NRES 798 — Lab 12

Hierarchical and mixed effects models

1 Data, pine height growth

Load the Loblolly data set (`data(Loblolly)`), inspect it: `?Loblolly`, `summary`, `head`, `plot`, etc.

"`Seed`" seems to be just a tree id number. We want an unordered id: `Loblolly$tree <- factor(Loblolly$Seed, ordered=FALSE)`.

`table(Loblolly$age)` shows that there are measurements on 14 trees at a fixed set of ages. Most are at 5-year intervals. We will use 5-year height growth increments. Within-tree increments can be expected to be more nearly independent than current height (cumulative growth). At least if the height measurement error is not too large.

Since the data is ordered by age within trees (verify), an easy way of obtaining the increments (including some garbage that we will eliminate soon) is: `Loblolly$growth <- c(diff(Loblolly$height), 0)`. Make sure that you understand this; see what happens if you omit the `c(·, 0)`.

Drop the rows for the 2-year increments at age 3, and the "increments" at age 25 that are not increments at all, leaving ages 5 to 20: `lob <- Loblolly[Loblolly$age %in% c(5, 10, 15, 20), ]`. Check what we have now.

2 Two-level fixed effects models

What would be a good growth predictor? Plot growth over age, and over height. Plot height over age. Multicolinearity? Which one do you think makes more biological sense, size or elapsed time?

2.1 Common model

Let's choose height, it also looks more linear. Fit a simple linear regression: `slr <- lm(....` Plot growth over height, and add the regression line with `abline(slr)`. Seems reasonable?

This pools the data from all 14 trees. Maybe the trees have different slopes and/or intercepts? Let's see:

```
library(lattice)
xyplot(growth ~ height, lob, type='b', groups=tree)
```

2.2 Variable intercepts

Add the factor `tree` as a predictor: `summary(ints <- lm(growth ~ height + tree, lob))`. We get a common slope, which is the coefficient of `height`, and a different intercept for each tree. Remember that with the default contrast, the first level (first tree) is left out, so that `(Intercept)` is the intercept for the first tree. The intercepts for the other trees are obtained by adding the respective coefficients. Which is the first tree? Try `levels(lob$tree)`.

If this confuses you, use the "sum" contrast: `summary(ints.sum <- lm(growth ~ height + tree, lob, contrasts = list(tree = 'contr.sum')))`. Now `(Intercept)` is the mean intercept, and the tree coefficients are deviations from this. The deviations add to 0, so the last tree is omitted. Check that you get the same intercepts (and slope, and fit statistics).

Test the tree effect for significance: `anova(slr, ints)` (should get the same with `ints.sum`). Is there evidence of different intercepts? Does this agree with what you saw in the `xyplot` graph?

This is a (fixed-effects) *hierarchical* model or *multilevel analysis* (specifically, two-level). There are two units of analysis, or hierarchical levels. The first or lowest level are the growth measurements, which are *nested* into the next level, the trees. Some parameters, here the slope, are common to all the second level units (trees). Other parameters, here the intercept, are specific (different for) each unit. Our model for measurement j in unit (tree) i can be written as

$$Y_{ij} = \alpha_i + \beta x_{ij} + \varepsilon_{ij} \ .$$

2.3 Variable slopes

How about a common intercept and different slopes? That is,

$$Y_{ij} = \alpha + \beta_i x_{ij} + \varepsilon_{ij} \ .$$

The computer equivalent is `growth ~ height:tree` (think about it). Fit it, save the result as `slopes`. Examine and understand the `summary`.

Compare the fit statistics with those of the variable intercepts model, which one is better? Use `anova` to compare with `slr` and see if the contribution of the interaction (variable slopes) is statistically significant.

2.4 Separate regressions

Try both variable intercepts and slopes: `growth ~ height + tree + height:tree`, or `growth ~ height * tree`.

Note that this is the same as fitting separate regressions to each tree (we are letting each tree to have its own intercept and slope). Fit the simple linear regression to one tree, and check that the intercepts and slopes match.

Compare with the best model so far, see if there is a significant improvement.

The hypothesis testing side of all this, with balanced data, is an instance of analysis of covariance (ANCOVA).

3 Random effects

Our best model was of the form

$$Y_{ij} = \alpha_i + \beta x_{ij} + \varepsilon_{ij} \ ,$$

where Y_{ij} is the 5-year height growth increment in tree i, observation j, and x_{ij} is the corresponding current height. There is a common slope β, and a different intercept α_i for each tree. There are lots of parameters: one β, 14 α's, and σ.

A different way of modelling this is to assume that the trees are a simple random sample from a large population of trees, where the α's have a normal

distribution. Instead of the α_i being unknown parameters, they are assumed to be normal random variables with an unknown mean α and an unknown variance η^2. The α_i was a *fixed effect*, its replacement is a *random effect*. The model becomes

$$Y_{ij} = \alpha + \varepsilon_i + \beta x_{ij} + \varepsilon_{ij} \ ,$$

where ε_i is a normal random variable with mean 0 and variance η^2.

This is a (linear) *mixed effects model*. Ordinary regression models have only one random variable ε, these models have more than one. In R they are fitted with functions from packages `nlme` or `lme4`. Of course, much more complex examples are possible, there may be more than one random effect, multiple hierarchical levels, and the regression can be nonlinear. We stick with this simple one.

Load the package: `library(nlme)`. The model translates to

```
lme(growth ~ height, lob, random = ~ 1 | tree)
```

The first formula is the fixed effects part. The second one under `random` indicates a random intercept (1) varying among trees. Run and display the `summary`. What are the estimated α, β, σ, and η?

Compare to `summary(ints.sum)`, specifically the estimated slope and mean intercept, the residual standard error, and the residuals. You can `plot` the result of `lme` to get a graph of standardized residuals, compare it to the one from plotting `ints`.