     1.73522
```

Here we did not model variation in the slope of thorndensity among sites, so there is only one slope which can be extracted as:

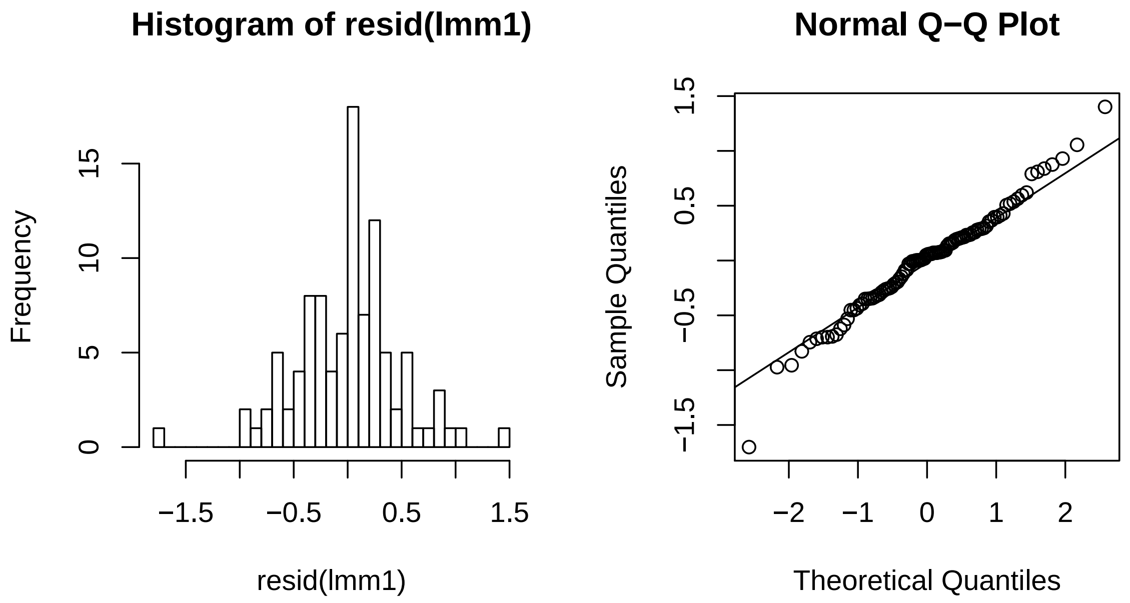
```
fixef(lmm1)[2]

## thorndensity
##      -0.9189

#or
fixef(lmm1)["thorndensity"]

## thorndensity
##      -0.9189
```

That means that the five regression lines must be parallel.
Finally, it can be good to check residuals you can extract them as:

```
lmm1resids <- resid(lmm1)
```

And visualize them:

```
par(mfrow=c(1,2))
hist(resid(lmm1), breaks = 30)
qqnorm(resid(lmm1))
qqline(resid(lmm1))
```



```
par(mfrow=c(1,1))
```

** Exercise 1 Random intercept model practice

Load the data set `fishmatechoice.csv`. It contains simplified data from an experiment by Upama Aich (thanks to her!) where female fish are presented with four male fish with different properties. Upama measured for each female the time she spends with each male (variable `Time_spent`). Each female is tested four times (on two `Trial_order` on two `Days`) with the four same males, in a tank labelled `Block` in the data set. For each test you have four rows, each corresponding to one male. Here, let's pretend that we are interested in how male Body_length influences how long a female spends with a male and we want to control for the experimental design as well as possible. As a bonus question we may want to test whether the variable "Treatment" has an effect on `Time_spent`.

1. Fit a linear mixed model of `Time_spent` with appropriate fixed and random effects.
2. What is the variance in the random effects? What is the effect of Body length?
3. Extract random and fixed effects to calculate the predicted intercept for each Block.
4. Extract the model residuals. What is the problem and where do you think it comes from? Do you know what could be done to improve the model?
5. (bonus open question) Make graphics to visualize the model results in as much details as you can.

2 Random interactions

Let's go back to the thorn example. Our previous model assumed the slope did not vary among sites, only the intercept. That does not need to be the case, and maybe there is some biological insight in checking how much the slope varies. In particular a low variation among sites would support the generality of the relationship herbivory/thorns.

We had the model, which means that the intercept (1) varies by site:

```
lmm1 <- lmer(herbivory ~ 1 + thorndensity + (1|site), data=thorns)
```

We can model variation in the intercept AND the effect of thorndensity by site as:

```
lmm2 <- lmer(herbivory ~ 1 + thorndensity + (1 + thorndensity|site), data=thorns)
```

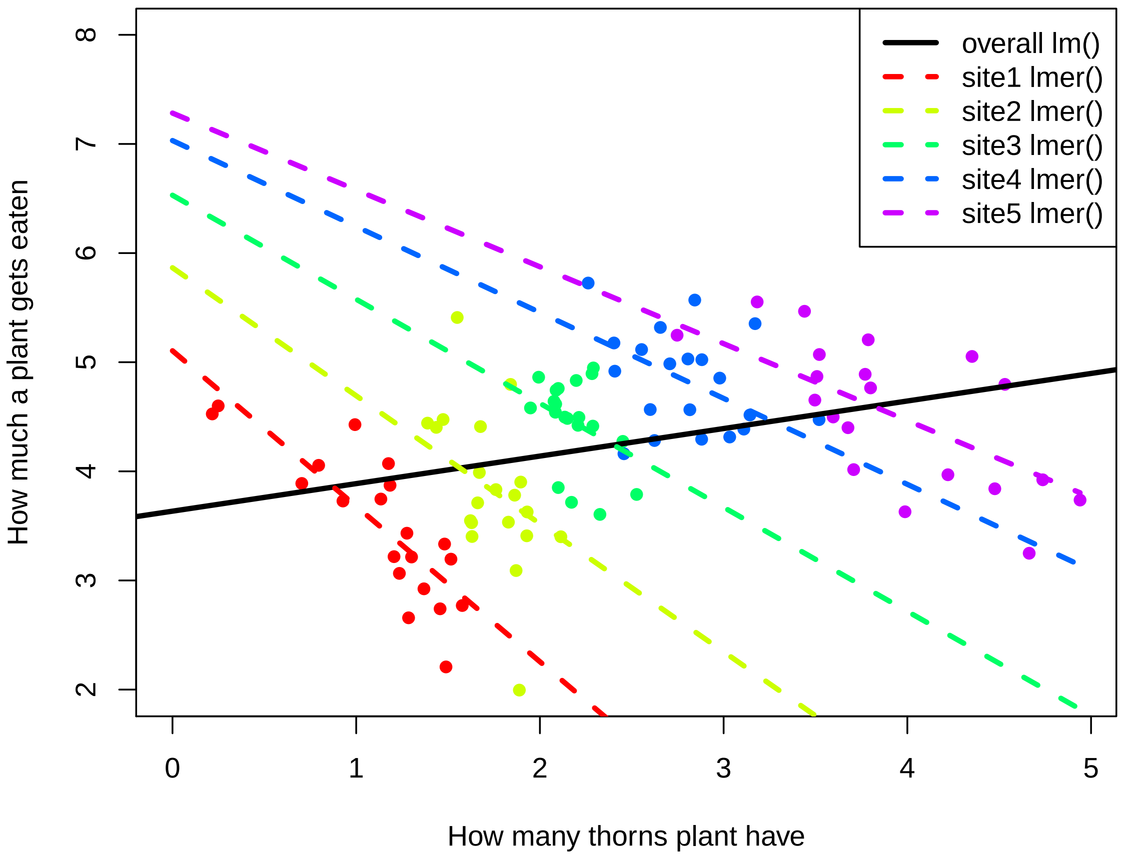
Look at the summary of lmm2.

Now we estimate two random variances (one for the intercept and one for the slope), and a correlation between those variances. Accordingly, we have two sets of random deviations:

```
ranef(lmm2)

## $site
##   (Intercept) thorndensity
## a     -1.2582     -0.41605
## b     -0.4973     -0.16445
## c      0.1671      0.05527
## d      0.6682      0.22098
## e      0.9201      0.30425
##
## with conditional variances for "site"
```

We can visualize the model as:

```
plot(thorns$thorndensity, thorns$herbivory, col=rainbow(5)[thorns$site],
     pch=16, xlab="How many thorns plant have",
     ylab="How much a plant gets eaten", xlim = c(0,max(thorns$thorndensity)),
     ylim = c(min(thorns$herbivory),8))
abline(model_0, lwd=3)


xr <- seq(0, max(thorns$thorndensity), length.out = 3)
for(i in 1:nrow(ranef(lmm2)$site))
{
  yr <- fixef(lmm2)[1]+ranef(lmm2)$site[i,1]+
    (fixef(lmm2)[2]+ranef(lmm2)$site[i,2])*xr

  lines(xr, yr, col=rainbow(5)[i], lwd=3, lty=2)
}


legend(x = "topright",legend = c("overall lm()",
                                 "site1 lmer()", "site2 lmer()",
                                 "site3 lmer()", "site4 lmer()",
                                 "site5 lmer()"),
       col = c("black", rainbow(5)), lwd=3, lty = c(1,rep(2,5)))
```



How does random effect formula work again?

Random effect formula in lme4 and many other R-packages look like (1+x|group). On the right-hand side of the pipe (|) is the variable that groups data points; for instance location, time, species. . .

On the left-hand side of the pipe is what varies according to grouping. The 1 stands for **intercept** and in random intercept models your formula will look like (1+|group). However, fixed effects other than the intercept can be included in the formula. For instance, (1+x|group) means a random effect where the intercept varies according to group and the effect of the covariate x varies according to group (and, to be exact, the variation in the intercept and in x is correlated).

2.1 Random slopes

Let’s practice with some more examples where we have a random effect on a slope.

* Exercise 2 Random slopes and unbalanced data

Load the dataset `hares.csv`. It contains (fake) measurements of snowshoe hare color (darkness) and their detectability against the background where they live. Measurements were taken in 50 different locations. We want to know whether darkness has an effect on detectability.

1. Fit a simple linear model of detectability on darkness. What is the effect?
2. Add a random intercept to the previous model. Does it change the result quantitatively?
3. Add a random slope. What do you see now and how to interpret that?

** Exercise 3 Visualize random slopes

Visualize the effect from the random slope model, that is, plot the relationship detectability over darkness for every location. Add the overall relationship (for instance extracted from a simple linear regression). You can use the functions `ranef()` and `fixef()`. Why did the fixed effect of darkness changed so much when you added the random slope?

Isn't a random intercept supposed to protect you against bias in the slope?

With the hare example we saw that a random intercept on location was not sufficient to find the negative estimate of slope we expected, we needed a random slope for that. However, we saw in the thorn example that adding a random intercept changes the direction of the slope from a wrong to a right answer. Why is a random intercept the solution sometimes and sometimes not?
A random intercept can correct for structure in the response variable related to a grouping factor, so that the slope does not simply come from differences in mean response among sites. However, a random intercept does not protect you against `imbalance` in the grouping factors (that is, some sites are sampled more often than others.) If you want to be sure your inference is general across sites you will need a random slope; otherwise, a site that is observed many more times than the other sites could drive the slope.

** Exercise 4 Does natural selection vary?

Load the dataset `AllM.txt`. It contains true data from the long term monitoring of a wild animal population. We are interested in quantifying natural selection on Weight. To simplify let's assume natural selection is the slope of `fitnessR` on `Weight`.

1. Fit a linear regression of fitnessR on Weight. Include `Age` as a predictor Is there evidence for selection?
2. Change your model to a mixed model with year as a random intercept. Do you think fitnessR varies a lot among years?

3. Now add a random slope on weight. How much variation is there in selection?

4. Make a graph to visualize selection on different years (the function `ranef()` extract random effects) (you can make the graph for adults, for juveniles, or both).

5. Looking at the estimated variance for the intercept and for Weight, which one looks more important? Is that your impression graphically? Why?

6. Bonus: test for the statistical significance of the variation in selection (you can use `anova()` to compare two models).

2.2 Random factor interaction

* Exercise 5 Random interaction with a factor

Load the file `interfactor.csv` and fit two random interaction modelw of y as a function of treat, using both the reaction norm and the character state approach. How do the estimate differ? Use the functions `AIC()`, `fitted()` and `resid()` to compare the fit of the two models? What can you conclude?

** Exercise 6 Beetles: build a model

Load the dataset "beetles.csv". It contains (fake) data from an (real) experiment on gene-by-environment interactions. The variable of interest is the mass of beetles born in two different environments, from different parents, and in different cages. Assuming that we can measure genetic varition with parent random effects, we wonder if different genomes respond differently to different environments. **Build the model corresponding to this question in lme4.**

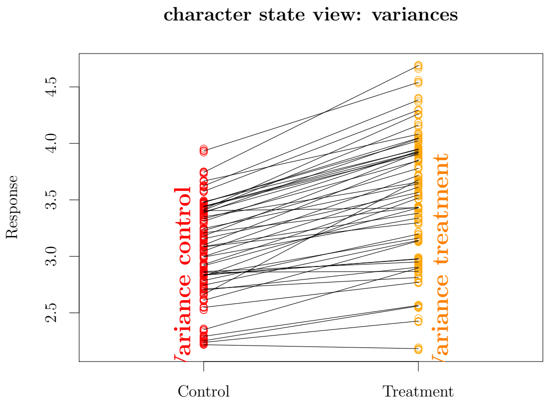
(hints: you could start from a lm() of mass modeled by environment, then add random intercepts, and finally a little something more).

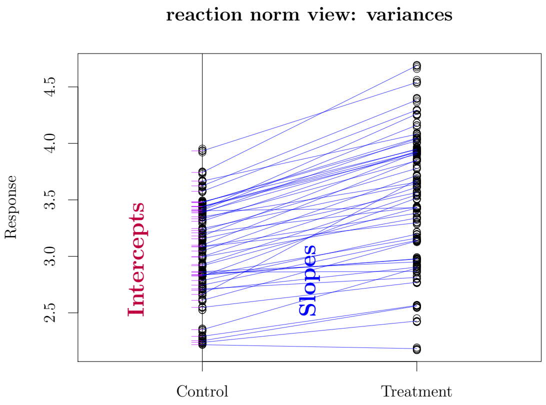
** Exercise 7 Beetles: look at the model

What are the variances related to genetic differences? How are they correlated? Does genetic variation explain a lot of the total variation we observe? Try and draw a representation of genetic variation in the two environments.

*** Exercise 8 Beetles: interpret

Interpret model outputs (use raw numbers and / or graphes) to answer the following:
Is there evidence for genetic variation? Do the two environment differ in their effects on beetles?
Is there evidence for genetic variation in the response to the environment?
Does that mean that genomes good at environment 1 are bad at environment 2?





2.3 Is there still time? Then let's talk about correlation between random intercept and random slope!

3 More resources

Ben Bolker's GLMM FAQ is a gold mine if you need some technical details about mixed models (generalized or not).

If you are going to do lots of mixed models consider subscribing to the mailing-list https://stat.ethz.ch/mailman/listinfo/r-sig-mixed-models, ask questions or search the archive (anyway, Google is likely to directly send you to the archive).