| **In our last class on regression, we will explore mixed models, and specifically, linear mixed models** | |
| --- | --- |
| The first task is to create the data. As usual, creating data is illuminating, and helps us understand the structure of the data that form the basis of the model. Use the code to the right to create an independent variable. We have 300 observations. |  |
| Now we will create a grouping indicator. This is a label associated with each observation. The labels correspond to groups of data. Ideally, these are determined as part of an experimental design. In this example, imagine that these are 10 plots in a forest. In each plot there is the same species of plant. So each observation is a plant, and each plant is in exactly one plot. |  |
| Now we want to generate a vector of the correct length |  |
| Now we’ll put these vectors in one data frame. We now have a dataframe with two columns: x1 and g1 |  |
| We could now create a y variable the standard way. We can think of y as something like the height of the plant. Let’s assume that x1 is a measure of fertilizer applied to soil. So we have created data in which height is a function of fertilizer. |  |
| **Q1. Use lm to analyze the data. How might you incorporate g1 into this modelling process if you had to?** | |
| Now we will create a group effect. In the data creation process, we will assume that in each plot there is some process at work that affects how the plants grow. Each plot is different, and affects the plants in different ways. |  |
| **Q2. In one or two sentences, explain what is happening in this code.** | |
| It is important to note that in the real world, we do not know the g1\_effect values. These are **random effects**—differences between plots that we do not know before our analysis. Note the value of 5 in the code above. This is called a **fixed effect**. In the real world, we don’t know this value either. Until now, we have used models to estimate fixed effects. Now we are exploring random effects as well. | |
| Now let’s create the dependent variable, this time with the random effect included |  |
| **Q3. Write code that will create a unique regression model of the association between x1 and yre for each group. This should result in 10 models. Write code to store the results of each model in separate files (out1, out2, …, out10). You may wish to consider using a for loop for this task, and saving the output from the models in a single list() data structure.** | |
| Having 10 different models seems clumsy, and it does not easily tell us whether the variation across the plots is noteworthy. | |
| We will now use a new library to analyze a mixed model in R. This will estimate the fixed effect (x1) as well as the random effect. The random effect is a measure of variance. It will estimate the values in the process that generates the random effect. In this case (*see the code to the right*), it will estimate the value of ‘10’, which is the standard deviation of the process we used to generate the data. Actually, it also estimates this value squared (10 x 10), which is the variance. |  |
| Install and load the lme4 package. |  |
| Write and run the code to the right. |  |
| You should now see output similar to this: | |
| The fixed effect for x1 is easy to interpret—it is the same as usual. However, the random effects are more tricky. Note that the g1 random effect is 14.343 (we expected 10). The other random effect is the normal residual. It is a value of ~1. Notice that the variation captured in the differences between groups is quite a bit larger than the differences between individual observations. This particular example shows that the differences in height between groups (the ‘plots’ in our hypothetical example) are very important for explaining differences between the plants, especially when compared to the variations in height between the plants alone. | |
| What this modeling process has done is to help us understand model error (meaning differences between the observed value of y and predicted value of y) of two types—the between group errors, and the within group errors.    Note that the plants differ in height. But the variations between the group averages are pretty big compared to the differences within the groups. For example, see that the bottom right plants are all pretty big, the bottom left are all pretty small, and so on. So in this example, the differences between groups are probably more important than the differences between individuals.  The model results above suggest something like this—where the variation in groups is larger than the variation between them. This makes sense given how we generated the data. The group effect had a standard deviation of 10—which is large. The remaining error (at the level of the individual) has a standard deviation of 1 (remember that by default, rnorm() has a mean of 0 and a standard deviation of 1). | |
| It is possible to pull out random intercept effects for each of the groups. These are the effects associated with each group, so the vector, re, is the same length as the number of groups. |  |
| Above we created the random effects and stored them in a vector called group\_effect. We should expect group\_effect (the data we created) and the model estimated random effect (re) to be similar. We can plot imputed and modeled random effects using the code to the right |  |
| **Q4. How would you judge the success of the model at estimating the random effects associated with the groups based on looking at the plot? Explain your reasoning.** | |
| The next step is to explore the idea of random slope effects. Normally, we fit regression models assuming that an independent variable has a ‘global’ association with the dependent variable. In other words, it’s a single relationship that can be understood with a term or two in a regression equation. However, it is possible that the relationship between a dependent and independent variable varies by group. For example, imagine we are modeling the relationship between individual health and individual wealth internationally. Say we have 40 countries, and there is a sample of 1000 people in each country. We assess their personal wealth and their health. It is easy to imagine that that relationship might be stronger in some countries than others. For example, in wealthy countries where there is universal health care, you may think that wealth does not influence health that much. In poorer or middle income countries, wealth and health might be more strongly associated with one another. The result might be a graphical representation something like what you see below:    In this figure, health varies across countries (the random intercept effect) and the relationship between wealth and health varies across countries too. | |
| Rather than synthesizing this, we will use some real data. Download these data to your computer:  <https://drive.google.com/file/d/1OJJ24UXcCCFbigCdc0Pr1UByWMV12-nO/view?usp=drive_link>  These are real-estate data from the US.  Import it into R, remembering to set the directory to wherever you saved the data: | |
| Once you’ve imported the data, fit a simple regression model using the code to the right. The independent variable is the living area (in square feet) and the dependent variable is sale price. |  |
| **Q5. Interpret the results of this model. What is the impact of living area (in square feet) on the value of a home?** | |
| We’re going to treat neighbourhoods as groups in our analysis. So best practice is to ensure that the grouping variable is a factor |  |
| In practice, it is often easiest to model the standard scores (or some other transformation) of data rather than the raw data. This involves projecting the observations on a Standard Normal distributions with a mean of 0 and a standard deviation of 1. |  |
| Now let’s use regular regression. Because these are standard scores, we have to interpret the model in those terms. For every 1 standard deviation increase in living space, there is a ~ 0.71 increase in sales price. |  |
| Now run a random intercept only model |  |
|  | |
| In this model, we only have the random effects (errors at the neighbourhood level and the household level) and the fixed intercept. The errors are of similar magnitude (0.8357 for the neighbourhood-level variation and 0.6562 for the house variation). | |
| And now finally: | |
| The results from this model says a lot. First, there is important variation in random intercepts and random slopes. This means that house prices vary across neighbourhoods (random intercept effect) and that the relationship between house price and living space varies by neighbourhood. Moreover, the 0.71 under the column ‘Corr.’ tells us there is a positive correlation between the random intercept and random slope effects. This suggests that the relationship between sales price and living space is stronger when the neighbourhood has higher sales prices. Note that the living area fixed effect predicts the sales price (1 standard deviation increase in living space corresponds to a ~ 0.52 increase in the standard deviation of sales price). | |
|  | |
| **Q6. What is the ‘N’ vector, and what is it doing in this code?** | |
| Wow! Each **dot** is a model prediction of sales prices based on living space for each house. Each **line** is the model prediction for one of the neighbourhoods. At the top are neighbourhoods with high predicted sales prices. Note also that the slope is steeper for these neighbourhoods. In contrast, in neighbourhoods with lower sales prices the slope is less steep. These results are consistent with the idea that 1) price varies by neighbourhood (random intercept > 0) 2) sales price is associated with living space (fixed effect > 0) 3) the association between price and living space varies systematically across neighbourhoods and (random\_slope\_effect > 0) 4) in neighbouhoods with higher sales price, the relationship between sales price and living space is stronger than in neighbourhoods with lower sales prices (correlation between random intercept and slope effects). | |
| **Q7. Use another independent variable in this data set and go through the same steps above, interpreting the results accordingly.** | |